BUSS256 빅데이터분석과해석의기초

Final Report

2020120036 오윤진

1장. 서론

1.1 분석의 목적

본 보고서에서는 큰 환경 문제 중 하나로 대두되고 있는 미세먼지에 대해 알아보고자 한다. 미세먼지는 입자의 크기에 따라 지름이 10㎛ 보다 작은 미세먼지(PM10)와 지름이 2.5㎛ 보다 작은 초미세먼지(PM2.5)로 나뉜다. 본 보고서는 그 중 초미세먼지를 기준으로 하여, 주변 대기 환경에 따라 어떠한 미세먼지 등급이 나타나는지 예측 분석을 통해 알아보고자 하였다.

1.2 분석 데이터에 관한 기본 정보

- **이름:** 스마트서울 도시데이터 센서(S-DoT) 환경정보
- **출처:** 서울 열린데이터 광장(data.seoul.go.kr)
- **URL:** http://data.seoul.go.kr/dataList/OA-15969/S/1/datasetView.do
- **내용:** 2021년 3월 1일부터 2021년 3월 7일까지, 서울시 전역에 설치된 스마트 서울 도시데이터 1,100대를 통해 측정된 미세먼지, 온도, 습도, 조도, 소음, 진동, 자외선, 풍향, 풍속 등의 환경정보에 대한 1시간 단위 평균값
- **크기:** 118435행*28열

```
> head(M_org)
   Organization OutserverNum DataNum   Model   SerialNum Sortation MicroPM PM Temp RelHumidity WindDirec WindSpeed GustDirec GustSpeed Illumination
1:       서울시           48       1 SDOT001 0C3CL200010         1       2  3  6.7          43        NA        NA        NA        NA           NA
2:       서울시           48       1 SDOT001 0C3CL200012         1       2  4  5.8          44        NA        NA        NA        NA           NA
3:       서울시           48       1 SDOT001 0C3CL200011         1       2  3 21.6          88        NA        NA        NA        NA           NA
4:       서울시           48       1 SDOT001 0C3CL200017         1       1  2  6.4          42        NA        NA        NA        NA           NA
5:       서울시           48       1 SDOT001 0C3CL200013         1       1  2  5.6          43        NA        NA        NA        NA           NA
6:       서울시           48       1 SDOT001 0C3CL200016         1       3  5  6.2          41        NA        NA        NA        NA           NA
   UVrays Noise Vib_x Vib_y Vib_z MaxVib_x MaxVib_y MaxVib_z WaterEva MicroPMrev PMrev TransmitTime              Date
1:      0    39    NA    NA    NA       NA       NA       NA      4.0          2     3 202103021700 2021-03-02 18:07:00
2:      0    60    NA    NA    NA       NA       NA       NA      3.3          2     4 202103021700 2021-03-02 18:07:00
3:      0    49    NA    NA    NA       NA       NA       NA     17.7          2     3 202103021700 2021-03-02 18:07:00
4:      0    67    NA    NA    NA       NA       NA       NA      5.1          1     2 202103021700 2021-03-02 18:07:00
5:      0    48    NA    NA    NA       NA       NA       NA      3.3          1     2 202103021700 2021-03-02 18:07:00
6:      0    52    NA    NA    NA       NA       NA       NA      3.4          3     5 202103021700 2021-03-02 18:07:00
```

데이터는 총 118435개의 행과 28개의 열로 이루어져 있으며, 각각의 행은 특정 시간대마다 각각의 센서에서 측정한 환경정보 개별 값에 해당한다.

28개의 열은 위와 같이 기관명(Organization), 송신 서버 번호(OutserverNum), 데이터 번호(DataNum), 모델명(Model), 시리얼(SerialNum), 구분(Sortation), 전송시간(TransmitTime), 등록일자(Date) 등 **관측 센서에 관한 항목**과 초미세먼지(MicroPM), 미세먼지(PM), 기온(Temp), 상대습도(RelHumidity), 풍향(WindDirec) 등 **대기 환경에 관한 항목**, 그리고 마지막으로 소음(Noise), 진동_x(Vib_x), 진동_y(Vib_y) 등 **대기 환경과 무관한 소음 정보**로 구성되어 있다. 본 보고서에서는 이 중 대기환경에 관한 항목의 열만 사용하는데, 이때 기온, 풍향, 풍속, 돌풍 풍향, 돌풍 풍속, 자외선, 흑구 운도 열은 소수점 단위의 실수 값(numeric), 상대 습도, 조도 열은 정수 값(integer)을 가지는 것을 볼 수

있다.

2장. Codebook

2.1 데이터 클렌징

데이터 코드북 작성에 앞서, 데이터 클렌징 작업을 진행하였다. 클렌징 작업에는 Notepad++와 R 두가지 프로그램이 활용되었다.

S-DoT_NATURE_2021.03.01-03.07.csv

```
"기관 명","송신 서버 번호","데이터 번호","모델명","시리얼","구분","초미세먼지(㎍/㎥)","미세먼지(㎍/㎥)","기온(℃)","상대습도( %)",
"서울시","48","1","SDOT001","OC3CL200010","1","2","3","6.7","43","","","","","","0.0","39","","","","
"서울시","48","1","SDOT001","OC3CL200012","1","2","4","5.8","44","","","","","","0.0","60","","","","
"서울시","48","1","SDOT001","OC3CL200011","1","2","3","21.6","88","","","","","","0.0","49","","","","
"서울시","48","1","SDOT001","OC3CL200017","1","1","2","6.4","42","","","","","","0.0","67","","","","
"서울시","48","1","SDOT001","OC3CL200013","1","1","2","5.6","43","","","","","","0.0","48","","","","
"서울시","48","1","SDOT001","OC3CL200016","1","3","5","6.2","41","","","","","","0.0","52","","","","
"서울시","48","1","SDOT001","OC3CL200015","1","2","4","5.5","41","","","","","","0.0","63","","","","
"서울시","48","1","SDOT001","OC3CL200019","1","2","4","7.9","37","","","","","","0.0","63","","","","
"서울시","48","1","SDOT001","OC3CL200014","1","3","4","5.6","42","","","","","","0.0","64","","","","
"서울시","48","1","SDOT001","OC3CL200020","1","0","1","5.9","40","","","","","","","70","","","","
"서울시","48","1","SDOT001","OC3CL200018","1","2","4","5.6","45","","","","","","0.0","53","","","","
"서울시","48","1","SDOT001","OC3CL200022","1","1","2","5.2","41","","","","","","","70","","","","
"서울시","48","1","SDOT001","OC3CL200021","1","2","4","5.2","41","","","","","","","66","","","","
"서울시","48","1","SDOT001","OC3CL200023","1","4","6","6.5","39","","","","","","","65","","","","
```

Figure 1. Original Dataset

우선 Notepad++를 활용하여 한글로 열 이름이 표기되어 있는 1행 하단(2행)에, 영어 열 이름을 추가하였다. 각 행에서 유일하게 “기관 명” 열에 해당하는 “서울시”만 한글로 표기되어 있는데, 해당 “기관 명” 열은 뒤 이은 클렌징 과정에서 곧바로 제거될 예정이기 때문에 별다르게 “서울시”를 영문으로 바꾸는 작업은 하지 않았다.

S-DoT_NATURE_2021.03.01-03.07.csv

```
"기관 명","송신 서버 번호","데이터 번호","모델명","시리얼","구분","초미세먼지(㎍/㎥)","미세먼지(㎍/㎥)","기온(℃)","상대습도( %)"
"Organization","OutserverNum","DataNum","Model","SerialNum","Sortation","MicroPM","PM","Temp","Rel
"서울시","48","1","SDOT001","OC3CL200010","1","2","3","6.7","43","","","","","","0.0","39","","","","
"서울시","48","1","SDOT001","OC3CL200012","1","2","4","5.8","44","","","","","","0.0","60","","","","
"서울시","48","1","SDOT001","OC3CL200011","1","2","3","21.6","88","","","","","","0.0","49","","","","
"서울시","48","1","SDOT001","OC3CL200017","1","1","2","6.4","42","","","","","","0.0","67","","","","
"서울시","48","1","SDOT001","OC3CL200013","1","1","2","5.6","43","","","","","","0.0","48","","","","
"서울시","48","1","SDOT001","OC3CL200016","1","3","5","6.2","41","","","","","","0.0","52","","","","
"서울시","48","1","SDOT001","OC3CL200015","1","2","4","5.5","41","","","","","","0.0","63","","","","
"서울시","48","1","SDOT001","OC3CL200019","1","2","4","7.9","37","","","","","","0.0","63","","","","
"서울시","48","1","SDOT001","OC3CL200014","1","3","4","5.6","42","","","","","","0.0","64","","","","
"서울시","48","1","SDOT001","OC3CL200020","1","0","1","5.9","40","","","","","","","70","","","","
"서울시","48","1","SDOT001","OC3CL200018","1","2","4","5.6","45","","","","","","0.0","53","","","","
"서울시","48","1","SDOT001","OC3CL200022","1","1","2","5.2","41","","","","","","","70","","","","
"서울시","48","1","SDOT001","OC3CL200021","1","2","4","5.2","41","","","","","","","66","","","","
```

Figure 2. Primary Dataset

이후 R 프로그램의 {dplyr} 라이브러리를 이용해, Predictive Analytics에서 Predictor로 활용될 대기환경에 관한 열을 제외하고는 select()함수를 통해 모두 제거하였다. 또한 개별 행의 값이 모두 같거나 고유한 열들도 apply(), select() 함수 등을 활용해 모두 제거하였다.

```
> # data cleansing
> M1 <- select(M_org, -(Noise:MaxVib_z)) #irrelevant columns
> M2 <- select(M1, -(MicroPMrev:Date)) #irrelevant & duplicate columns
> M3 <- select(M2, -SerialNum) #irrelevant columns
> dim(M3)
[1] 118433     16
>
> # remove unique columns
> del_idx <- apply(M3, 2, function(x) ifelse(length(unique(x)) == 1, TRUE, FALSE))
> sum(del_idx)
[1] 5
> col <- names(M3)[!del_idx]
> M4 <- M3 %>% dplyr::select(all_of(col))
> dim(M4)
[1] 118433     11
>
> # remove all-rows-different columns
> del_idx2 <- apply(M4, 2, function(x) ifelse(length(unique(x)) == nrow(M4), TRUE, FALSE))
> sum(del_idx2)
[1] 0
> col <- names(M4)[!del_idx2]
> M5 <- M4 %>% select(all_of(col))
> dim(M5)
[1] 118433     11
```

한편 위 데이터셋의 가장 큰 문제점이 바로 상당한 결측치의 개수와 그 불규칙성이었는데, 이에 대해서는 {mice} 라이브러리를 활용하여 imputation 작업을 수행하였다. 단순히 na.omit()함수를 활용할 경우, 결측치의 불규칙성으로 인해 데이터셋 내의 모든 행이 제거되었기 때문이다.

```
> # imputation (N/A)
> library(mice)
>
> Mz_imp <- M5
> pmiss(Mz_imp)
    MicroPM          PM        Temp  RelHumidity    WindDirec    WindSpeed    GustDirec   GustSpeed Illumination       UVrays     WaterEva
  0.4770630    0.4770630    0.4601758    0.4601758   97.1553537   97.1553537   97.1553537   97.1553537   18.9820405   18.8857835   96.2789088
>
> N_na <- sum(is.na(Mz_imp))
> tot_element <- nrow(Mz_imp) * ncol(Mz_imp)
> na_percent <- (N_na)/(tot_element) * 100
> cat("NA % = ", na_percent, "%\n")
NA % =  47.69478 %
>
> if (na_percent < 0.1) {
+   M6 <- na.omit(Mz_imp)
+ } else {
+   md.pattern(Mz_imp)
+   imp <- mice(Mz_imp, m = 1)
+   densityplot(imp)
+   M6 <- complete(imp, 1)
+ }

 iter imp variable
  1   1  MicroPM  PM  Temp  RelHumidity  WindDirec  WindSpeed  GustDirec  GustSpeed  Illumination  UVrays  WaterEva
  2   1  MicroPM  PM  Temp  RelHumidity  WindDirec  WindSpeed  GustDirec  GustSpeed  Illumination  UVrays  WaterEva
  3   1  MicroPM  PM  Temp  RelHumidity  WindDirec  WindSpeed  GustDirec  GustSpeed  Illumination  UVrays  WaterEva
  4   1  MicroPM  PM  Temp  RelHumidity  WindDirec  WindSpeed  GustDirec  GustSpeed  Illumination  UVrays  WaterEva
  5   1  MicroPM  PM  Temp  RelHumidity  WindDirec  WindSpeed  GustDirec  GustSpeed  Illumination  UVrays  WaterEva
>
> head(M6)
  MicroPM PM Temp RelHumidity WindDirec WindSpeed GustDirec GustSpeed Illumination UVrays WaterEva
1       2  3  6.7          43       126       0.6     136.0       1.3          2525      0      4.0
2       2  4  5.8          44        11       0.4      18.7       1.1             0      0      3.3
3       2  3 21.6          88        73       2.8     141.0       4.5             0      0     17.7
4       1  2  6.4          42       110       1.3      92.9       2.7          2637      0      5.1
5       1  2  5.6          43        33       1.1     108.0       1.8          1454      0      3.3
6       3  5  6.2          41        21       1.4     127.0       3.5          1534      0      3.4
> dim(M6)
[1] 118433     11
>
```

마지막으로는 예측 분석을 통해 알아보고자 하는 값인 초미세먼지(MicroPM)를 농도에 따라 “Good”과 “Bad”의 두 가지 등급으로 분류하였다. 두 가지 등급의 분류 기준은 환경부의 미세먼지 농도별 예보 등급 기준을 따랐다. 실제 환경부의 미세먼지 농도별 등급 기준을 살펴보면 “좋음”, “보통”, “나쁨”, “매우 나쁨”의 총 4가지 등급으로 구분되어 있으나 본 프로젝트에서는 2가지 등급으로만 구분하였다. 그 이유는 뒤에서 살펴볼 Predictive Analytics의 Preprocessing 과정에 언급된다. 한편 등급별 자세한

내용은 이어지는 코드북에 설명되어 있다.

```
> # mutate
> M7 <- dplyr::mutate(M6, Y = ifelse( M6$MicroPM <= 35, "Good", "Bad"))
> Mz <- select(M7, -(MicroPM:PM))
> head(Mz)
  Temp RelHumidity WindDirec WindSpeed GustDirec GustSpeed Illumination UVrays WaterEva    Y
1  6.7          43       126       0.6     136.0       1.3         2525      0      4.0 Good
2  5.8          44        11       0.4      18.7       1.1            0      0      3.3 Good
3 21.6          88        73       2.8     141.0       4.5            0      0     17.7 Good
4  6.4          42       110       1.3      92.9       2.7         2637      0      5.1 Good
5  5.6          43        33       1.1     108.0       1.8         1454      0      3.3 Good
6  6.2          41        21       1.4     127.0       3.5         1534      0      3.4 Good
> dim(Mz)
[1] 118433     10
```

위와 같은 클렌징 과정을 모두 거친 데이터의 형태는 다음과 같다.

```
> head(Mz)
  Temp RelHumidity WindDirec WindSpeed GustDirec GustSpeed Illumination UVrays WaterEva    Y
1  6.7          43        94       1.3      89.1       2.2            2      0      4.0 Good
2  5.8          44         6       0.6      65.8       1.9            0      0      3.3 Good
3 21.6          88       277       1.0     300.0       1.4            0      0     17.7 Good
4  6.4          42       267       0.2     164.0       1.8            0      0      5.1 Good
5  5.6          43        51       0.1      50.5       1.5         2580      0      3.3 Good
6  6.2          41       201       1.0     195.0       2.4            0      0      3.4 Good
> dim(Mz)
[1] 118433     10
```

2.2 코드북

다음은 각 Column Name에 대한 설명과, 예측하고자 하는 초미세먼지(MicroPM)의 등급에 대한 정보가 담긴 코드북이다.

2.2.1 Description of Variables

Variable	Description	Value
Temp	기온(°C)	6.7
RelHumidity	상대 습도(%)	43
WindDirec	풍향(°)	94
WindSpeed	풍속(m/s)	1.3
GustDirec	돌풍 풍향(°)	89.1
GustSpeed	돌풍 풍속(m/s)	2.2
Illumination	조도(lux)	2
UVrays	자외선(UVI)	0
WaterEva	흑구 운도(°C)	4.0
Y	초미세먼지 농도에 따른 등급	Good

2.2.2 Classification of MicroPM

MicroPM	Y	Description
35	Good	환자군에게 만성 노출 시 경미한 영향이 유발될 수 있는 수준
35 +	Bad	환자군 및 민감군에게 노출 시 유해한 영향 유발, 일반인도 건강상의 불쾌감 혹은 악영향이 유발될 수 있는 수준

3장. Descriptive Analytics

Predictive Analytics를 진행하기에 앞서, Descriptive Analytics를 통한 과거 데이터의 시각화를 진행하였다. 이때 데이터셋은 최종 클렌징 데이터 Mz가 아닌 그 바로 전 데이터 M7을 활용하였다. 두 데이터셋의 차이점은 연속적 데이터인 MicroPM과 PM을 포함하는지 여부인데, Mz는 모델링 과정에서 연산을 줄이기 위해 class variable과 내용이 동일한 MicroPM과 PM 열을 제거했다. 그러나 본 Descriptive Analytics는 그 분석의 특성 상 연속적인 본래의 데이터 MicroPM과 PM 열을 활용하는 것이, 미세먼지와 타 환경정보와의 상관관계를 파악하기에 적절하기 때문에 M7 데이터를 활용했다. 먼저 {ggplot2} 라이브러리를 사용하여 density graph(밀도 그래프)와 point graph(산점도)를 통해, predictor가 되는 주요 환경정보별 초미세먼지(MicroPM)의 분포를 살펴보았다.

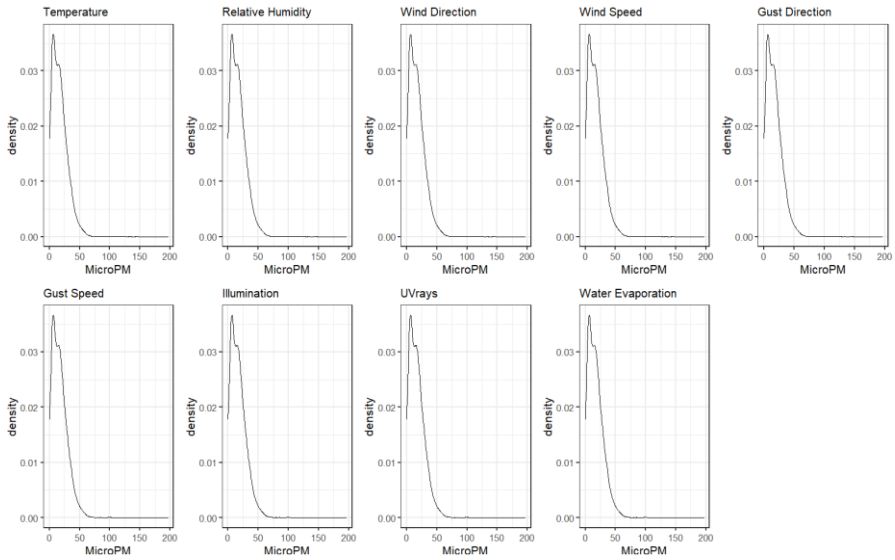

Figure 3. MicroPM Concentration by Environment Factors (Density Graph)

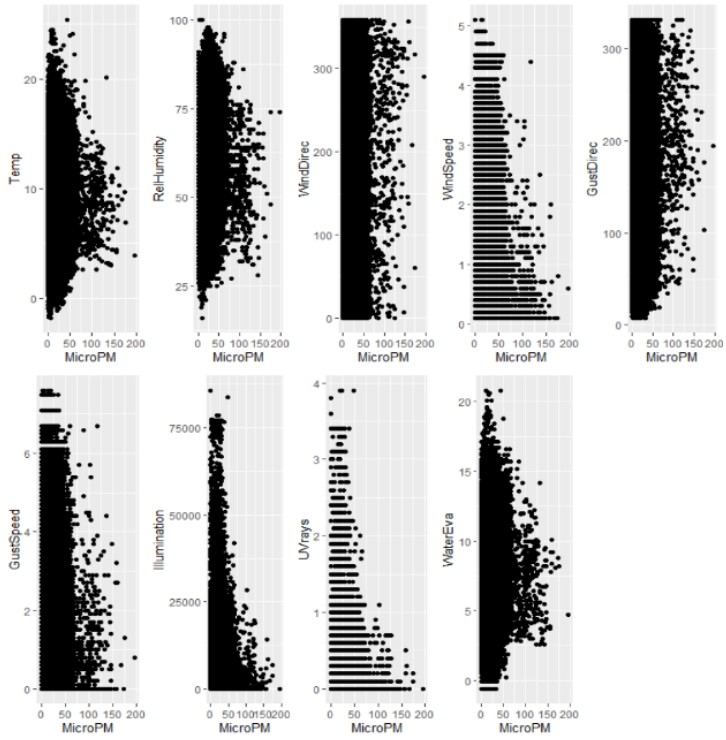

Figure 4. MicroPM Concentration by Environment Factors (Point Graph)

밀도 그래프 분석 결과, 3월 첫째 주(원본 데이터가 수집된 기간)의 대기환경 요소에서는 전반적으로 낮은 초미세먼지 수치를 보인다. 각 대기환경 요소에 따른 초미세먼지 분포가 유사한 형태를 보이는 것 또한 알 수 있다. 한편 아래 산점도를 통해서는, 앞선 그래프보다 더 다양한 결과를 얻을 수 있다. 특히 기온, 상대습도, 풍향, 돌풍속도, 조도, 흑구 운도의 대기환경 요소에 대해서는, 특정 구간의 분포 내에서 초미세먼지 농도가 높게 나타나는 것을 알 수 있다.

다음으로 {corrplot} 라이브러리를 활용하여 correlation graph를 통해 초미세먼지 및 미세먼지 수치와 각 대기환경 요소 간의 상관계수를 그래프로 살펴보았는데, 그 결과는 다음과 같다.

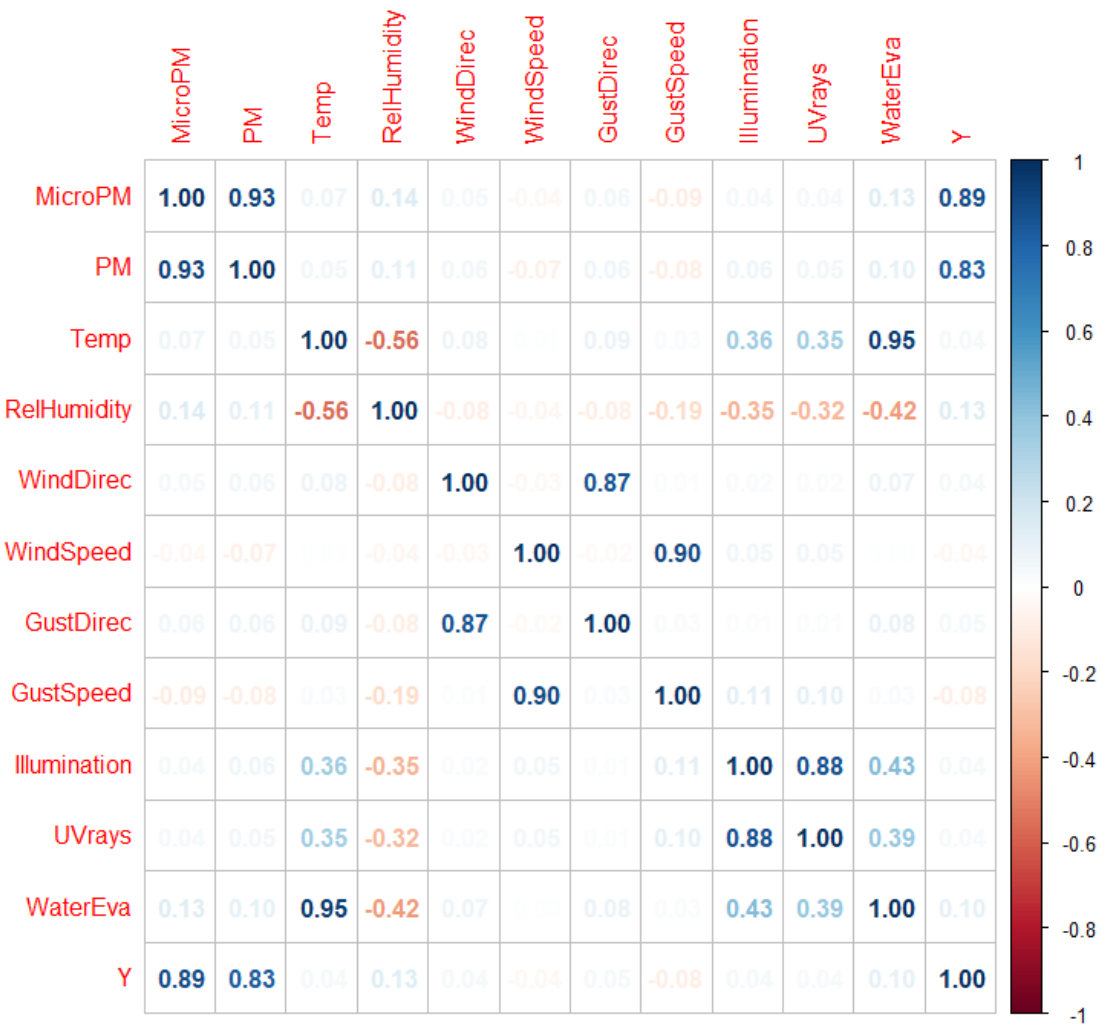

Figure 5. Correlation between Each Attribute

MicroPM을 중심으로 상관계수 분포를 살펴봤을 때, 다른 기후환경 요소들과 유의미한 상관관계는 보이지 않는 것으로 나타난다.

4장. Predictive Analytics

4.1 Preprocessing

Predictive Analytics 모델을 만들기 위해 데이터 프로세싱 작업을 시작하였다. 우선 분산이 0이거나 0에 가까운 변수가 있는지 nearZeroVar을 이용해 살펴보았으나 모두 FALSE값을 얻어 nzv값을 제거할 필요는 없었다.

```
> nzv <- nearZeroVar(Mz[, 1:9], saveMetrics = TRUE)
> nzv
             freqRatio percentUnique zeroVar   nzv
Temp          1.027027     0.21531161   FALSE FALSE
RelHumidity   1.040471     0.06923746   FALSE FALSE
WindDirec     3.474716     0.30228061   FALSE FALSE
WindSpeed     2.371197     0.03968489   FALSE FALSE
GustDirec     3.057459     0.57585301   FALSE FALSE
GustSpeed     2.296190     0.05910515   FALSE FALSE
Illumination  9.886382    14.78641933   FALSE FALSE
UVrays        8.106739     0.03208565   FALSE FALSE
WaterEva      1.101237     0.16718313   FALSE FALSE
```

그러나 데이터셋에서 Class Variable에 해당하는 Y가 특정 class로 편중되어 있는 모습을 보였다. 이러한 데이터 불균형 문제를 해결하기 위해, {ROSE} 라이브러리를 활용하여 SMOTE 방법을 통해 데이터를 resampling 해주었다. 앞서 잠시 언급하였듯, 본 프로젝트에서 class variable Y (미세먼지 등급)를 두 가지만 구분한 이유도 이 SMOTE 방법을 사용하기 위해서였다. Under Sampling 혹은 Over Sampling에 비해 SMOTE의 성능이 우수한데, Y가 3가지 이상의 class로 나뉠 경우 이 SMOTE 방법 적용이 불가능했기 때문이다.

```
> # re-sampling due to the class imbalance
> library(ROSE)
Loaded ROSE 0.0-4

> table(Mz$Y)

   Bad   Good
 10603 107830
> prop.table(table(Mz$Y))

       Bad       Good
0.08952741 0.91047259
>
> M <- ROSE(Y ~., data = Mz, seed = 1)$data
> table(M$Y)

  Bad  Good
59202 59231
```

한편 이후의 작업은 preProcess 함수를 활용해 모델링 단계에서 데이터에 대한 scaling 작업을 수행하려 하였다. 그러나 이후 모델링에서 *preProc = c("center", "scale")*을 함께 입력했을 때, 계속 오류가 발생하여 이 과정은 불가피하게 생략하게 되었다.

4.2 Data Splitting

데이터를 Training Data와 Testing Data로 나누는 작업을 수행하였다. 데이터는 Training Data 80%, Testing Data 20%로 나누었다.

4.3 Modeling

모델링은 {caret} 라이브러리 내의 Random Forest, R Partition, IDA(Linear Discriminant Analysis) 등의 기법을 사용해 진행하였다. Random Forest의 샘플링 방식으로는 repeated cross validation을 사용하였다.

4.4 Model Evaluation

4.4.1 Random Forest

```
> my_trControl1 <- trainControl(method = "repeatedcv",
+                               number = 5,
+                               repeats = 3)
>
> model_rf <- caret::train(Y ~., method = "rf", data = train_data, trControl = my_trControl1)
> model_rf
Random Forest

94746 samples
    9 predictor
    2 classes: 'Bad', 'Good'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 75797, 75797, 75797, 75796, 75797, 75796, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.6646226  0.3292300
  5     0.6624167  0.3248238
  9     0.6598309  0.3196544

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```
>
> test_rf <- test_rocFUN(model_rf, test_data)
Confusion Matrix and Statistics

          Reference
Prediction  Bad Good
      Bad  7835 3848
      Good 4108 7896

               Accuracy : 0.6641
                 95% CI : (0.6581, 0.6701)
    No Information Rate : 0.5042
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3283

 Mcnemar's Test P-Value : 0.003688

            Sensitivity : 0.6560
            Specificity : 0.6723
         Pos Pred Value : 0.6706
         Neg Pred Value : 0.6578
             Prevalence : 0.5042
         Detection Rate : 0.3308
   Detection Prevalence : 0.4932
      Balanced Accuracy : 0.6642

       'Positive' Class : Bad

[[1]]
[1] 0.729366

[[1]]
[1] 0.729366
```

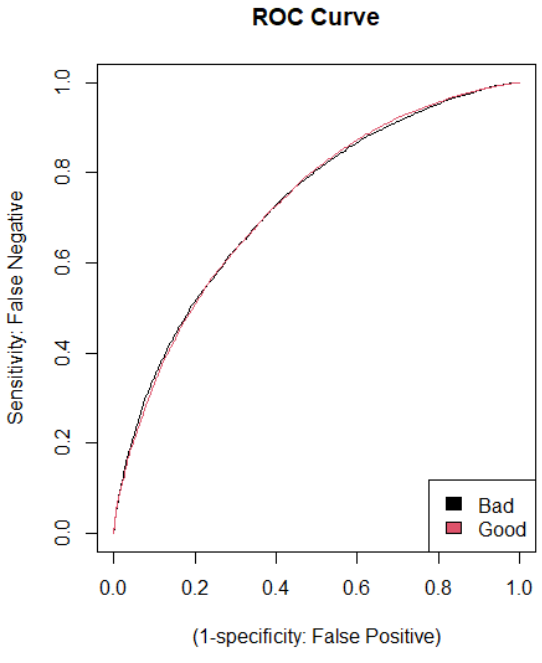

Figure 6. ROC Curve for Random Forest

4.4.2 R Partition

```
> model_rpart <- caret::train(Y ~., method = "rpart", data = train_data)
> model_rpart
CART

94746 samples
    9 predictor
    2 classes: 'Bad', 'Good'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 94746, 94746, 94746, 94746, 94746, 94746, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.01125712  0.6213307  0.24296459
  0.02102245  0.6065091  0.21258641
  0.19259824  0.5422817  0.08521066

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01125712.
>
```

```
> test_rpart <- test_rocFUN(model_rpart, test_data)
Confusion Matrix and Statistics

          Reference
Prediction  Bad Good
      Bad  8456 5505
      Good 3487 6239

               Accuracy : 0.6204
                 95% CI : (0.6142, 0.6266)
    No Information Rate : 0.5042
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2396

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

            Sensitivity : 0.7080
            Specificity : 0.5312
         Pos Pred Value : 0.6057
         Neg Pred Value : 0.6415
             Prevalence : 0.5042
         Detection Rate : 0.3570
   Detection Prevalence : 0.5894
      Balanced Accuracy : 0.6196

       'Positive' Class : Bad

[[1]]
[1] 0.6474666

[[1]]
[1] 0.6474666
```

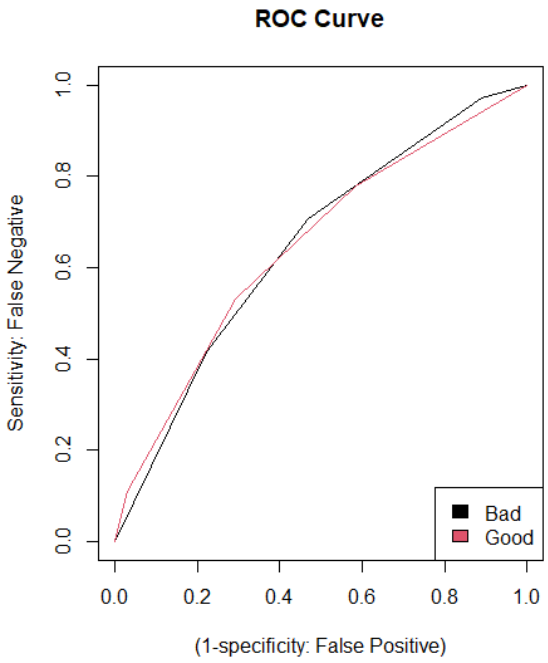

Figure 7. ROC Curve for R Partition

4.4.3 IDA(Linear Discriminant Analysis)

```
> model_lda <- caret::train(as.factor(Y) ~ ., method = "lda", data = train_data)
> model_lda
Linear Discriminant Analysis

94746 samples
    9 predictor
    2 classes: 'Bad', 'Good'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 94746, 94746, 94746, 94746, 94746, 94746, ...
Resampling results:

  Accuracy   Kappa
  0.6326992  0.2653258
```

```
> test_lda <- test_rocFUN(model_lda, test_data)
Confusion Matrix and Statistics

          Reference
Prediction  Bad Good
      Bad  7215 4005
      Good 4728 7739

               Accuracy : 0.6313
                 95% CI : (0.6251, 0.6375)
    No Information Rate : 0.5042
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.263

 Mcnemar's Test P-Value : 1.11e-14

            Sensitivity : 0.6041
            Specificity : 0.6590
         Pos Pred Value : 0.6430
         Neg Pred Value : 0.6208
             Prevalence : 0.5042
         Detection Rate : 0.3046
   Detection Prevalence : 0.4737
      Balanced Accuracy : 0.6315

       'Positive' Class : Bad

[[1]]
[1] 0.6775577

[[1]]
[1] 0.6775577
```

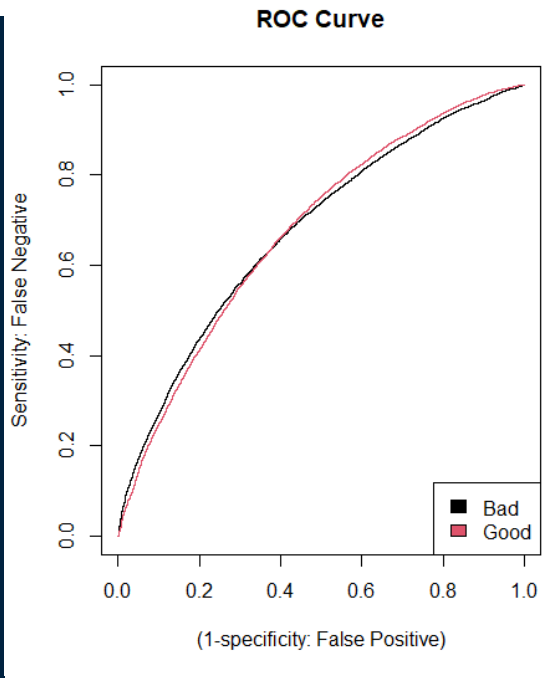

Figure 8. ROC Curve for IDA

본래 위 세가지의 알고리즘 외에 Support Vector Machine with Linear 알고리즘도 활용하려 하였으나, 필자의 컴퓨터로는 모델링 시간이 오래 소요되어(6시간 이상 소요) 그에 대한 정확도는 확인하지 못했다. 따라서 위 세가지의 모델만을 두고 그 테스트 결과를 비교해본 결과, Random Forest가 비교적 높은 Accuracy를 보이는 것을 알 수 있었다.

0.6204(**R Partition**) < 0.6313(**IDA**) < 0.6641(**Random Forest**)

5장. 결언

본 보고서는 Descriptive Analytics를 통해 특정 대기환경 요소에 따라 초미세먼지 농도가 보이는 패턴을 읽으려 시도하였고, Predictive Analytics를 통해 특정 대기 환경이 조성되었을 때의 초미세먼지 농도 등급을 예측해보려 하였다. 그 결과 기온이 5℃와 15℃ 사이일 경우 초미세먼지 농도가 높게 나타나는 패턴을 발견하는 등, 다양한 대기환경 요소에서 초미세먼지 농도 패턴을 목격할 수 있었다. 또한 초미세먼지 농도 그 자체가 미세먼지 농도를 제외한 기타 대기환경 요소와는 크게 상관관계를 보이지는 않는 것으로 나타났다. 한편, Random Forest와 R Partition 등의 기법으로 모델링을 통해 초미세먼지 농도 등급으로 “Good” 혹은 “Bad”으로 예측할 수 있는 모델을 작성하였다.

그러나 초미세먼지 농도는 본 보고서에서 분석한 변수 외에도 시간대, 지역, 시내 교통량, 자국 및 주변국의 환경 정책 등 수많은 요소로부터 영향을 받는다. 그러므로 이 분석은 초미세먼지 농도와 관련된 중요 변수들을 모두 포함하지 못했다는 한계를 지닌다.

그러나 본 분석은 기상 관측 분야, 더 넓게는 환경 분야 전반에 대한 기계 학습의 적용 가능성을 보여주었다는 점에서 큰 의의가 있다. 미세먼지에 대한 사안은 국가적 차원뿐만 아니라 시민적 차원에서도 매우 관심 있게 주목하는 주제 중 하나이다. 본 보고서를 발판 삼아 미세먼지 농도에 대한 보다 완전한 기계 학습 예측 모델이 완성된다면, 이를 통해 국내 미세먼지 예방책이 더욱 발전될 수 있을 것이다.

■ 참조 문헌

- Handling Class Imbalance with R. https://goodtogreate.tistory.com/entry/Handling-Class-Imbalance-with-R-and-Caret-An-introduction
- 환경부 수도권대기환경청, 미세먼지 바로 알기. https://www.me.go.kr/mamo/web/index.do?menuId=16201