KOREATECH 한국기술교육대학교

Term Project Report

학번	2020136087
이름	윤아현
수강 과목	데이터마이닝
담당 교수	전강욱 교수님
제출일	2024.06.18

데이터마이닝 프로젝트

1 번. 도메인 설명 및 분석 목적

취수원(하천수, 호수, 지하수 등)의 물을 수질기준(수돗물, 식수, 등)에 적합한 물로 처리하는 과정을 정수처리라고 한다.

정수처리 과정은 한국 수처리 기업인, 한국수자원공사에서 관리하고 있다. 정수처리 과정은 크게 7 단계로 이루어져 있다. 아래의 표는 각 단계별 작업을 나타낸 그림 및 표이다.

단계	설명
1 단계: 취수	취수원에서 물이 들어오는 단계
2 단계: 착수	들어온 물을 안정화 시키는 단계
3 단계: 혼화	응집제와 물을 섞는 단계
4 단계: 응집	응집제와 물이 섞이면서 불순물을 덩어리로 만드는 단계
5 단계: 침전	무거워진 덩어리가 바닥으로 가라앉는 단계
6 단계: 여과	필터를 통해 물의 불순물을 제거하는 단계
7 단계: 소독	물에 남은 미생물을 제거하는 단계

이 과정에서, 우리가 주목해야 할 단계는 혼화 및 응집이다. 이 단계는 '응집제'라는 화학약품을 주입하여 미세입자, 박테리아, 및 불순물과 같은 더러운 물질을 응집시켜 침전시키는 역할을 한다. 이 때, 응집제 주입이 잘못될 경우 후공정에 악영향을 미칠 수 있다.

그렇지만, 현재는 실무자들의 경험을 바탕으로 응집제 주입률을 결정하고 있기 때문에 휴먼 에러가 발생할 가능성이 있고, 최종 의사결정까지 많은 시간이 소요되고 있다.

이러한 문제점을 해결하기 위해, 현재까지 수집된 여러 인자 및 응집제 주입률에 대한 경향성 및 숨겨진 패턴을 분석하려고 한다.

2 번. 수집한 데이터 설명

한국수자원공사 창원권지사 반송정수장에서 2013 년 01 월부터 2023 년 8 월까지 수집한 데이터이다. 실시간으로 수집되는 센서 데이터와 응집제 주입량이 변수로 들어가있다.

변수는 13 개이며, 총 93,457 개의 데이터로 이루어져 있다. 데이터는 1 시간 간격으로 꾸준히 측정된 시계열 데이터이다.

변수: logTime, 침전수 탁도, 기존 정수지 탁도, 원수 탁도, 원수 알칼리도, 원수 전기전도도, 원수유입유량, 원수 pH, 원수 온도, PAC, 주입량 PACS 주입량, 응집지 pH, Co2 주입량, Co2 주입량 Run

각 선택한 변수에 대한 설명을 아래의 표로 정리하였다.

변수	설명
탁도	물의 탁한 정도 (탁도가 높을수록, 물이 뿌옇게 보임)
pH	물의 산성도나 알칼리도를 나타내는 척도 (하천수는 5~7 로 나타남
수온	물의 온도 (계절적 영향을 많이 받음)
전기전도도	물이 전기를 전도하는 능력
알칼리도	물이 산을 중화시키는 능력
PAC, PACS 주입량	부유물질을 응집시켜 침전시키는 데 사용

- **이 데이터는 외부에 노출되면 안되기 때문에, 따로 첨부하지 않았다.**

3 번. 데이터 전처리

※ 발표 당시 R 코드로 저장하지 않은 문제로 다시 수행하였습니다. 미세한 차이는 있지만, 전체적인 과는 동일합니다.

데이터를 알아보기 쉬운 Column 명으로 변경해주었고, Data Type 이 chr 로 지정되어 있어, numeric 으로 모두 변경을 수행해주었다.

또한, 응집제 주입량과의 Feature Importance 가 가장 높은 변수만을 선택하여 분석하였다.

선택한 변수: 원수 탁도, pH, 수온, 알칼리도, 전기전도도, 원수유입유량

이 때, PAC 과 PACS 가 응집제 주입량을 나타낸다.

1) 기본 통계량 확인하기

총 data 개수: 93,460 개

```
> str(kwater_df)
tibble [93,460 × 9] (S3: tbl_df/tbl/data.frame)
 $ logTime     : chr [1:93460] "2013/01/01 01:00" "2013/01/01 02:00" "2
013/01/01 03:00" "2013/01/01 04:00" ...
 $ 탁도        : num [1:93460] 7.7 7.26 6.92 6.91 6.75 ...
 $ 알칼리도    : num [1:93460] 99.5 99.5 99.6 99.5 99.5 ...
 $ 전기전도도  : num [1:93460] 315 317 318 319 320 ...
 $ 원수유입유량: num [1:93460] 2426 2424 2442 2442 2436 ...
 $ pH          : num [1:93460] 7.65 7.65 7.65 7.65 7.65 ...
 $ 수온        : num [1:93460] 2.5 2.44 2.65 2.38 2.48 ...
 $ PAC         : num [1:93460] 0 0 0 0 0 0 0 0 0 0 ...
 $ PACS        : num [1:93460] 47.2 46.6 46.6 47 47.6 ...
```

2) 결측치 확인하기

2 개로 나누어진 PAC 와 PACS 는 모두 같은 응집제이기 때문에 이 두개의 칼럼을 "PACS 주입량"으로 하나의 칼럼으로 합친다.

그 뒤, 결측치를 확인해보았을 때 아래와 같은 결과가 나왔다.

```
> print(missing_counts)
        탁도      알칼리도  전기전도도 원수유입유량            pH
        8105          8106          8106          8132          8109
        수온           PAC          PACS 응집제주입량
        8105          8122          8122          8122
>
```

응집제주입량이 0 이거나 NA 인 것들은 다 삭제하였다. 삭제 후에도, 몇몇 데이터에 결측값이 있는 것을 아래의 그림으로 확인할 수 있다.

```
> colSums(is.na(df_new))
     logTime         탁도         수온     알칼리도  전기전도도
           0            2            2            3            3
원수유입유량           pH          PAC         PACS 응집제주입량
          25            6            0            0            0
```

데이터가 sequential 하기 때문에 선형보간법으로 결측값을 채워주었다.

총 데이터 개수: 81,725

***파생변수 생성**

1. PAC 주입량 및 PACS 주입량은 모두 동일한 응집제이기 때문에 두 개의 값을 더해주어 "응집제 주입량" 이라는 파생변수를 생성하였다.

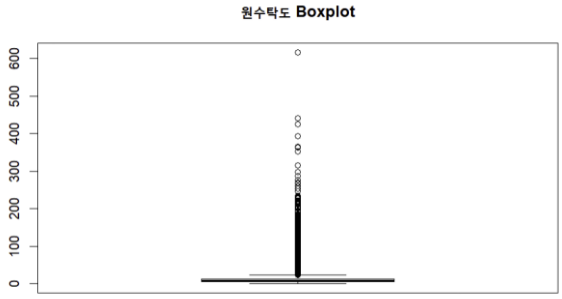
2. 응집제는 주입량이 아닌 주입률 형태로 주입되기 때문에 새로운 파생변수를 생성하였다.

응집제 주입률 = (응집제 주입량) / (원수유입유량) * 1,000 (ppm 단위)

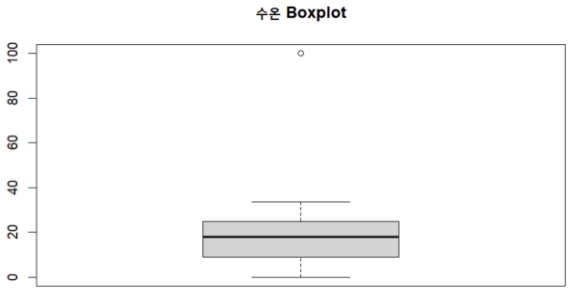
3) 이상치 확인하기

각각의 특징 변수들에 대해 boxplot 을 그려 이상치를 확인한 뒤, 삭제를 진행한다.

1. 원수탁도



2. 수온

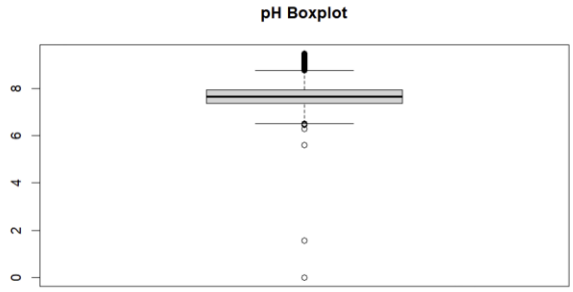


(1) 원수탁도: 500NTU 이상 삭제

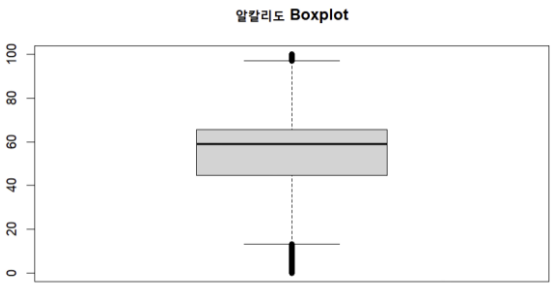
원수탁도와 같은 경우, 범위가 넓고 특정 상황에서는 200NTU 가 넘을 수 있기 때문에 삭제하지 않았음

(2) 수온: 수온이 100 이상인 값이 존재하면 안되기 때문에 삭제하였음

3. pH



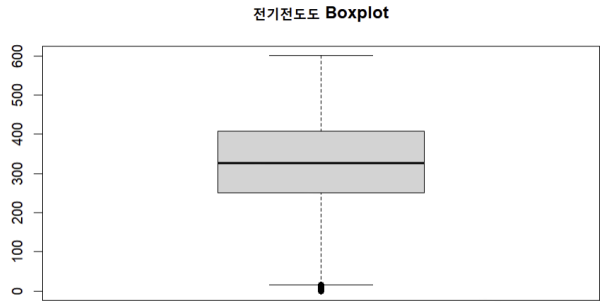
4. 알칼리도



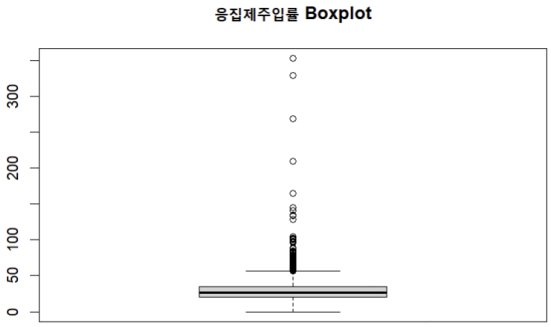
(3) pH: 하천수의 pH 는 5 보다 작으면 안됨, 삭제함

(4) 알칼리도: 알칼리도가 20 보다 이하일 수 없기 때문에 삭제함

5. 전기전도도



6. 응집제 주입률



(5) 전기전도도: 10 아래 삭제함

(6) 응집제 주입률: 100 이상 삭제함

총 데이터 개수: 81,725

4 번. 데이터 분석

1) 연관규칙 분석

(1)각 인자간들 사이의 상관관계 및 (2)응집제 주입률과 다른 인자 간의 상관관계를 파악하기 위한 목적으로 연고나 규칙 분석을 수행하였다.

모든 인자들의 값이 수치형이기 때문에, 이를 범주형으로 변경해주었다. 각 인자들에서 비율을 정하여 Low, Medium, High 로 나타내주었다.

(1) 각 인자간들 사이의 상관관계

```
lhs                                    rhs                  support
[1] {pH=High, Conductivity=High}     => {WaterTemp=Low}    0.1209549
[2] {WaterTemp=High, Alkalinity=Low} => {Conductivity=Low} 0.1245613
[3] {Turbidity=High, WaterTemp=High} => {Conductivity=Low} 0.1274635
[4] {Turbidity=High, Alkalinity=Low} => {Conductivity=Low} 0.1511177
    confidence coverage  lift     count
[1] 0.8355142  0.1447670 2.531814  9961
[2] 0.8342550  0.1493085 2.527905 10258
[3] 0.8266656  0.1541899 2.504908 10497
[4] 0.8061277  0.1874613 2.442676 12445
```

이 규칙들을 확인해보았을 때, 대부분 전기전도도와 수온에 대한 연관규칙이 생성된 것을 확인할 수 있다.

이를 통해, 낙동강 유역의 하천에서는 전기전도도와 수온이 반비례하는 성질을 가진다는 것을 관찰할 수 있다.

(2) 응집제 주입률과 다른 인자간의 상관관계

```
     lhs                          rhs             support confidence
coverage     lift count
[1]  {pH=Low,
      Turbidity=High,
      WaterTemp=High,
      Conductivity=Low,
      Alkalinity=Low}    => {PACS=High} 0.05567496  0.8389753
0.06636067 2.542302  4585
[2]  {pH=Low,
      Turbidity=High,
      WaterTemp=High,
      Alkalinity=Low}    => {PACS=High} 0.05835853  0.8284778
0.07044066 2.510492  4806
[3]  {pH=Low,
      Turbidity=High,
      WaterTemp=High,
      Conductivity=Low}  => {PACS=High} 0.07376780  0.8272059
0.08917708 2.506637  6075
```

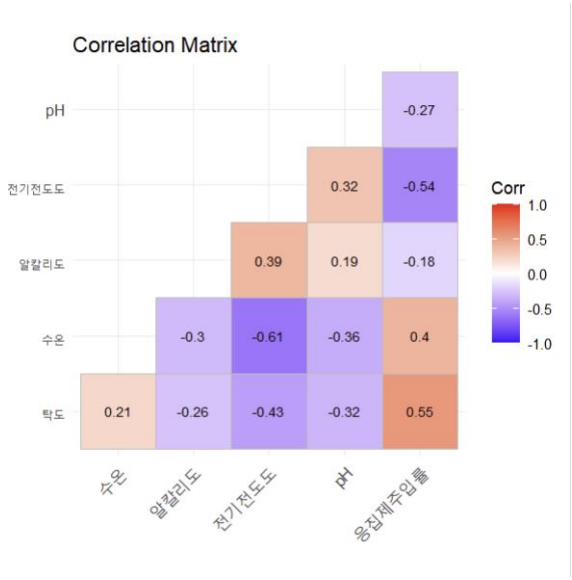
상위 10 개의 규칙 중, 유의미한 결과 몇 개를 뽑아냈을 때의 결과이다.

대부분의 연관규칙에서 응집제 주입률이 높아지는 시점은 탁도와 수온이 높고, 알칼리도와 전기전도도가 낮을 때 발생한다는 사실을 알 수 있다.

탁도는 강수량과 같은 부유물질이 하천으로 떠내려오는 현상이 발생할 때 높아지며, 수온은 여름에 높다.

이 두가지 사실을 통해, 비가 오는 여름철에 응집제 주입률인 높아진다는 사실을 간접적으로 알 수 있다.

2) 상관관계 확인하기



응집제 주입률과 다른 특징 변수들 간의 상관관계를 시각화하였을 때,

응집제 주입률과 원수탁도의 양의 상관관계가 0.55 로 가장 높았고 전기전도도와는 음의 상관관계가 -0.54 로 높았다.

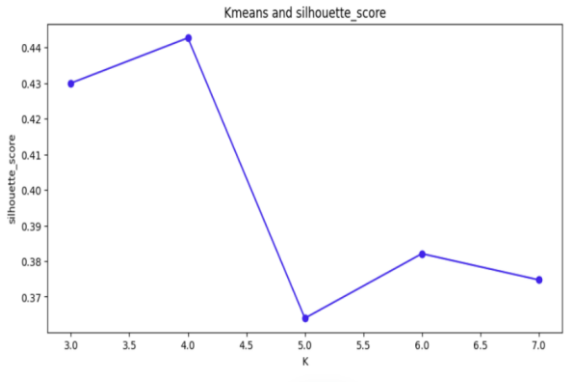
3) 군집화

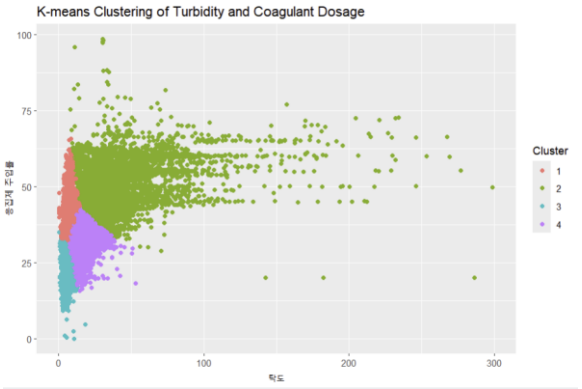
응집제 주입률와 원수탁도만을 적용하여, 군집화를 수행하였다. 그 이유는, 원수탁도와 상관관계가 가장 높기도 하고 다른 인자들을 모두 적용하여 군집화를 수행하였을 때 경향성 파악이 불가하였기 때문이다.

이 때, 탁도의 경향성을 더 잘 나타내기 위해 log 변환을 수행하였다.

군집화 방식은 KMeans 클러스터링을 수행하였다. (데이터가 모두 밀집되어 있고, 거리 기반으로 수행하였을 때 결과가 가장 잘 나옴)

아래는 실루엣 계수 및 군집화의 결과를 나타낸 그림이다. (실루엣 계수는 python 으로 확인함 – R 언어는 Memory 오류 발생)

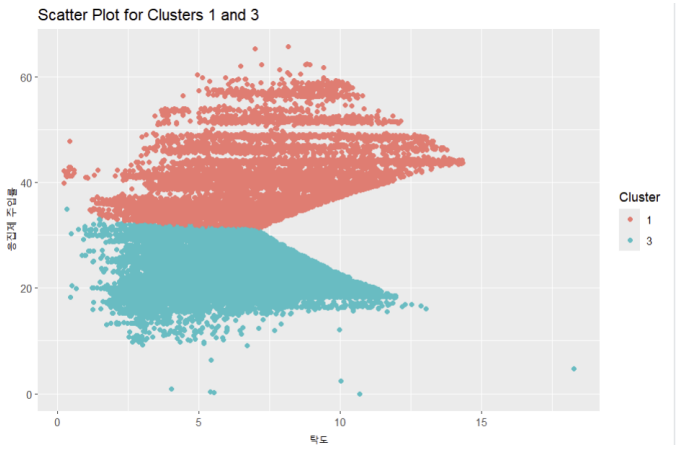




각 Cluster 에서 탁도 및 응집제 주입률에 대한 평균 값을 확인해보았을 때, 군집 1 와 3 서 탁도의 평균은 비슷하지만, 응집제 주입률에서 차이가 나타나는 것을 확인할 수 있다.

```
# A tibble: 4 × 3
  cluster mean_탁도 mean_응집제주입률
    <int>     <dbl>             <dbl>
1       1      6.85              40.9
2       2     36.6               50.1
3       3      5.97              21.8
4       4     14.0               30.6
```

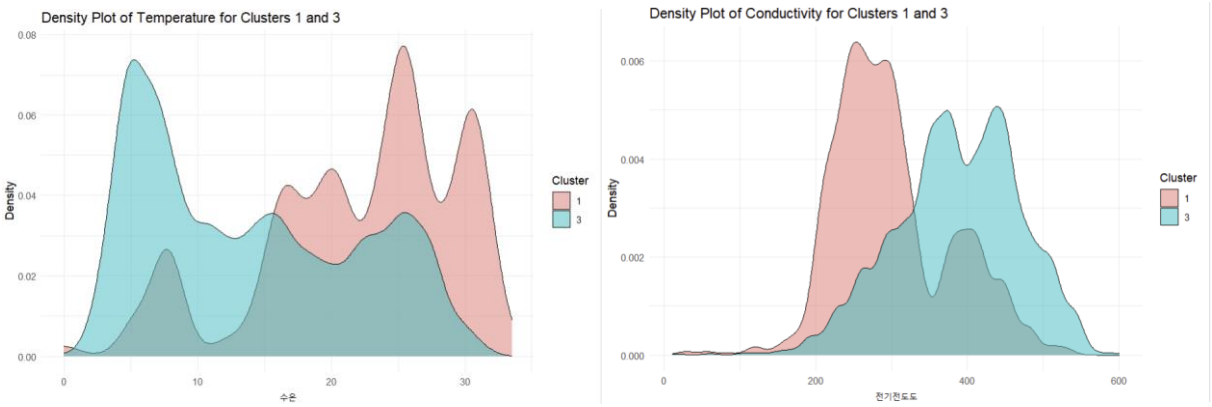
아래의 그림에서도 응집제 주입률을 기준으로 군집이 나누어진 것을 확인할 수 있다.



이러한 이유가 나타나는 이유를 파악하기 위해 추가적인 분석을 수행하였다.

3) 군집화 후 분석 결과

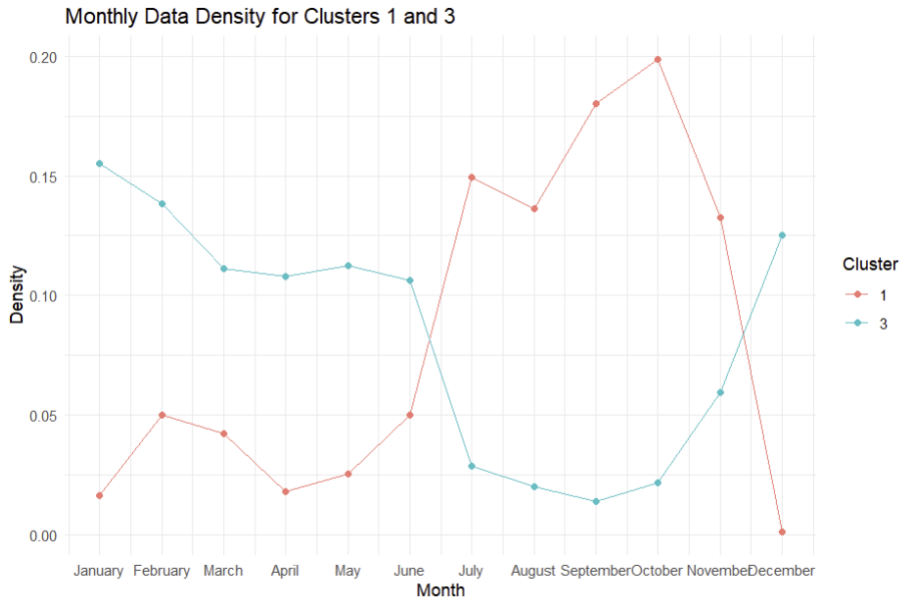
군집 1 과 3 에서 수온 및 전기전도도에서의 밀도 비교를 수행하였고 결과는 아래의 그림과 같다.



결과를 보았을 때 군집 1 에 존재하는 데이터는 대부분 높은 수온, 낮은 전기전도도를 보였고 군집 3 은 낮은 수온, 높은 전기전도도를 보인다는 것을 확인할 수 있다.

연관규칙 분석에서 확인하였듯이 이 두개의 변수는 반비례한다는 사실을 한 번 더 확인할 수 있었다.

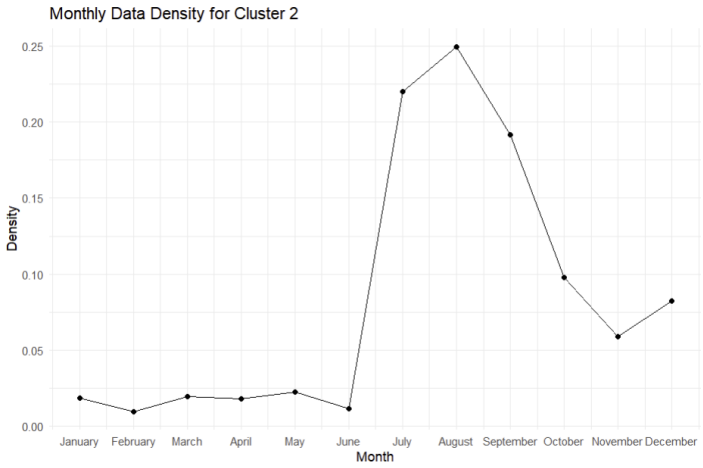
수온의 차이는 곧 계절의 차이라고 생각하여, 저탁도 군집(군집 1, 3)에서의 월간 데이터 비율을 비교해보았다. 그 결과는 아래와 같다.



추측한대로, 군집 1 은 여름철(7 월~10 월), 즉 태풍 및 장마철이 자주 발생하는 달에 데이터가 몰려있고 군집 3 은 그 외의 계절(봄, 가을, 겨울)에 데이터가 몰려있는 것을 확인할 수 있다.

즉, 같은 저탁도 군집이어도 **계절에 따라 응집제 주입률의 차이**가 나타나는 것을 확인할 수 있다.

이외에도, 고탁도 군집에서의 월간 데이터 비율을 비교해보았을 때 결과는 아래와 같다.



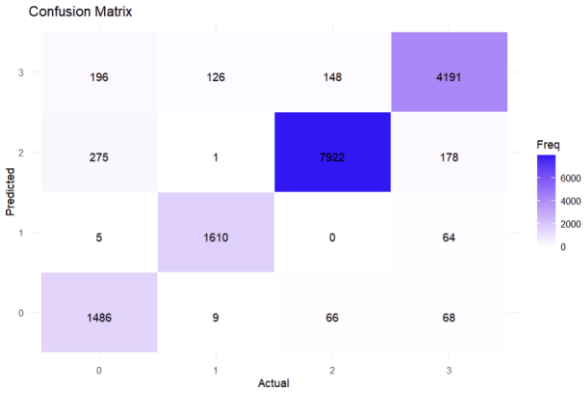
고탁도 또한, 여름철(7, 8, 9, 10 월)에 자주 발생하는 것을 알 수 있다.

4) 분류 모델 생성

이 분류 모델은 실시간으로 데이터가 주입되었을 때, 해당하는 Cluster 로 정확하게 분류하기 위한 목적으로 설계되었다.

분류 모델의 독립 변수로 탁도, pH, 수온, 전기전도도, 알칼리도를 사용하였고 종속 변수로는 Cluster 를 사용하였다. 독립 변수에 대한 표준 정규화를 수행하여 단위를 맞춰주었다.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2    3
         0 1486    9   66   68
         1    5 1610    0   64
         2  275    1 7922  178
         3  196  126  148 4191

Overall Statistics

               Accuracy : 0.9305
                 95% CI : (0.9265, 0.9344)
    No Information Rate : 0.4978
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.892

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3
Sensitivity          0.75739   0.9221   0.9737   0.9311
Specificity          0.99006   0.9953   0.9447   0.9603
Pos Pred Value       0.91222   0.9589   0.9458   0.8992
Neg Pred Value       0.96765   0.9907   0.9731   0.9735
Prevalence           0.12004   0.1068   0.4978   0.2754
Detection Rate       0.09091   0.0985   0.4847   0.2564
Detection Prevalence 0.09966   0.1027   0.5125   0.2852
Balanced Accuracy    0.87372   0.9587   0.9592   0.9457
```



분류 모델로는, 성능이 가장 좋은 앙상블 모델인 XgBoost Classifier 를 사용하였다.

대표 성능 지표는 Accuracy 를 사용하였으며, 93.05%로 높은 성능을 보였다. Confusion Matrix 를 확인했을 때 대부분 해당 군집에 잘 분류하지만 데이터 간의 불균형으로 인해 몇 몇의 군집들은 제대로 분류하지 못하는 것을 볼 수 있다.

5 번. 결과

연관규칙 및 군집화를 통해 유의미한 결과를 얻어낼 수 있었다. 대부분, 화학적 성질에 의해 전기전도도와 수온은 비례하는 성질을 가지고 있는데 낙동강 유역은 하천수의 성질로 인해 전기전도도와 수온이 반비례하는 성질을 가진다는 신기한 사실을 관찰할 수 있었다. 또한, 군집화를 통해 응집제 주입률이 실시간 주입되는 주요 인자들에 따라 대부분 동일하게 들어가는 것이 아닌 계절적 영향에 따라 다르게 주입되고 있다는 사실을 분석할 수 있었다.

6 번. 고찰

사실, Python 으로만 데이터를 분석해봐서 처음에는 R 언어가 어색하고 낯설었지만, 프로젝트 덕분에 R 언어 사용법에 익숙해졌다. 또한, 연관규칙기법을 처음 학습하였는데 연관규칙을 통해 유의미한 패턴을 찾을 수 있었다. 군집화보다 연관규칙이 더욱 빠르고 쉽게 결과를 파악할 수 있는 것 같아 종종 데이터 분석을 수행할 때 사용해 봐야겠다는 생각이 들었다.

이번 과제를 통해, 데이터 분석을 수행하는 단계와 방법에 대해 한 번 더 학습하고 실습에 적용할 수 있어 매우 유익한 시간이었다.

Code 첨부

※ 주요 코드만 첨부함

- 연관규칙

```
> breaks_turbidity <- quantile(data$Turbidity, probs = c(0, 0.33, 0.67, 1),
na.rm = TRUE)
> data$Turbidity <- cut(data$Turbidity, breaks = breaks_turbidity,
include.lowest = TRUE, labels = c("Low", "Medium", "High"))
breaks_pH <- quantile(data$pH, probs = c(0, 0.33, 0.67, 1), na.rm = TRUE)
data$pH <- cut(data$pH, breaks = breaks_pH, include.lowest = TRUE, labels =
c("Low", "Medium", "High"))
breaks_waterTemp <- quantile(data$WaterTemp, probs = c(0, 0.33, 0.67, 1),
na.rm = TRUE)
data$WaterTemp <- cut(data$WaterTemp, breaks = breaks_waterTemp,
include.lowest = TRUE, labels = c("Low", "Medium", "High"))
breaks_conductivity <- quantile(data$Conductivity, probs = c(0, 0.33, 0.67,
1), na.rm = TRUE)
data$Conductivity <- cut(data$Conductivity, breaks = breaks_conductivity,
include.lowest = TRUE, labels = c("Low", "Medium", "High"))
breaks_alkalinity <- quantile(data$Alkalinity, probs = c(0, 0.33, 0.67, 1),
na.rm = TRUE)
data$Alkalinity <- cut(data$Alkalinity, breaks = breaks_alkalinity,
include.lowest = TRUE, labels = c("Low", "Medium", "High"))
breaks_PACS <- quantile(data$PACS, probs = c(0, 0.33, 0.67, 1), na.rm =
TRUE)
```

```
data$PACS <- cut(data$PACS, breaks = breaks_PACS, include.lowest = TRUE,
labels = c("Low", "Medium", "High"))
transactions <- as(data, "transactions")
```

(1) 주요 변수 간의 연관규칙

```
> rules_excluding_PACS <- apriori(transactions,
+                                 parameter = list(supp = 0.1, conf = 0.8),
+                                 appearance = list(none = c("PACS=Low",
"PACS=Medium", "PACS=High")))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1
 maxlen target  ext
     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 8235

set item appearances ...[3 item(s)] done [0.00s].
set transactions ...[18 item(s), 82353 transaction(s)] done [0.03s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.04s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [4 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> num_rules_excluding_PACS <- length(rules_excluding_PACS)
> if (num_rules_excluding_PACS > 0) {
+     inspect(sort(rules_excluding_PACS, by = "confidence")[1:min(10,
num_rules_excluding_PACS)])
+ } else {
+     cat("No rules found for rules_excluding_PACS.\n")
+ }
```

(2) 응집제와 주요 변수 간의 연관규칙

```
> rules_towards_PACS <- apriori(transactions,
+                               parameter = list(supp = 0.05, conf = 0.6),
+                               appearance = list(rhs = c("PACS=Low",
"PACS=Medium", "PACS=High")))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.6    0.1    1 none FALSE            TRUE       5    0.05      1
 maxlen target  ext
     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
```

```
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 4117

set item appearances ...[3 item(s)] done [0.00s].
set transactions ...[18 item(s), 82353 transaction(s)] done [0.03s].
sorting and recoding items ... [18 item(s)] done [0.00s].
creating transaction tree ... done [0.03s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [49 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> num_rules_towards_PACS <- length(rules_towards_PACS)
> if (num_rules_towards_PACS > 0) {
+     inspect(sort(rules_towards_PACS, by = "confidence")[1:min(10,
num_rules_towards_PACS)])
+ } else {
+     cat("No rules found for rules_towards_PACS.\n")
+ }
```

- 군집화

```
> df_kmeans <- df_outlier %>% select(탁도, 응집제주입률)
> df_kmeans <- df_kmeans %>% mutate(log_탁도 = log(탁도))
> data_scaled <- scale(df_kmeans[, c("log_탁도", "응집제주입률")])
> set.seed(123)  # 결과 재현을 위한 시드 설정
> kmeans_result <- kmeans(data_scaled, centers = 4, nstart = 25)
> df_kmeans <- df_kmeans %>% mutate(cluster = kmeans_result$cluster)

> # df_kmeans_result 데이터프레임 생성 (원본 탁도, 응집제주입률, cluster 포함)
> df_kmeans_result <- df_outlier %>%
+     select(탁도, 응집제주입률) %>%
+     mutate(cluster = df_kmeans$cluster)
> > ggplot(df_kmeans_result, aes(x = 탁도, y = 응집제주입률, color =
factor(cluster))) +
+     geom_point() +
+     xlim(0, 300) +
+     labs(color = "Cluster", title = "K-means Clustering of Turbidity and
Coagulant Dosage") +
+     xlab("탁도") +
+     ylab("응집제 주입률")


> library(xgboost)
> library(caret)
> data <- df_outlier[, c("탁도", "pH", "수온", "전기전도도", "알칼리도",
"cluster")]
> data[, -6] <- scale(data[, -6])
> > set.seed(123)
> index <- createDataPartition(data$cluster, p = 0.8, list = FALSE)
> train <- data[index, ]
> test <- data[-index, ]
```

```
> > train_matrix <- xgb.DMatrix(data = as.matrix(train[, -6]), label =
as.numeric(train$cluster) - 1)
> test_matrix <- xgb.DMatrix(data = as.matrix(test[, -6]), label =
as.numeric(test$cluster) - 1)
> params <- list(
+     booster = "gbtree",
+     objective = "multi:softmax",
+     num_class = length(unique(data$cluster)),
+     eta = 0.1,
+     gamma = 0,
+     max_depth = 6,
+     min_child_weight = 1,
+     subsample = 0.8,
+     colsample_bytree = 0.8
+ )
> > xgb_model <- xgb.train(params = params, data = train_matrix, nrounds =
100, verbose = 0)
> preds <- predict(xgb_model, test_matrix)
> > confusion_matrix <- confusionMatrix(factor(preds, levels =
0:(length(unique(data$cluster))-1)),
+                                     factor(as.numeric(test$cluster) - 1,
levels = 0:(length(unique(data$cluster))-1)))
> print(confusion_matrix)
> conf_matrix_df <- as.data.frame(confusion_matrix$table)
> names(conf_matrix_df) <- c("Prediction", "Reference", "Freq")
>
> # 혼동 행렬 시각화
> ggplot(data = conf_matrix_df, aes(x = Reference, y = Prediction)) +
+     geom_tile(aes(fill = Freq), color = "white") +
+     scale_fill_gradient(low = "white", high = "blue") +
+     geom_text(aes(label = Freq), vjust = 1) +
+     labs(x = "Actual", y = "Predicted", title = "Confusion Matrix") +
+     theme_minimal()
```