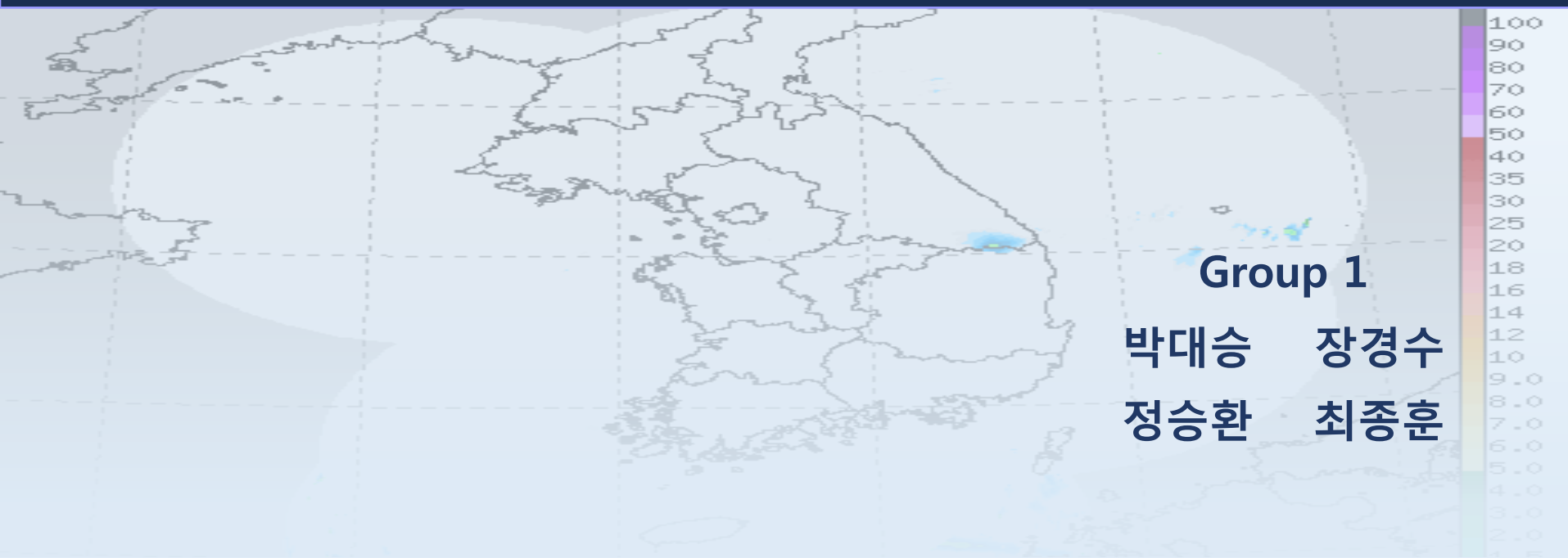


빅데이터를 활용한 기상 현상 & 계절 변화 예측



INDEX

- I. 배경과 목적**
- II. Team 소개 및 역할**
- III. 프로젝트 진행 일정**
- IV. 데이터 수집 및 전처리**
- V. 분석 / 평가**
- VI. 결론**
- VII. Lessons**
- VIII. 시연**

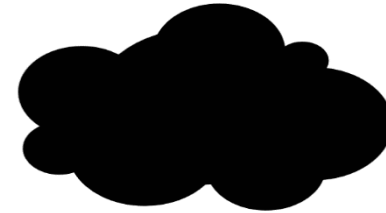
I. 배경 및 목적



I. 배경 및 목적

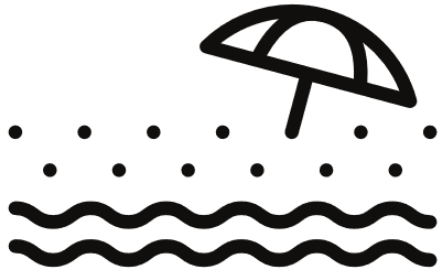


☔ ?



“오늘 구름 낀 것 봐. **비가 오려나?**”
정말일까?

I. 배경 및 목적



“온난화에 의한 **4계절** → **2계절**”
정말일까?

I. 연구목적

기상 관련 빅데이터 활용,

기상 현상(비,눈)

: 영향을 끼치는 변수 분석

4계절 → 2계절

:진위 여부 판단 및 예상 날짜 예측

II. 팀원 소개 및 역할

Group 1

박대승

- 데이터 조사 및 수집
- 데이터 전처리
- 발표자료 작성

장경수

- 총괄
- 데이터 분석
- 발표자료 작성

정승환

- 데이터 수집 / 전처리
- 데이터 분석 / 시각화
- 발표자료 작성

최종훈

- 데이터 조사 및 수집
- 데이터 분석
- 발표자료 작성

III. 프로젝트 진행 일정

➤ 프로젝트 진행 일정 : 5/9 ~ 5/23 (총 15일)

	5/9	5/10	5/11	5/12	5/13	5/16	5/17	5/18	5/19	5/20	5/23
주제 선정											
자료 수집											
데이터 전처리											
분석											
평가/수정											
PT 자료 작성											



IV. 데이터 수집 및 전처리

RAW DATA 정보

- 수집 경로 : 웹 사이트 (www.wunderground.com)
- 내용 : 서울 지역 기상 정보 (1996/10/01 ~ 2016/05/09)
- 세부 데이터 소개

[시계열 정보] KST(일자)

[온도 정보] Max.TemperatureC / Mean.TemperatureC / Min.TemperatureC

[이슬점 정보] MaxDes.PointC / MeanDew.PointC / Min.DewpointC

[기압 정보] Max.Sea.Level.PressurehPa / Mean.Sea.Level.PressurehPa / Min.Sea.Level.PressurehPa

[가시도 정보] Max.가시도Km / Mean.가시도Km / Min.가시도km

[풍속 정보] Max.Wind.SpeedKm.h / Mean.Wind.SpeedKm.h / Max.Gust.SpeedKm.h

[강수량 정보] 강수.확률mm

[구름량 정보] CloudCover

[기상 현상] Events

[풍향 정보] WindDirDegrees

IV. 데이터 수집 및 전처리

➤ Column 추가 생성

- + 연/월/일 별 column
- + 온도 기준 계절 구분 column
- + 날짜 기준 계절 구분 column
- + 기상 현상(이벤트)별 column

➤ 변수 정의

```
# data.frame':          7227 obs. of  29 variables:
# $ Date                : POSIXlt / 날짜
# $ TempMax              : int    / 최고온도
# $ TempMean             : int    / 평균온도
# $ TempMin              : int    / 최저온도
# $ DewPointMax          : int    / 최고이슬점
# $ DewPointMean         : int    / 평균이슬점
# $ DewPointMin          : int    / 최저이슬점
# $ HumidityMax           : int    / 최고습도
# $ HumidityMean         : int    / 평균습도
# $ HumidityMin          : int    / 최저습도
# $ SeaLevelPressureMax  : int    / 최고해수면수압
# $ SeaLevelPressureMean : int    / 평균해수면수압
# $ SeaLevelPressureMin  : int    / 최저해수면수압
# $ VisibilityMax        : int    / 최대가시도
```

➤ 계절별 날짜 구분 기준

(* 기상청 자료 참조)

- + 봄 : 3/12 ~ 5/29
- + 여름 : 5/30 ~ 9/22
- + 가을 : 9/23 ~ 11/24
- + 겨울 : 11/25 ~ 3/11

➤ 계절별 온도 구분 기준

계절	평균기온	최저기온	최고기온
봄	5℃ 이상	0℃ 이상	-
여름	20℃ 이상	-	25℃ 이상
가을	20℃ 이하	-	25℃ 이하
겨울	5℃ 이하	0℃ 이하	-

V. 분석 및 평가 – 기상 Event

➤ 최적의 분류 모델 찾기

- k-Nearest Neighbors
- Naive Bayes
- Decision Tree
- Random Forest
- Support Vector Machine
- Artificial Neural Network

➤ 분류 모델 선정 기준

- 분류 정확도
- 중요 변수 파악 가/부

V. 분석 및 평가 – 기상 Event – 분류 전처리

➤ 기상 Event 요인별 분류모델 생성

- x, y 변수 정의
 - x : 기상 데이터 (온도, 습도, 이슬점, 구름 정도 등)
 - y : 기상 Event (비, 눈, 안개)
- 훈련데이터와 검정데이터 생성

```
# 불필요 변수 제거
```

```
w.na = na.omit(w)
str(w.na)
names(w)
w.e = w.na[c(-1,-19,-21,-22,-24,-25,-26)]
names(w.e)
```

```
# y변수 분리
```

```
w.rain = as.factor(w.e$E_rain)
w.rain <- factor(w.rain, levels = c('0','1'), labels = c('No','Yes'))
w.snow = as.factor(w.e$E_snow)
w.snow <- factor(w.snow, levels = c('0','1'), labels = c('No','Yes'))
w.mist = as.factor(w.e$E_mist)
w.mist <- factor(w.mist, levels = c('0','1'), labels = c('No','Yes'))
w.rain = data.frame(w.rain)
w.snow = data.frame(w.snow)
w.mist = data.frame(w.mist)
str(w.rain)
str(w.snow)
str(w.mist)
```

```
# x변수 분리
```

```
names(w.e)
w.x = w.e[c(-20,-21,-22)]
str(w.x)
```

```
# x변수 정규화
```

```
summary(w.x)
w.xn = as.data.frame(lapply(w.x, scale))
summary(w.xn)
```

```
# 훈련데이터와 검정데이터 생성
```

```
set.seed(1)
idx = sample(1:nrow(w.xn), 0.7*nrow(w.xn))
w.xn_train = w.xn[idx, ]
w.xn_test = w.xn[-idx, ]
w.rain_train = w.rain[idx, ]
w.rain_test = w.rain[-idx, ]
w.snow_train = w.snow[idx, ]
w.snow_test = w.snow[-idx, ]
w.mist_train = w.mist[idx, ]
w.mist_test = w.mist[-idx, ]
```

```
# 데이터 합치기
```

```
w.xn.rain_train= cbind(w.xn_train,w.rain_train)
w.xn.rain_test= cbind(w.xn_test,w.rain_test)
w.xn.snow_train= cbind(w.xn_train,w.snow_train)
w.xn.snow_test= cbind(w.xn_test,w.snow_test)
w.xn.mist_train= cbind(w.xn_train,w.mist_train)
w.xn.mist_test= cbind(w.xn_test,w.mist_test)
```

```
# 정규화 안한 데이터 생성 및 훈련 데이터 생성
```

```
w.x.rain = cbind(w.x,w.rain)
w.x.snow = cbind(w.x,w.snow)
w.x.mist = cbind(w.x,w.mist)
```

```
set.seed(1)
```

```
idx = sample(1:nrow(w.xn), 0.7*nrow(w.xn))
w.x.rain_train = w.x.rain[idx, ]
w.x.rain_test = w.x.rain[-idx, ]
w.x.snow_train = w.x.snow[idx, ]
w.x.snow_test = w.x.snow[-idx, ]
w.x.mist_train = w.x.mist[idx, ]
w.x.mist_test = w.x.mist[-idx, ]
```

V. 분석 및 평가 – 기상 Event – kNN

➤ kNN 분석

- Event별 비율 테이블

```
> prop.table(table(w.rain))  
w.rain  
      No      Yes  
0.6903461 0.3096539
```

```
> prop.table(table(w.snow))  
w.snow  
      No      Yes  
0.941864 0.058136
```

```
> prop.table(table(w.mist))  
w.mist  
      No      Yes  
0.8917729 0.1082271
```

- 최적 파라미터 선정

```
> w.rain_df[order(w.rain_df$w.rain_re, decreasing = T),]
```

```
cnt w.rain_re  
10 10 0.8381386  
7 7 0.8371270  
13 13 0.8366211
```

```
> w.snow_df[order(w.snow_df$w.snow_re, decreasing = T),]
```

```
cnt w.snow_re  
16 16 0.9590288  
13 13 0.9580172  
15 15 0.9580172
```

```
> w.mist_df[order(w.mist_df$w.mist_re, decreasing = T),]
```

```
cnt w.mist_re  
6 6 0.9312089  
18 18 0.9296915  
17 17 0.9291856
```

- 최적 모델 선정

```
> w.rain_p <- knn(w.xn_train,w.xn_test,w.rain_train, k=10)
```

```
> w.rain_t = table(w.rain_p, w.rain_test)
```

```
> w.rain_t
```

```
      w.rain_test  
w.rain_p No Yes  
No 1224 207  
Yes 128 418
```

```
> (w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.xn_test)  
[1] 0.8305513
```

```
> w.snow_p <- knn(w.xn_train,w.xn_test,w.snow_train, k=16)
```

```
> w.snow_t = table(w.snow_p, w.snow_test)
```

```
> w.snow_t
```

```
      w.snow_test  
w.snow_p No Yes  
No 1840 80  
Yes 5 52
```

```
> (w.snow_t[1,1]+w.snow_t[2,2])/nrow(w.xn_test)  
[1] 0.9570056
```

```
> w.mist_p <- knn(w.xn_train,w.xn_test,w.mist_train, k=6)
```

```
> w.mist_t = table(w.mist_p, w.mist_test)
```

```
> w.mist_t
```

```
      w.mist_test  
w.mist_p No Yes  
No 1740 97  
Yes 40 100
```

```
> (w.mist_t[1,1]+w.mist_t[2,2])/nrow(w.xn_test)  
[1] 0.9307031
```

☞ 비 예측 모델 83% / 눈 예측 모델 96% / 안개 예측 모델 93%

V. 분석 및 평가 – 기상 Event – NB

➤ Naive Bayes 분석

- 모델 생성(분류기)

```
w.rain_NB = naiveBayes(w.rain_train~ . ,data=w.xn.rain_train)
w.rain_NB
w.snow_NB = naiveBayes(w.snow_train~ . ,data=w.xn.snow_train)
w.snow_NB
w.mist_NB = naiveBayes(w.mist_train~ . ,data=w.xn.mist_train)
w.mist_NB
```

- 모델 평가(예측기)

```
> w.rain_t = table(w.rain_p, w.xn.rain_test$w.rain_test)
> w.rain_t

w.rain_p    No  Yes
No    1027  159
Yes     325  466

> w.snow_t = table(w.snow_p, w.xn.snow_test$w.snow_test)
> w.snow_t

w.snow_p    No  Yes
No    1482    5
Yes     363  127

> w.mist_t = table(w.mist_p, w.xn.mist_test$w.mist_test)
> w.mist_t

w.mist_p    No  Yes
No    1607   38
Yes     173  159

> (w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.xn.rain_test)
[1] 0.7551846
> (w.snow_t[1,1]+w.snow_t[2,2])/nrow(w.xn.snow_test)
[1] 0.8138594
> (w.mist_t[1,1]+w.mist_t[2,2])/nrow(w.xn.mist_test)
[1] 0.8932726
```

☞ 비 예측 모델 76% / 눈 예측 모델 81% / 안개 예측 모델 89%

V. 분석 및 평가 – 기상 Event – DT

➤ Decision Tree

- 모델 생성

```
> w.rain_DT = rpart(w.rain ~ . ,data=w.x.rain_train)
> w.rain_DT
n= 4611
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 4611 1415 No (0.69312514 0.30687486)
 2) cloudCover< 6.5 3531 541 No (0.84678561 0.15321439)
   4) cloudCover< 4.5 2210 119 No (0.94615385 0.05384615) *
   5) cloudCover>=4.5 1321 422 No (0.68054504 0.31945496)
   10) windSpeedMax< 23.5 1049 281 No (0.73212583 0.26787417) *
   11) windSpeedMax>=23.5 272 131 Yes (0.48161765 0.51838235)
    22) DewPointMean< -6.5 43 4 No (0.90697674 0.09302326) *
    23) DewPointMean>=-6.5 229 92 Yes (0.40174672 0.59825328)
     46) visibilityMin>=7.5 36 6 No (0.83333333 0.16666667) *
     47) visibilityMin< 7.5 193 62 Yes (0.32124352 0.67875648) *
 3) cloudCover>=6.5 1080 206 Yes (0.19074074 0.80925926)
   6) DewPointMax< -0.5 65 18 No (0.72307692 0.27692308) *
   7) DewPointMax>=-0.5 1015 159 Yes (0.15665025 0.84334975) *
```

```
> w.snow_DT = rpart(w.snow ~ . ,data=w.x.snow_train)
> w.snow_DT
n= 4611
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 4611 251 No (0.945564953 0.054435047)
 2) TempMax>=5.5 3805 35 No (0.990801577 0.009198423) *
 3) TempMax< 5.5 806 216 No (0.732009926 0.267990074)
   6) cloudCover< 3.5 392 23 No (0.941326531 0.058673469) *
   7) cloudCover>=3.5 414 193 No (0.533816425 0.466183575)
   14) visibilityMin>=3.5 172 44 No (0.744186047 0.255813953) *
   15) visibilityMin< 3.5 242 93 Yes (0.384297521 0.615702479)
    30) visibilityMax< 9.5 106 39 No (0.632075472 0.367924528)
     60) windSpeedMax< 20 73 19 No (0.739726027 0.260273973)
      120) HumidityMin< 64.5 48 7 No (0.854166667 0.145833333) *
       121) HumidityMin>=64.5 25 12 No (0.520000000 0.480000000) *
```

```
> w.mist_DT = rpart(w.mist ~ . ,data=w.x.mist_train)
> w.mist_DT
n= 4611
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 4611 516 No (0.88809369 0.11190631)
 2) visibilityMin>=0.5 4218 181 No (0.95708867 0.04291133) *
 3) visibilityMin< 0.5 393 58 Yes (0.14758270 0.85241730) *
```

V. 분석 및 평가 – 기상 Event – DT

➤ Decision Tree

- 모델 평가(예측기)

```
w.rain_p = predict(w.rain_DT, w.x.rain_test, type='class')
w.snow_p = predict(w.snow_DT, w.x.snow_test, type='class')
w.mist_p = predict(w.mist_DT, w.x.mist_test, type='class')
w.rain_t = table(w.rain_p, w.x.rain_test$w.rain)
w.rain_t
w.snow_t = table(w.snow_p, w.x.snow_test$w.snow)
w.snow_t
w.mist_t = table(w.mist_p, w.x.mist_test$w.mist)
w.mist_t
(w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.x.rain_test)
(w.snow_t[1,1]+w.snow_t[2,2])/nrow(w.x.snow_test)
(w.mist_t[1,1]+w.mist_t[2,2])/nrow(w.x.mist_test)
```

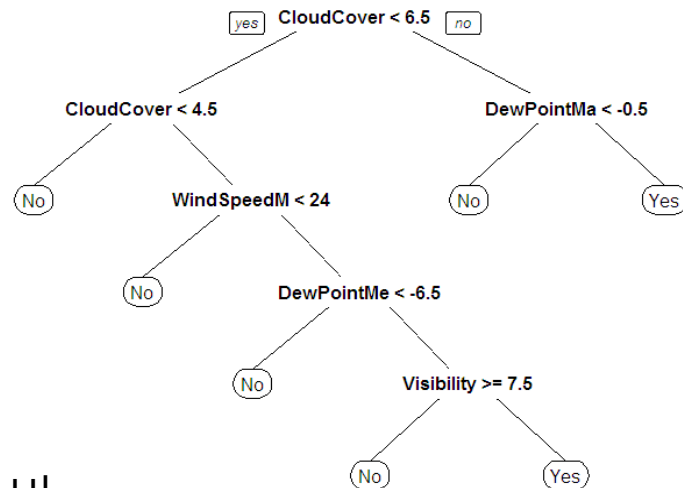
```
> (w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.x.rain_test)
[1] 0.8406677
> (w.snow_t[1,1]+w.snow_t[2,2])/nrow(w.x.snow_test)
[1] 0.9590288
> (w.mist_t[1,1]+w.mist_t[2,2])/nrow(w.x.mist_test)
[1] 0.9514416
```

☞ 비 예측 모델 84% / 눈 예측 모델 96% / 안개 예측 모델 95%

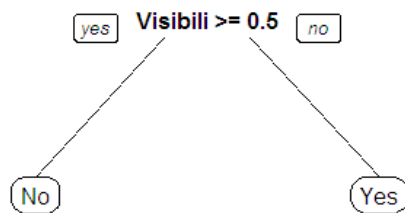
V. 분석 및 평가 – 기상 Event – DT

➤ Decision Tree

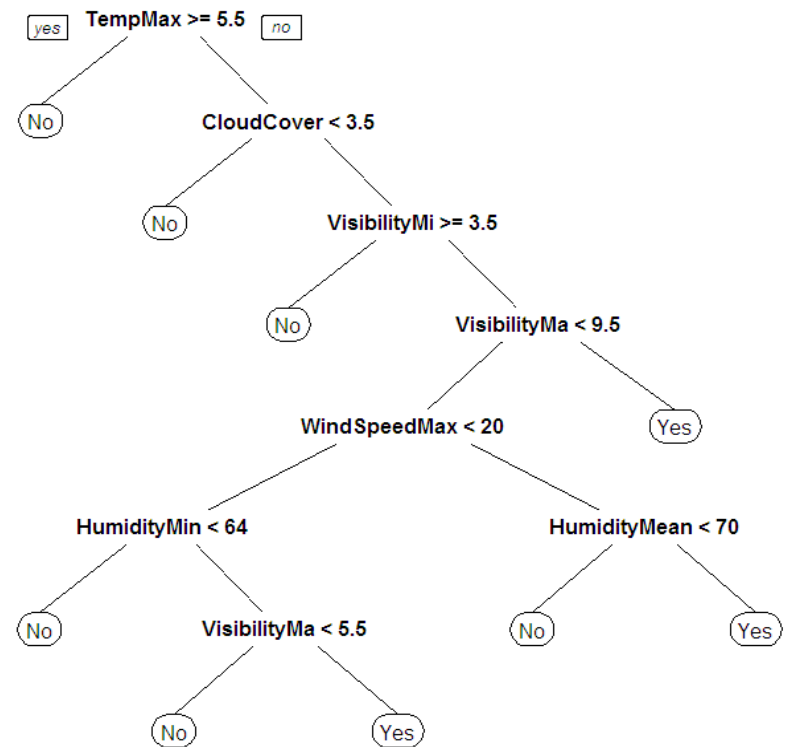
- 모델 시각화(중요 변수 평가)



비



안개



번개

V. 분석 및 평가 – 기상 Event – RF

➤ Random Forest

- 모델 생성

```
# model 생성
w.rain_RF = randomForest(w.rain ~ . , data=w.x.rain, ntree=300, mtry=4, importance = T)
w.rain_RF
w.snow_RF = randomForest(w.snow ~ . , data=w.x.snow, ntree=300, mtry=4, importance = T)
w.snow_RF
w.mist_RF = randomForest(w.mist ~ . , data=w.x.mist, ntree=300, mtry=4, importance = T)
w.mist_RF
```

Call:

```
randomForest(formula = w.rain ~ ., data = w.x.rain, ntree = 300, mtry = 4, importance = T)
```

Type of random forest: classification

Number of trees: 300

No. of variables tried at each split: 4

OOB estimate of error rate: 13.28%

Confusion matrix:

	No	Yes	class.error
No	4232	316	0.06948109
Yes	559	1481	0.27401961

최적 파라미터 도출

```
ntree <- c(300, 400, 500, 600, 700)
mtry <- c(1:5)
param <- data.frame(n=ntree, m=mtry)
param
nt = numeric()
mt = numeric()
ra = numeric()
cnt = 1
# rain
for(i in param$n){
  for(j in param$m){
```

V. 분석 및 평가 – 기상 Event – RF

➤ Random Forest

- 최적 모델 분류 정확도 판단

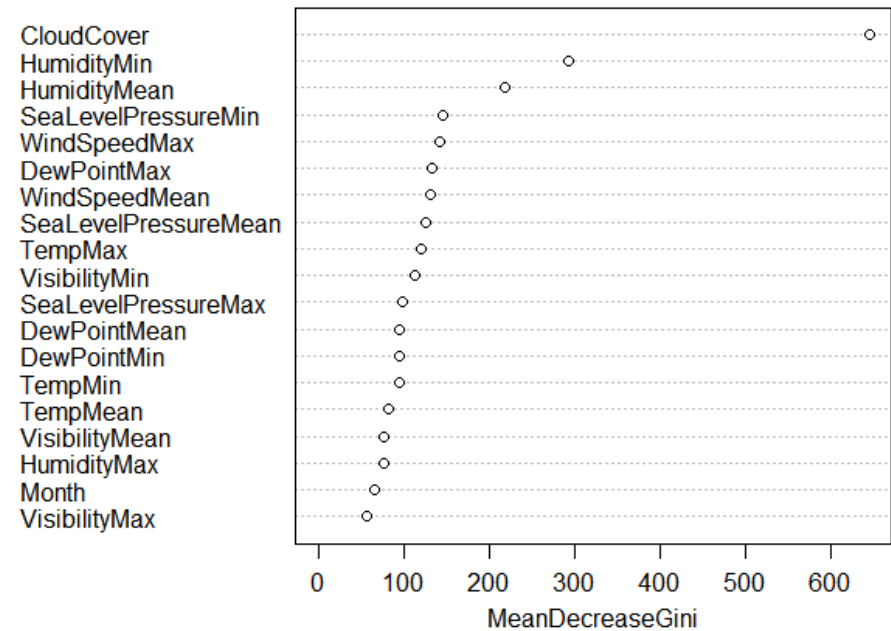
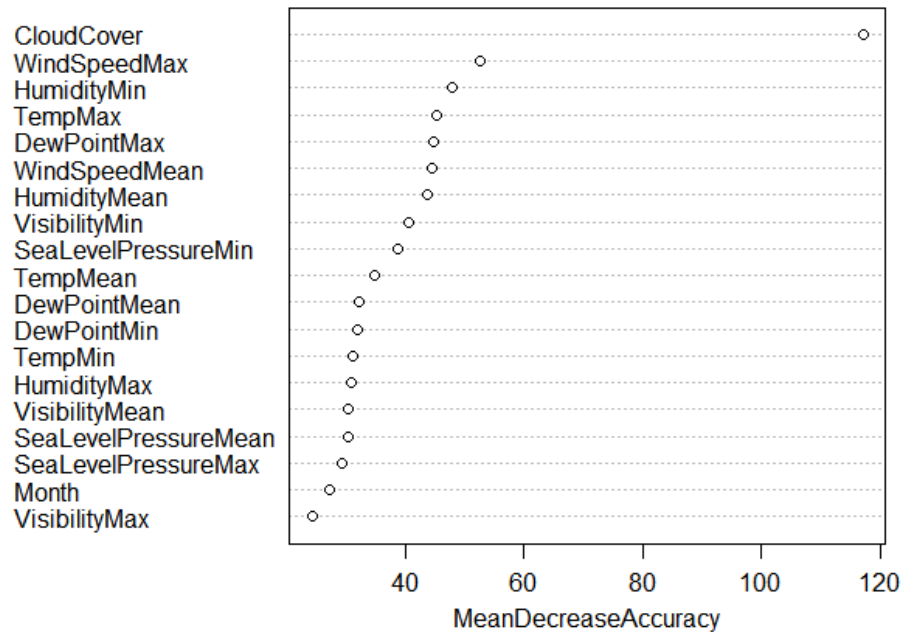
```
> w.rain_RF = randomForest(w.rain ~ . ,data=w.x.rain,ntree=700, mtry=3, importance = T)
> w.snow_RF = randomForest(w.snow ~ . ,data=w.x.snow,ntree=300, mtry=3, importance = T)
> w.mist_RF = randomForest(w.mist ~ . ,data=w.x.mist,ntree=300, mtry=3, importance = T)
> (w.rain_RF$confusion[1,1]+w.rain_RF$confusion[2,2])/sum(w.rain_RF$confusion[, -3])
[1] 0.8658166
> (w.snow_RF$confusion[1,1]+w.snow_RF$confusion[2,2])/sum(w.snow_RF$confusion[, -3])
[1] 0.9679721
> (w.mist_RF$confusion[1,1]+w.mist_RF$confusion[2,2])/sum(w.mist_RF$confusion[, -3])
[1] 0.9511233
```

☞ 비 예측 모델 87% / 눈 예측 모델 97% / 안개 예측 모델 95%

V. 분석 및 평가 – 기상 Event – RF

➤ Random Forest 중요 변수 분석 (비)

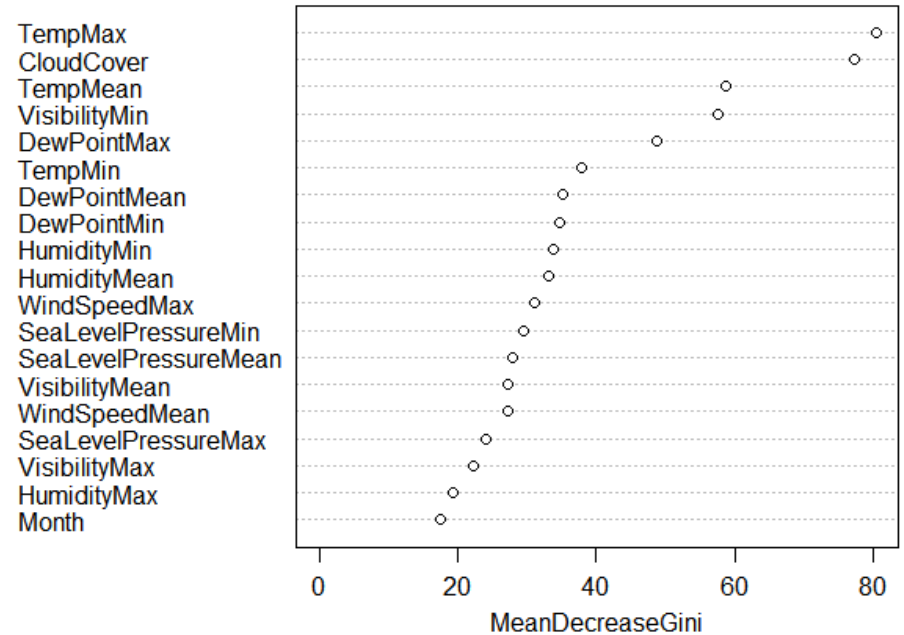
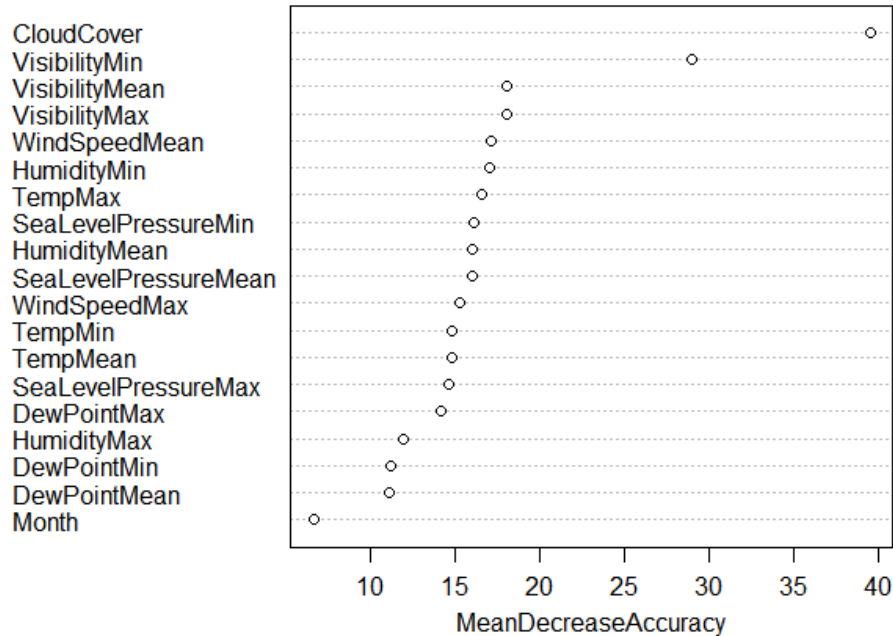
w.rain_RF



V. 분석 및 평가 – 기상 Event – RF

➤ Random Forest 중요 변수 분석 (눈)

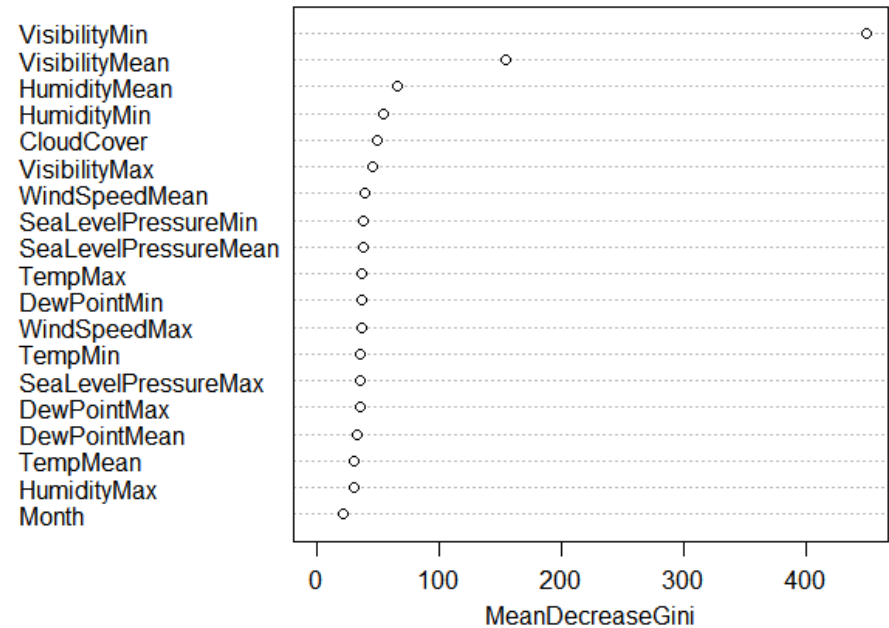
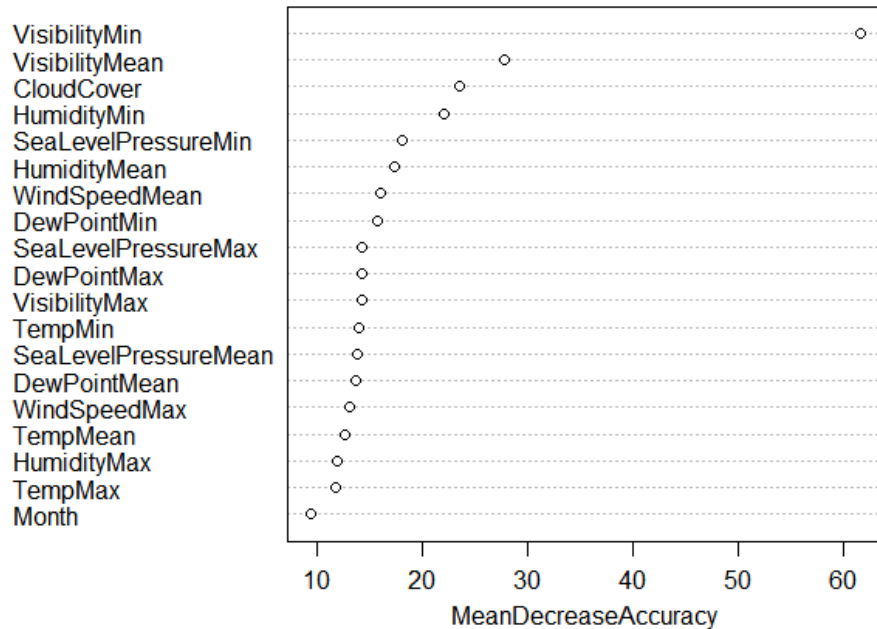
w.snow_RF



V. 분석 및 평가 – 기상 Event – RF

➤ Random Forest 중요 변수 분석 (안개)

w.mist_RF



V. 분석 및 평가 – 기상 Event – SVM

➤ SVM

- 모델 생성

```
# model 생성
w.rain_SVM_r = svm(w.rain ~ . ,data=w.x.rain, kernel='radial')
w.rain_SVM_l = svm(w.rain ~ . ,data=w.x.rain, kernel='linear')
w.snow_SVM_r = svm(w.snow ~ . ,data=w.x.snow, kernel='radial')
w.snow_SVM_l = svm(w.snow ~ . ,data=w.x.snow, kernel='linear')
w.mist_SVM_r = svm(w.mist ~ . ,data=w.x.mist, kernel='radial')
w.mist_SVM_l = svm(w.mist ~ . ,data=w.x.mist, kernel='linear')
```

- 모델 평가 (예 : RAIN)

```
> w.rain_p_r = predict(w.rain_SVM_r, w.x.rain_test, type='class')
> w.rain_p_l = predict(w.rain_SVM_l, w.x.rain_test, type='class')
> w.rain_t_r = table(w.rain_p_r, w.x.rain_test$w.rain)
> w.rain_t_r

w.rain_p_r   No   Yes
      No 1264  167
      Yes   88  458
> (w.rain_t_r[1,1]+w.rain_t_r[2,2])/nrow(w.x.rain_test)
[1] 0.8710167
> w.rain_t_l = table(w.rain_p_l, w.x.rain_test$w.rain)
> w.rain_t_l

w.rain_p_l   No   Yes
      No 1212  155
      Yes  140  470
> (w.rain_t_l[1,1]+w.rain_t_l[2,2])/nrow(w.x.rain_test)
[1] 0.850784
```

☞ 비 예측 모델 86%(Radial)
85%(Linear)

☞ 눈 예측 모델 96%(Radial)
96%(Linear)

☞ 안개 예측 모델 96%(Radial)
95%(Linear)

V. 분석 및 평가 – 기상 Event – ANN

➤ ANN

- 최적 size 찾기

```
# 최적의 size 찾기
cnt = 1:15
result = numeric()
# rain
for(i in cnt){
  w.rain_NN = nnet(w.rain_train ~ . ,data=w.xn.rain_train, size=i)
  w.rain_p = predict(w.rain_NN, w.xn.rain_test, type='class')
  w.rain_t = table(w.rain_p, w.xn.rain_test$w.rain_test)
  if(dim(w.rain_t)[1]==2){
    re = (w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.xn.rain_test)
    cat('\nsize:',i)
    print(w.rain_t)
    cat('정분류율:',re,'\n\n')
    result[i]=re
  }else{
    result[i]=NA
  }
}
```

최적 size
Rain : 2
Snow : 8
Mist : 2

- 모델 생성

```
w.rain_NN = nnet(w.rain_train ~ . ,data=w.xn.rain_train, size=2)
w.snow_NN = nnet(w.snow_train ~ . ,data=w.xn.snow_train, size=8)
w.mist_NN = nnet(w.mist_train ~ . ,data=w.xn.mist_train, size=2)
```


V. 분석 및 평가 – 기상 Event – ANN

➤ ANN

- 모델 평가

```
> w.rain_p = predict(w.rain_NN, w.xn.rain_test, type='class')
> w.rain_t = table(w.rain_p, w.xn.rain_test$w.rain_test)
> w.rain_t

w.rain_p   No   Yes
      No 1200  156
      Yes  152 469
> (w.rain_t[1,1]+w.rain_t[2,2])/nrow(w.xn.rain_test)
[1] 0.8442084
> w.snow_p = predict(w.snow_NN, w.xn.snow_test, type='class')
> w.snow_t = table(w.snow_p, w.xn.snow_test$w.snow_test)
> w.snow_t

w.snow_p   No   Yes
      No 1800   52
      Yes   45  80
> (w.snow_t[1,1]+w.snow_t[2,2])/nrow(w.xn.snow_test)
[1] 0.9509358
> w.mist_p = predict(w.mist_NN, w.xn.mist_test, type='class')
> w.mist_t = table(w.mist_p, w.xn.mist_test$w.mist_test)
> w.mist_t

w.mist_p   No   Yes
      No 1742   65
      Yes   38 132
> (w.mist_t[1,1]+w.mist_t[2,2])/nrow(w.xn.mist_test)
[1] 0.9479009
```

비 예측 모델: 84%
눈 예측 모델: 95%
안개 예측 모델: 95%

V. 분석 및 평가 – 기상 Event – 결론

➤ 분석 방법 비교

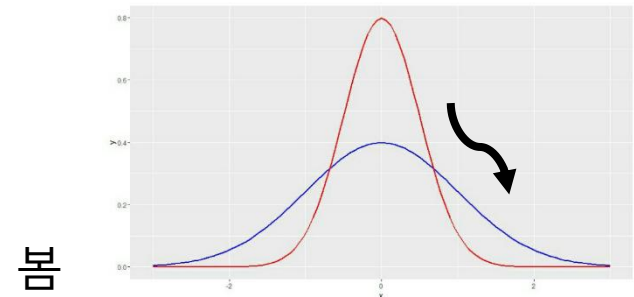
분석법	비	눈	안개	중요도 판단
kNN	83	96	93	-
NB	76	81	89	-
DT	84	96	95	가능
RF	87	97	95	가능
SVM(Rad)	86	96	96	-
ANN	84	95	95	-
최적안	RF	RF	SVM	

 RandomForest 분류법 선택

V. 분석 및 평가 – 2계절

➤ 연도별 온도/기간에 따른 계절 구분 후 빈도 분석

- 만약, 두 데이터의 겹치지 않는 경향이 연도 별로 **커진다**면?
 - ex) **봄**(일정기간, 3~5월)의 온도 특성이 점차 여름/겨울화 되고있다.
 - 기온 변화로 인해 봄/가을이 없어지고 있다.



➤ 과거의 일정 시기 vs 현재의 일정 시기 비교(table, t-test)

- 만약, 비교 후 **유의미한 온도차이**가 있다면?
 - 예년에 비해 분명한 온도 변화가 일어나고 있다.

V. 분석 및 평가 - 2계절 - 빈도분석

➤ 데이터 전처리(연도 별분류)

```
# 연도별 데이터 분류
w.1996 = subset(w,w$Year==1996)
w.1997 = subset(w,w$Year==1997)
w.1998 = subset(w,w$Year==1998)
w.1999 = subset(w,w$Year==1999)
w.2000 = subset(w,w$Year==2000)
w.2001 = subset(w,w$Year==2001)
w.2002 = subset(w,w$Year==2002)
w.2003 = subset(w,w$Year==2003)
w.2004 = subset(w,w$Year==2004)
w.2005 = subset(w,w$Year==2005)
w.2006 = subset(w,w$Year==2006)
w.2007 = subset(w,w$Year==2007)
w.2008 = subset(w,w$Year==2008)
w.2009 = subset(w,w$Year==2009)
w.2010 = subset(w,w$Year==2010)
w.2011 = subset(w,w$Year==2011)
w.2012 = subset(w,w$Year==2012)
w.2013 = subset(w,w$Year==2013)
w.2014 = subset(w,w$Year==2014)
w.2015 = subset(w,w$Year==2015)
w.2016 = subset(w,w$Year==2016)
```

➤ 테이블 작성 + 일치율 비교

```
table(w.2016$Season_temp,w.2016$Season_date)

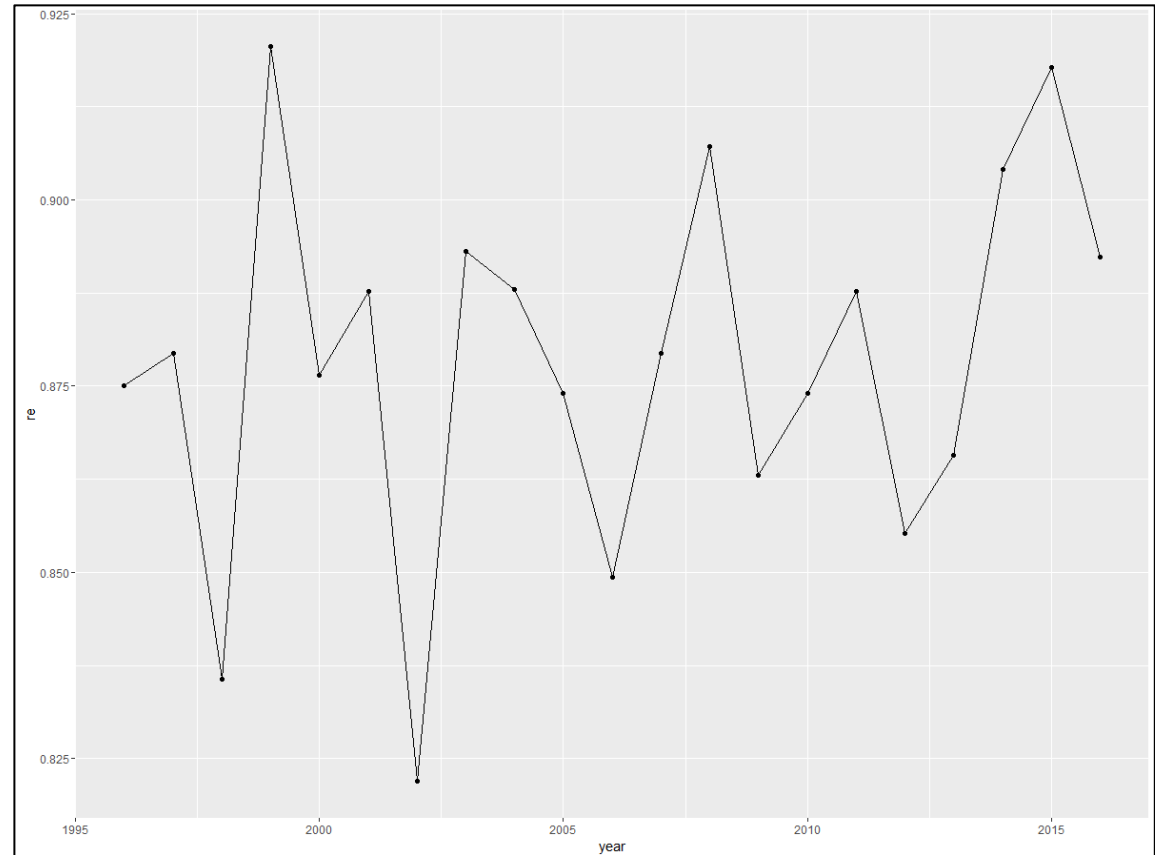
a = list(w.1996,w.1997,w.1998,w.1999,w.2000,
         w.2001,w.2002,w.2003,w.2004,w.2005,
         w.2006,w.2007,w.2008,w.2009,w.2010,
         w.2011,w.2012,w.2013,w.2014,w.2015,w.2016)
cnt = 1:21
re = numeric()
year = numeric()
for(i in cnt){
  year[i] = i+1995
  cat('Year:',year[i])
  t = table(a[[i]]$Season_temp,a[[i]]$Season_date)
  print(t)
  re[i] = (t[1,1]+t[2,2]+t[3,3]+t[4,4])/nrow(a[[i]])
}
re

season_df = data.frame(year,re)
season_df
```

V. 분석 및 평가 - 2계절 - 빈도분석

➤ 연도별 일치율 비교

	year	re
1	1996	0.8750000
2	1997	0.8794521
3	1998	0.8356164
4	1999	0.9205479
5	2000	0.8764706
6	2001	0.8876712
7	2002	0.8219178
8	2003	0.8931507
9	2004	0.8879781
10	2005	0.8739726
11	2006	0.8493151
12	2007	0.8794521
13	2008	0.9071038
14	2009	0.8630137
15	2010	0.8739726
16	2011	0.8876712
17	2012	0.8551913
18	2013	0.8657534
19	2014	0.9041096
20	2015	0.9178082
21	2016	0.8923077



👉 불규칙한 변화 = 계절별 줄어듦의 추세가 불분명하다 (없다)

V. 분석 및 평가 - 2계절 - 봄 온도 비교



온전한 데이터를 가진 양 극단 년도의 평균 온도 비교

1997년 봄 온도 <-> 2015년 봄 온도
1997년 가을 온도 <-> 2015년 가을 온도

V. 분석 및 평가 – 2계절 – 봄 온도 비교

➤ 분포 모양 검정 (봄)

```
> var.test(w.spring_1997$TempMean, w.spring_2015$TempMean, paired = T)
```

```
F test to compare two variances
```

```
data: w.spring_1997$TempMean and w.spring_2015$TempMean
F = 0.87589, num df = 78, denom df = 78, p-value = 0.5598
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5601092 1.3696921
sample estimates:
ratio of variances
 0.8758865
```

P-Value = 0.5598 > 0.05 → 분포의 차이는 없다.

V. 분석 및 평가 – 2계절 – 봄 온도 비교

➤ 가설 검정 (양측 검정)

귀무가설 : 두 년도의 봄 평균온도의 차이가 없다.

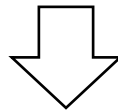
대립가설 : 두 년도의 봄 평균온도의 차이가 있다.

```
> t.test(w.spring_1997$TempMean, w.spring_2015$TempMean, paired = T)
```

Paired t-test

```
data: w.spring_1997$TempMean and w.spring_2015$TempMean
t = -3.5931, df = 78, p-value = 0.0005699
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.1442253 -0.6152684
sample estimates:
mean of the differences
 -1.379747
```

P-Value = 0.0005699 < 0.05



즉, 귀무 가설 기각

두 년도의 봄 평균 온도의 차이가 있다

V. 분석 및 평가 – 2계절 – 봄 온도 비교

➤ 가설 검정 (단측 검정)

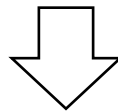
가설 : 1997년의 봄 평균기온이 2015년의 봄 평균기온보다 낮다.

```
> t.test(w.spring_1997$TempMean, w.spring_2015$TempMean, paired=TRUE,  
+ alter="less", conf.int=TRUE, conf.level=0.95)
```

Paired t-test

```
data: w.spring_1997$TempMean and w.spring_2015$TempMean  
t = -3.5931, df = 78, p-value = 0.000285  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf -0.7405367  
sample estimates:  
mean of the differences  
-1.379747
```

P-Value = 0.000285 < 0.05



즉, 귀무 가설 기각

1997년의 봄 평균 기온이 2015년보다 낮다고 할 수 있다

V. 분석 및 평가 – 2계절 – 가을 온도 비교

➤ 분포 차이 검정 (가을)

```
> var.test(w.fall_1997$TempMean, w.fall_2015$TempMean, paired = T)
```

```
F test to compare two variances
```

```
data: w.fall_1997$TempMean and w.fall_2015$TempMean  
F = 0.98997, num df = 62, denom df = 62, p-value = 0.9685  
alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:  
 0.5989845 1.6361662  
sample estimates:  
ratio of variances  
 0.9899688
```

P-Value = 0.9685 > 0.05 ➔ 분포의 차이는 없다.

V. 분석 및 평가 – 2계절 – 가을 온도 비교

➤ 가설 검정 (양측 검정)

귀무가설 : 두 년도의 가을 평균온도의 차이가 없다.

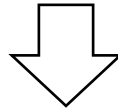
대립가설 : 두 년도의 가을 평균온도의 차이가 있다.

```
> t.test(w.fall_1997$TempMean, w.fall_2015$TempMean, paired = T)
```

Paired t-test

```
data: w.fall_1997$TempMean and w.fall_2015$TempMean  
t = -8.5682, df = 62, p-value = 4.111e-12  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
-4.228460 -2.628683  
sample estimates:  
mean of the differences  
-3.428571
```

P-Value = $4.111e-12 < 0.05$



즉, 귀무 가설 기각

두 년도의 가을 평균 온도의 차이가 있다

V. 분석 및 평가 – 2계절 – 가을 온도 비교

➤ 가설 검정 (단측 검정)

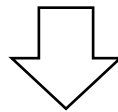
가설 : 1997년의 봄 평균기온이 2015년의 가을 평균기온보다 낮다.

```
> t.test(w.fall_1997$TempMean, w.fall_2015$TempMean, paired=TRUE,  
+ alter="less", conf.int=TRUE, conf.level=0.95)
```

Paired t-test

```
data: w.fall_1997$TempMean and w.fall_2015$TempMean  
t = -8.5682, df = 62, p-value = 2.055e-12  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf -2.760399  
sample estimates:  
mean of the differences  
-3.428571
```

P-Value = $2.055e-12 < 0.05$



즉, 귀무 가설 기각

1997년의 가을 평균 기온이 2015년보다 낮다고 할 수 있다

IV. 분석 및 평가 – 2계절 – 검정

➤ 가설 검정 (단측 검정)

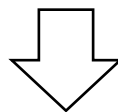
가설 : 1997년의 봄평균 기온이 2015년의 가을평균 기온보다 낮다.

```
> t.test(w.fall_1997$TempMean, w.fall_2015$TempMean, paired=TRUE,  
+ alter="less", conf.int=TRUE, conf.level=0.95)
```

Paired t-test

```
data: w.fall_1997$TempMean and w.fall_2015$TempMean  
t = -8.5682, df = 62, p-value = 2.055e-12  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf -2.760399  
sample estimates:  
mean of the differences  
-3.428571
```

P-Value = $2.055e-12 < 0.05$

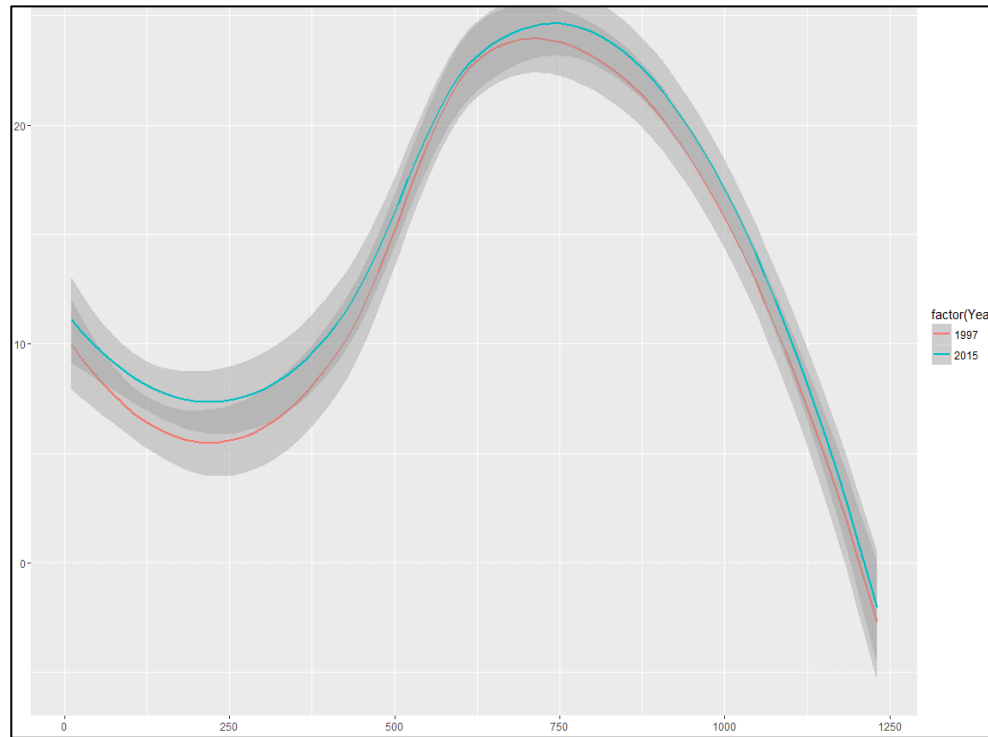


즉, 귀무 가설 기각

1997년의 가을 평균 기온이 2015년보다 낮다고 할 수 있다

V. 분석 및 평가 – 2계절 – 결론

- 1997년의 봄/가을 **평균 기온**이 2015년의 기온보다 **낮다**.



- 앞선 빈도 분석 결과와 다르게 의미 있는 결과를 보여주므로 주어진 DATA 외의 **변수를 새로이 고려**하여 연구를 지속할 필요가 있다.

VI. 결론

결론 1. 4계절 → 2계절

- 기상 데이터를 이용하여 '봄과 가을이 짧아진다'에 대한 의미 있는 결과를 확보할 수 **없었다**.
- 이것에는 분석 방법의 부족함이나 **다른 변동 요인**을 고려하지 못했을 가능성을 배제할 수 없다.

결론 2. 날씨에 영향을 끼치는 요소

- 비는 구름량, 바람, 습도가 큰 영향을 끼친다.
- 눈은 온도와 구름량이 큰 영향을 끼치는 요소이다.
- 안개는 가시성에만 영향을 받았으므로 의미가 미약하다.

VII. Lessons

Lesson 1.

- 기상은 매우 주기적으로 변화하나, 변화의 정도에 있어서 예측 가능한 경향을 보이지 않았다.
- 따라서 통찰을 위해선 위도 경도, 고도, 구름색 등 다양한 변수를 추가적으로 고려하여 판단해야 할 필요가 있다.

Lesson 2.

- 분류 정확도 판단을 명확히 하기위해 1) 파라미터의 정규화 및 2) 다각도의 분류 방법 비교를 수행했지만, 더 정확한 접근을 위해서는 명확한 변수 분석과 관련 요인의 파악이 필요하다.

VIII. 시연

Subject 1. Random Forest

Subject 2. 평균 기온 차이 비교 시각화