**코로나 감염자 dataset을 활용한, 사망여부 & 생존일수 예측**

목차

[**분석 목표** 3](#_Toc142519598)

[**데이터 전처리** 3](#_Toc142519599)

[1. 사용 데이터 3](#_Toc142519600)

[2. 데이터 구성 3](#_Toc142519601)

[3. 추가 Column 생성 4](#_Toc142519602)

[4. 데이터 전처리 – Missing Value 제거 4](#_Toc142519603)

[5. 데이터 전처리 – “is\_dead == 1” (사망) 인 레코드 취합 5](#_Toc142519604)

[6. 데이터 전처리 – 미사용 변수 제거 5](#_Toc142519605)

[7. 데이터 전처리 – Train / Test 파일로 분리 6](#_Toc142519606)

[8. 데이터 전처리 – 클래스 불균형 해소를 위한 Oversampling 6](#_Toc142519607)

[9. 최종 데이터 정리 7](#_Toc142519608)

[**분석 방식** 7](#_Toc142519609)

[1. 데이터 읽어오기 + 형 변환 7](#_Toc142519610)

[2. 사망여부 예측 모델 학습 7](#_Toc142519611)

[3. 생성한 모델을 이용하여 Test 데이터 사망여부 예측 수행 8](#_Toc142519612)

[4. 예측한 test 데이터의 정확도 측정 8](#_Toc142519613)

[5. 정확도 상세 분석 – True-positive, True-negative, False-positive, False-negative rate 측정 8](#_Toc142519614)

[6. 생존일수 예측 모델 학습 9](#_Toc142519615)

[7. 생성한 모델을 이용하여 Test 데이터 생존일수 예측 수행 9](#_Toc142519616)

[**모델 별 분석** 10](#_Toc142519617)

[1. Tree 10](#_Toc142519618)

[2. rpart 13](#_Toc142519619)

[3. C4.5 15](#_Toc142519620)

[4. C5.0 17](#_Toc142519621)

[5. chaid 19](#_Toc142519622)

[6. DT-ensemble 21](#_Toc142519623)

[7. Random Forest 23](#_Toc142519624)

[8. Boosting 26](#_Toc142519625)

[9. MBR 28](#_Toc142519626)

[10. Neural network 29](#_Toc142519627)

[11. Regression 31](#_Toc142519628)

[12. Logistic Regression 34](#_Toc142519629)

[**분석결과** 37](#_Toc142519630)

[1. 사망여부 예측 37](#_Toc142519631)

[2. 생존일수 예측 38](#_Toc142519632)

[**한계점 및 개선사항** 39](#_Toc142519633)

[1. Oversampling 관련 데이터 생성 문제 39](#_Toc142519634)

# **분석 목표**

1. 코로나 환자 데이터를 이용한, 사망여부에 큰 영향을 미치는 변수 도출

2. 코로나 환자 데이터를 이용한, 생존일수에 큰 영향을 미치는 변수 도출

3. Train 데이터 통한 모델 학습 및 해당 모델 통한 Test 데이터 결과 예측

4. 각 모델의 예측 정확도 측정

# **데이터 전처리**

## 1. 사용 데이터

Kaggle / covidrisk dataset / 레코드 수 : 566602 개

<https://www.kaggle.com/datasets/neerajonline/covidrisk?select=Covid+Patient+list-+Population+Data.csv>

## 2. 데이터 구성

|  |  |  |  |
| --- | --- | --- | --- |
| 변수명 | 설명 | 데이터 형 | 비고 |
| Id | 개인 식별용 id | Chr |  |
| Sex | 성별 | Factor |  |
| Patient\_type | 내원 여부 | Factor |  |
| Entry\_date | 자료 입력일자 | Chr |  |
| Date\_symptoms | 증상 발현일자 | Chr |  |
| Date\_died | 사망일자 | Chr | 생존 시, 9999-99-99 |
| Intubated | 인공호흡기 사용여부 | Factor |  |
| Pneumonia | 폐렴 여부 | Factor |  |
| Age | 연령 | Numeric |  |
| Pregnancy | 임신 여부 | Factor |  |
| Diabetes | 당뇨 여부 | Factor |  |
| Copd | 폐질환 여부 | Factor |  |
| Asthma | 천식 여부 | Factor |  |
| inmsupr | 면역 억제 여부 | Factor |  |
| hypertension | 고혈압 여부 | Factor |  |
| other\_disease | 기타 기저질환 여부 | Factor |  |
| cardiovascular | 심장 질환 여부 | Factor |  |
| obesity | 비만 여부 | Factor |  |
| renal\_chronic | 신장 질환 여부 | Factor |  |
| tobacco | 흡연 여부 | Factor |  |
| contact\_other\_covid | 밀접 접촉 여부 | Factor |  |
| covid\_res | 검사 방법 | Factor | 1,2,3 으로 구성 |
| icu | 중환자실 입원여부 | Factor |  |

## 3. 추가 Column 생성

분석 편의성을 위해, 아래 2개의 Column 을 추가

|  |  |  |  |
| --- | --- | --- | --- |
| 변수명 | 설명 | 데이터 형 | 비고 |
| Is\_dead | 사망 여부 | Factor | Date\_died != “9999-99-99” |
| Day\_cnt | 생존일수 | Numeric | Date\_died – date\_symptoms |

## 4. 데이터 전처리 – Missing Value 제거

레코드 중, Missing value를 가지고 있는 record를 제외

covid\_original\_without\_missing = covid\_original[

(covid\_original$intubated == 1 | covid\_original$intubated == 2)

& (covid\_original$pneumonia == 1 | covid\_original$pneumonia == 2)

& (covid\_original$pregnancy == 1 | covid\_original$pregnancy == 2)

& (covid\_original$diabetes == 1 | covid\_original$diabetes == 2)

& (covid\_original$copd == 1 | covid\_original$copd == 2)

& (covid\_original$asthma == 1 | covid\_original$asthma == 2)

& (covid\_original$inmsupr == 1 | covid\_original$inmsupr == 2)

& (covid\_original$hypertension == 1 | covid\_original$hypertension == 2)

& (covid\_original$other\_disease == 1 | covid\_original$other\_disease == 2)

& (covid\_original$cardiovascular == 1 | covid\_original$cardiovascular == 2)

& (covid\_original$obesity == 1 | covid\_original$obesity == 2)

& (covid\_original$renal\_chronic == 1 | covid\_original$renal\_chronic == 2)

& (covid\_original$tobacco == 1 | covid\_original$tobacco == 2)

& (covid\_original$contact\_other\_covid == 1 | covid\_original$contact\_other\_covid == 2)

& (covid\_original$icu == 1 | covid\_original$icu == 2)

,];

제외 시, 레코드 수 : 23158 개

## 5. 데이터 전처리 – “is\_dead == 1” (사망) 인 레코드 취합

레코드 수 : 4020개

covid\_original\_dead\_without\_missing =  
 covid\_original\_without\_missing[covid\_original\_without\_missing$is\_dead==1,];

## 6. 데이터 전처리 – 미사용 변수 제거

|  |  |
| --- | --- |
| 제거변수명 | 제거 사유 |
| Id | 모델링에서 미사용 |
| Entry\_date | 모델링에서 미사용 |
| Date\_symptoms | 모델링에서 미사용 |
| Date\_died | 모델링에서 미사용 |
| Day\_cnt | 사망여부 판단에는 사용하지 않음 |
| Intubated | 분석 방향과 다른 변수 |
| Pregnancy | 데이터 전처리 시, 1개 level만 남음 |
| Covid\_res | 모델링에서 미사용 |
| icu | 분석 방향과 다른 변수 |

|  |  |
| --- | --- |
| 제거변수명 | 제거 사유 |
| Day\_cnt | 사망여부 예측에는 사용하지 않음 |
| Is\_dead | 생존일수 예측에는 사용하지 않음 |

## 7. 데이터 전처리 – Train / Test 파일로 분리

Covid\_train 레코드 수 : 20000개 / Covid\_test 레코드 수 : 3158개

covid\_train = covid\_original\_without\_missing[1:20000,];

covid\_test = covid\_original\_without\_missing[-(1:20000),];

Covid\_dead\_train 레코드 수 : 3600 개 / Covid\_dead\_test 레코드 수 : 420 개

covid\_dead\_train = covid\_original\_dead\_without\_missing[1:3600,];

covid\_dead\_test = covid\_original\_dead\_without\_missing[-(1:3600),];

## 8. 데이터 전처리 – 클래스 불균형 해소를 위한 Oversampling

Covid\_train : is\_dead == 1 : 3822 개 / is\_dead == 2 : 16178 개

Covid\_test : is\_dead == 1 : 3822 개 / is\_dead == 2 : 16178 개

ROSE 패키지 사용. SMOTE (Synthetic Minority Over-sampling Technique) 알고리즘 통해 최근접 데이터 생성 및 기존 데이터에 취합 (Train 12356개, Test 2762개 레코드 생성)

library(ROSE);

train\_oversampling\_cnt = nrow(covid\_train) - 2\*nrow(covid\_train[covid\_train$is\_dead == 1, ]);

train\_oversampling\_data = ROSE(is\_dead ~ ., data = covid\_train, N=50000, seed=10 )$data;

train\_oversampling\_data\_dead = train\_oversampling\_data[train\_oversampling\_data$is\_dead==1,];

covid\_train\_oversampled =  
 rbind(covid\_train, train\_oversampling\_data\_dead[1:train\_oversampling\_cnt,]);

test\_oversampling\_cnt = nrow(covid\_test) - 2\*nrow(covid\_test[covid\_test$is\_dead == 1, ]);

test\_oversampling\_data = ROSE(is\_dead ~ ., data = covid\_test, N=50000, seed=20 )$data;

test\_oversampling\_data\_dead = test\_oversampling\_data[test\_oversampling\_data$is\_dead==1,];

covid\_test\_oversampled = rbind(covid\_test, test\_oversampling\_data\_dead[1:test\_oversampling\_cnt,]);

## 9. 최종 데이터 정리

|  |  |  |
| --- | --- | --- |
| 파일명 | 설명 | 레코드 수 |
| Covid\_train.csv | 사망여부 예측 train 데이터 | 32356 개 |
| Covid\_test.csv | 사망여부 예측 test 데이터 | 5920 개 |
| Covid\_dead\_train.csv | 생존일수 예측 train 데이터 | 3600 개 |
| Covid\_dead\_test.csv | 생존일수 예측 test 데이터 | 420 개 |

# **분석 방식**

## 1. 데이터 읽어오기 + 형 변환

covid\_train = read.csv("covid\_train.csv", header=T);

covid\_train$sex = as.factor(covid\_train$sex);

covid\_train$patient\_type = as.factor(covid\_train$patient\_type);

covid\_train$pneumonia = as.factor(covid\_train$pneumonia);

covid\_train$age = as.numeric(covid\_train$age);

……

## 2. 사망여부 예측 모델 학습

tree1 = tree (is\_dead ~ ., data=covid\_train, split = c("deviance"), na.action = na.pass, control = tree.control(nobs = nrow(covid\_train), minsize = 100, mindev = 0.005))

## 3. 생성한 모델을 이용하여 Test 데이터 사망여부 예측 수행

prediction = predict(tree1, covid\_test[], type="class")

summary(prediction)

## 4. 예측한 test 데이터의 정확도 측정

comparison=cbind(covid\_test,prediction)

comparison=as.data.frame(comparison)

print(paste("test 건수 : ",nrow(covid\_test)))

predictCorrect = comparison[comparison$is\_dead == comparison$prediction,];

print(paste("사망여부 예측성공 건수 : ", nrow(predictCorrect)))

print(paste("사망여부 예측 정확도 : " ,nrow(predictCorrect)/nrow(covid\_test)))

## 5. 정확도 상세 분석 – True-positive, True-negative, False-positive, False-negative rate 측정

tp = round(nrow(comparison[comparison$is\_dead == 1 & comparison$prediction == 1, ])/nrow(covid\_test),2);

tn = round(nrow(comparison[comparison$is\_dead == 2 & comparison$prediction == 2, ])/nrow(covid\_test),2);

fp = round(nrow(comparison[comparison$is\_dead == 2 & comparison$prediction == 1, ])/nrow(covid\_test),2);

fn = round(nrow(comparison[comparison$is\_dead == 1 & comparison$prediction == 2, ])/nrow(covid\_test),2);

confusion\_matrix = matrix(c(tp, fn, fp, tn), nrow = 2, byrow = TRUE, dimnames = list(c("Actual Positive", "Actual Negative"), c("Predicted Positive", "Predicted Negative")))

confusion\_matrix;

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | True-positive | False-negative |
| Actual Negative | False-positive | True-negative |

## 6. 생존일수 예측 모델 학습

tree11 = tree (day\_cnt ~ ., data = covid\_dead\_train,split = c("deviance"), na.action = na.pass, control = tree.control(nobs = nrow(covid\_dead\_train), minsize = 10, mindev = 0.001))

## 7. 생성한 모델을 이용하여 Test 데이터 생존일수 예측 수행

prediction\_dead = predict(tree11, covid\_dead\_test[], type="vector")

8. 예측한 test 데이터의 정확도 측정

comparison\_dead=cbind(covid\_dead\_test,prediction\_dead)

comparison\_dead=as.data.frame(comparison\_dead)

comparison\_dead$prediction\_dead = round(comparison\_dead$prediction\_dead)

print(paste("test 건수 : ", nrow(covid\_dead\_test)));

# 투병일수 예측성공 기준 설정

deadPredictCorrectCreteria = 10;

deadPredictCorrect = comparison\_dead[abs(comparison\_dead$day\_cnt-  
comparison\_dead$prediction\_dead)<=deadPredictCorrectCreteria, 0]

print(paste("투병일수 예측성공 건수(",deadPredictCorrectCreteria,"일) : " ,  
 nrow(deadPredictCorrect)));

print(paste("투병일수 예측 정확도(",deadPredictCorrectCreteria,"일) : ", nrow(deadPredictCorrect) / nrow(covid\_dead\_test)))

# **모델 별 분석**

## 1. Tree

tree1 = tree (is\_dead ~ ., data=covid\_train, split = c("deviance"), na.action = na.pass, control = tree.control(nobs = nrow(covid\_train), minsize = 100, mindev = 0.005))

텍스트, 도표, 직사각형, 라인이(가) 표시된 사진

자동 생성된 설명

1) root 32356 44850 1 ( 0.50000 0.50000 )

2) pneumonia: 1 24382 32880 1 ( 0.59671 0.40329 )

4) age < 53.0052 9144 12580 2 ( 0.44838 0.55162 )

8) age < 36.0006 3019 3836 2 ( 0.33157 0.66843 ) \*

9) age > 36.0006 6125 8490 1 ( 0.50596 0.49404 ) \*

5) age > 53.0052 15238 18970 1 ( 0.68572 0.31428 ) \*

3) pneumonia: 2 7974 8074 2 ( 0.20429 0.79571 )

6) age < 46.0175 3898 2135 2 ( 0.07799 0.92201 ) \*

7) age > 46.0175 4076 5141 2 ( 0.32507 0.67493 ) \*

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

중요 변수 : pneumonia, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3657"

[1] "사망여부 예측 정확도 : 0.617736486486486"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.34 | 0.16 |
| Actual Negative | 0.22 | 0.28 |

tree11 = tree (day\_cnt ~ ., data = covid\_dead\_train,split = c("deviance"), na.action = na.pass, control = tree.control(nobs = nrow(covid\_dead\_train), minsize = 10, mindev = 0.001))

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1) root 3600 199700.0 10.940

2) age < 79.5 3182 182500.0 11.160

4) renal\_chronic: 1 226 11150.0 9.549

8) age < 24.5 5 14.0 2.000 \*

9) age > 24.5 221 10850.0 9.719

18) age < 31.5 5 930.8 22.800 \*

19) age > 31.5 216 9043.0 9.417

38) age < 61.5 107 2763.0 8.196 \*

39) age > 61.5 109 5964.0 10.610

78) cardiovascular: 1 21 547.8 7.762 \*

79) cardiovascular: 2 88 5204.0 11.300

158) hypertension: 1 71 3138.0 10.460 \*

159) hypertension: 2 17 1813.0 14.760

318) age < 69.5 5 508.8 24.800 \*

319) age > 69.5 12 590.9 10.580 \*

5) renal\_chronic: 2 2956 170700.0 11.290

10) age < 47.5 594 32900.0 10.430

20) other\_disease: 1 29 3975.0 15.240

40) age < 33.5 17 2952.0 18.530 \*

41) age > 33.5 12 578.9 10.580 \*

21) other\_disease: 2 565 28220.0 10.180

42) age < 31.5 129 6266.0 8.682

84) tobacco: 1 6 261.3 14.330 \*

85) tobacco: 2 123 5804.0 8.407 \*

43) age > 31.5 436 21580.0 10.630

86) pneumonia: 1 394 20380.0 10.940 \*

87) pneumonia: 2 42 793.3 7.667 \*

11) age > 47.5 2362 137200.0 11.500

22) age < 69.5 1726 105600.0 11.800

44) diabetes: 1 757 44980.0 11.370 \*

45) diabetes: 2 969 60320.0 12.130 \*

23) age > 69.5 636 31110.0 10.700 \*

3) age > 79.5 418 15850.0 9.220

6) age < 94.5 403 15470.0 9.367

12) renal\_chronic: 1 24 437.3 6.167 \*

13) renal\_chronic: 2 379 14770.0 9.570 \*

7) age > 94.5 15 134.9 5.267 \*

텍스트, 폰트, 문서, 스크린샷이(가) 표시된 사진

자동 생성된 설명

중요 변수 : age, renal\_chronic

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 380"

[1] "투병일수 예측 정확도( 10 일) : 0.904761904761905"

## 2. rpart

covidrpart = rpart(is\_dead ~ ., data=covid\_train, method = "class", control = rpart.control(minsplit = 10, minbucket = 10, maxdepth = 10, cp = 0.005));

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1) root 32356 16178 1 (0.5000000 0.5000000)

2) pneumonia=1 24382 9833 1 (0.5967107 0.4032893)

4) age>=53.00521 15238 4789 1 (0.6857199 0.3142801) \*

5) age< 53.00521 9144 4100 2 (0.4483815 0.5516185)

10) age>=36.00065 6125 3026 1 (0.5059592 0.4940408)

20) age< 52.99964 5828 2793 1 (0.5207618 0.4792382)

40) age>=52.0017 214 0 1 (1.0000000 0.0000000) \*

41) age< 52.0017 5614 2793 1 (0.5024938 0.4975062)

82) diabetes=1 1872 752 1 (0.5982906 0.4017094) \*

83) diabetes=2 3742 1701 2 (0.4545697 0.5454303) \*

21) age>=52.99964 297 64 2 (0.2154882 0.7845118) \*

11) age< 36.00065 3019 1001 2 (0.3315667 0.6684333) \*

3) pneumonia=2 7974 1629 2 (0.2042889 0.7957111) \*

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 pneumonia, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3677"

[1] "사망여부 예측 정확도 : 0.621114864864865"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.29 | 0.21 |
| Actual Negative | 0.17 | 0.33 |

deadrpart = rpart(day\_cnt ~ ., data=covid\_dead\_train, method = "class", control = rpart.control(minsplit = 10, minbucket = 10, maxdepth = 10, cp = 0.001));

도표, 텍스트, 기술 도면, 스케치이(가) 표시된 사진

자동 생성된 설명

1) root 3600 3301 7 (0.0056 0.015 0.031 0.046 0.062 0.073 0.07 0.083 0.071 0.068 0.062 0.048 0.051 0.041 0.033 0.035 0.028 0.024 0.02 0.019 0.014 0.014 0.011 0.0072 0.0081 0.0081 0.0042 0.0069 0.005 0.0056 0.0044 0.0053 0.0011 0.0033 0.0019 0.0017 0.0033 0.0014 0.0019 0.00083 0.0011 0.0011 0.00083 0.00056 0.00028 0.0011 0.00028 0.00028 0.00028)

2) age< 27.5 105 91 3 (0.019 0.029 0.076 0.13 0.12 0.11 0.048 0.038 0.048 0.029 0.048 0 0.038 0.0095 0.029 0.0095 0.0095 0.038 0.019 0 0.019 0.0095 0.0095 0 0.0095 0 0.0095 0 0 0 0.019 0.019 0 0 0.0095 0 0.0095 0.0095 0 0 0.0095 0.0095 0 0 0 0 0 0 0)

4) inmsupr=1 15 10 2 (0 0.067 0.33 0 0.067 0 0 0.067 0.067 0.067 0 0 0 0.067 0.067 0 0 0 0.067 0 0 0 0.067 0 0 0 0 0 0 0 0 0.067 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0) \*

5) inmsupr=2 90 76 3 (0.022 0.022 0.033 0.16 0.13 0.13 0.056 0.033 0.044 0.022 0.056 0 0.044 0 0.022 0.011 0.011 0.044 0.011 0 0.022 0.011 0 0 0.011 0 0.011 0 0 0 0.022 0.011 0 0 0.011 0 0.011 0.011 0 0 0.011 0.011 0 0 0 0 0 0 0) \*

(생략)…..

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

중요 변수 : age, inmsupr, cardiovascular

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 373"

[1] "투병일수 예측 정확도( 10 일) : 0.888095238095238"

## 3. C4.5

J48Fit = J48(is\_dead ~ ., data = covid\_train, control = Weka\_control(C = 0.1));

스케치, 그림, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

age <= 53

| age <= 36

| | age <= -0.060638: 1 (52.0)

| | age > -0.060638

| | | age <= 35.993483

| | | | age <= 35.025615

| | | | | age <= 34.992045

| | | | | | age <= 34: 2 (5022.0/964.0)

| | | | | | age > 34: 1 (64.0)

| | | | | age > 34.992045: 2 (258.0/17.0)

| | | | age > 35.025615: 1 (51.0)

| | | age > 35.993483: 2 (258.0/18.0)

| age > 36

| | age <= 36.999387: 1 (88.0)

| | age > 36.999387

| | | age <= 37: 2 (266.0/28.0)

(생략….)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 5413"

[1] "사망여부 예측 정확도 : 0.914358108108108"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.43 | 0.07 |
| Actual Negative | 0.01 | 0.49 |

#C4.5 알고리즘은 Factor 형 변수를 다루기 위해 만들어진 알고리즘이라, numeric class 를 지원하지 않음

## 4. C5.0

C50Fit = C5.0(is\_dead~., data = covid\_train, trials=100);

스케치, 그림, 라인이(가) 표시된 사진

자동 생성된 설명

Attribute usage:

100.00% pneumonia

100.00% age

99.87% contact\_other\_covid

97.07% renal\_chronic

94.92% tobacco

94.70% inmsupr

93.12% obesity

90.71% copd

85.58% other\_disease

83.34% cardiovascular

75.97% asthma

75.02% diabetes

66.30% hypertension

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 4165"

[1] "사망여부 예측 정확도 : 0.703547297297297"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.36 | 0.14 |
| Actual Negative | 0.16 | 0.34 |

#C5.0 알고리즘은 Factor 형 변수를 다루기 위해 만들어진 알고리즘이라, numeric class 를 지원하지 않음

## 5. chaid

chaidFit = chaid(is\_dead~., data = covid\_train, control = chaid\_control(minsplit = 5000, minprob = 0.4));

그림, 스케치, 텍스트, 도표이(가) 표시된 사진

자동 생성된 설명

Model formula:

is\_dead ~ sex + patient\_type + pneumonia + diabetes + copd +

asthma + inmsupr + hypertension + other\_disease + cardiovascular +

obesity + renal\_chronic + tobacco + contact\_other\_covid

Fitted party:

[1] root

| [2] pneumonia in 1

| | [3] hypertension in 1

| | | [4] diabetes in 1

| | | | [5] renal\_chronic in 1: 1 (n = 849, err = 26.1%)

| | | | [6] renal\_chronic in 2: 1 (n = 4971, err = 31.7%)

| | | [7] diabetes in 2: 1 (n = 4507, err = 35.3%)

| | [8] hypertension in 2

| | | [9] diabetes in 1: 1 (n = 3544, err = 35.8%)

| | | [10] diabetes in 2

| | | | [11] contact\_other\_covid in 1: 2 (n = 2779, err = 42.0%)

| | | | [12] contact\_other\_covid in 2

| | | | | [13] cardiovascular in 1: 1 (n = 240, err = 29.6%)

| | | | | [14] cardiovascular in 2

| | | | | | [15] copd in 1: 1 (n = 259, err = 33.2%)

| | | | | | [16] copd in 2

| | | | | | | [17] obesity in 1: 1 (n = 1489, err = 43.6%)

| | | | | | | [18] obesity in 2

| | | | | | | | [19] renal\_chronic in 1: 1 (n = 141, err = 36.2%)

| | | | | | | | [20] renal\_chronic in 2: 1 (n = 5603, err = 48.3%)

| [21] pneumonia in 2

| | [22] hypertension in 1: 2 (n = 2123, err = 34.9%)

| | [23] hypertension in 2

| | | [24] contact\_other\_covid in 1: 2 (n = 2182, err = 7.6%)

| | | [25] contact\_other\_covid in 2: 2 (n = 3669, err = 19.7%)

Number of inner nodes: 12

Number of terminal nodes: 13

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, hypertension

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3579"

[1] "사망여부 예측 정확도 : 0.604560810810811"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.33 | 0.17 |
| Actual Negative | 0.23 | 0.27 |

#chaid 함수는 범주형 변수들로만 이루어진 데이터를 처리하기 위해 설계되어서, numeric class 를 지원하지 않음

## 6. DT-ensemble

baggFit = bagging(is\_dead~., data=covid\_train, nbag = 100, control=rpart.control(minsplit = 10),coob=T);

1) root 32356 16167 2 (0.4996600 0.5003400)

2) pneumonia=1 24319 9851 1 (0.5949258 0.4050742)

4) age>=53.00573 15119 4771 1 (0.6844368 0.3155632) \*

5) age< 53.00573 9200 4120 2 (0.4478261 0.5521739)

10) diabetes=1 2395 971 1 (0.5945720 0.4054280) \*

11) diabetes=2 6805 2696 2 (0.3961793 0.6038207) \*

3) pneumonia=2 8037 1699 2 (0.2113973 0.7886027) \*

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, age, diabetes

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3647"

[1] "사망여부 예측 정확도 : 0.616047297297297"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.29 | 0.21 |
| Actual Negative | 0.17 | 0.33 |

baggFit\_dead = bagging(day\_cnt~., data=covid\_dead\_train, nbag = 100, control=rpart.control(minsplit = 10),coob=T);

1) root 3600 196074.80 10.935000

2) age>=84.5 178 4082.14 7.544944 \*

3) age< 84.5 3422 189840.60 11.111340 \*

텍스트, 폰트, 화이트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

중요변수 : age

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 387"

[1] "투병일수 예측 정확도( 10 일) : 0.921428571428571"

## 7. Random Forest

rfFit = randomForest(is\_dead~., data=covid\_train, importance=TRUE, ntree=1000, mtry=2);

텍스트, 번호, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

중요변수 : pnuemonua, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3653"

[1] "사망여부 예측 정확도 : 0.617060810810811"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.31 | 0.19 |
| Actual Negative | 0.19 | 0.31 |

rfFit\_dead = randomForest(day\_cnt~., data=covid\_dead\_train, importance=TRUE, ntree=1000, mtry=2);

텍스트, 번호, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

중요변수 : age, renal\_chronic

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 388"

[1] "투병일수 예측 정확도( 10 일) : 0.923809523809524"

## 8. Boosting

gbmFit = gbm(is\_dead~., data=covid\_train,distribution = "multinomial", n.trees = 1000, shrinkage = 0.01, interaction.depth = 4);

var rel.inf

age age 68.67667382

pneumonia pneumonia 26.03684160

diabetes diabetes 1.78418201

contact\_other\_covid contact\_other\_covid 1.25199329

hypertension hypertension 0.72083882

obesity obesity 0.51382311

renal\_chronic renal\_chronic 0.45129265

tobacco tobacco 0.13463609

other\_disease other\_disease 0.11725880

cardiovascular cardiovascular 0.09241807

copd copd 0.09155611

inmsupr inmsupr 0.06817245

asthma asthma 0.06031319

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : age, pneumonia

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 4231"

[1] "사망여부 예측 정확도 : 0.714695945945946"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.35 | 0.15 |
| Actual Negative | 0.14 | 0.36 |

gbmFit\_dead = gbm(day\_cnt~., data=covid\_dead\_train,distribution = "gaussian", n.trees = 1000, shrinkage = 0.01, interaction.depth = 4);

var rel.inf

age age 60.0642032

other\_disease other\_disease 5.4262469

diabetes diabetes 4.6390326

renal\_chronic renal\_chronic 4.2719086

obesity obesity 4.2463639

pneumonia pneumonia 4.1910174

hypertension hypertension 4.0452540

contact\_other\_covid contact\_other\_covid 3.6572884

cardiovascular cardiovascular 2.7982949

inmsupr inmsupr 2.7207409

copd copd 2.1587412

tobacco tobacco 0.9795818

asthma asthma 0.8013262

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : age, other\_disease

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 387"

[1] "투병일수 예측 정확도( 10 일) : 0.921428571428571"

## 9. MBR

prediction = knn(covid\_train[,1:15], covid\_test[,1:15], cl=train\_target, k=3, prob=F);

knn 알고리즘은 내부적으로 decision tree를 만들지 않음

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 4403"

[1] "사망여부 예측 정확도 : 0.74375"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.30 | 0.20 |
| Actual Negative | 0.05 | 0.45 |

prediction\_dead = knn(covid\_dead\_train[,1:15], covid\_dead\_test[,1:15], cl=dead\_train\_target, k=3, prob=F);

knn 알고리즘은 내부적으로 decision tree를 만들지 않음

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 349"

[1] "투병일수 예측 정확도( 10 일) : 0.830952380952381"

## 10. Neural network

nn1 = neuralnet(is\_dead~., data=covid\_train, algorithm = "rprop+", act.fct = 'logistic',linear.output = TRUE, hidden =3);

텍스트, 도표, 스케치, 라인이(가) 표시된 사진

자동 생성된 설명

중요변수 : neuralnet 패키지에서는 기본적으로 변수 중요도를 직접 제공하지 않음

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3636"

[1] "사망여부 예측 정확도 : 0.614189189189189"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.32 | 0.18 |
| Actual Negative | 0.20 | 0.30 |

nn1\_dead = neuralnet(day\_cnt~., data=covid\_dead\_train, algorithm = "rprop+", act.fct = 'logistic',linear.output = TRUE, hidden =3, stepmax = 100000);

- 생존일수의 경우, 학습수렴 안함

## 11. Regression

regressionFit = lm(is\_dead ~ ., data=covid\_train);

Call:

lm(formula = is\_dead ~ ., data = covid\_train)

Residuals:

Min 1Q Median 3Q Max

-1.1267 -0.3810 -0.1290 0.4417 0.9735

Coefficients: (2 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.238965 0.057632 21.498 < 2e-16 \*\*\*

sex NA NA NA NA

patient\_type NA NA NA NA

pneumonia 0.308114 0.006036 51.049 < 2e-16 \*\*\*

age -0.005592 0.000143 -39.095 < 2e-16 \*\*\*

diabetes 0.059541 0.005754 10.348 < 2e-16 \*\*\*

copd -0.016383 0.011281 -1.452 0.14644

asthma -0.014946 0.014270 -1.047 0.29493

inmsupr 0.017241 0.013079 1.318 0.18743

hypertension 0.030859 0.005948 5.189 2.13e-07 \*\*\*

other\_disease 0.032470 0.012166 2.669 0.00761 \*\*

cardiovascular 0.001250 0.010668 0.117 0.90669

obesity 0.033247 0.005850 5.683 1.33e-08 \*\*\*

renal\_chronic 0.062355 0.010855 5.744 9.32e-09 \*\*\*

tobacco -0.035682 0.012384 -2.881 0.00396 \*\*

contact\_other\_covid -0.066049 0.005663 -11.662 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.451 on 32342 degrees of freedom

Multiple R-squared: 0.1867, Adjusted R-squared: 0.1864

F-statistic: 571.1 on 13 and 32342 DF, p-value: < 2.2e-16

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3617"

[1] "사망여부 예측 정확도 : 0.61097972972973"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.31 | 0.19 |
| Actual Negative | 0.20 | 0.30 |

regressionFit\_dead = lm(day\_cnt ~ ., data=covid\_dead\_train);

Call:

lm(formula = day\_cnt ~ ., data = covid\_dead\_train)

Residuals:

Min 1Q Median 3Q Max

-11.840 -5.118 -1.803 3.269 43.002

Coefficients: (2 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.789655 2.815601 2.411 0.01594 \*

sex NA NA NA NA

patient\_type NA NA NA NA

pneumonia -1.146756 0.420694 -2.726 0.00644 \*\*

age -0.005432 0.008215 -0.661 0.50855

diabetes 0.371892 0.264079 1.408 0.15914

copd 0.748439 0.510778 1.465 0.14293

asthma 0.089652 0.765361 0.117 0.90676

inmsupr 1.076194 0.672527 1.600 0.10964

hypertension 0.197530 0.272466 0.725 0.46852

other\_disease -1.026140 0.606404 -1.692 0.09070 .

cardiovascular 0.734718 0.490061 1.499 0.13390

obesity -0.024487 0.273988 -0.089 0.92879

renal\_chronic 1.587909 0.500762 3.171 0.00153 \*\*

tobacco -0.244416 0.633829 -0.386 0.69980

contact\_other\_covid -0.477998 0.292047 -1.637 0.10178

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.422 on 3586 degrees of freedom

Multiple R-squared: 0.01077, Adjusted R-squared: 0.007186

F-statistic: 3.004 on 13 and 3586 DF, p-value: 0.0002075

텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, renal\_chronic

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 387"

[1] "투병일수 예측 정확도( 10 일) : 0.921428571428571"

## 12. Logistic Regression

logisticFit = glm(is\_dead~., data =covid\_train, family = binomial);

Call:

glm(formula = is\_dead ~ ., family = binomial, data = covid\_train)

Coefficients: (2 not defined because of singularities)

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.1843568 0.2839009 4.172 3.02e-05 \*\*\*

sex NA NA NA NA

patient\_type NA NA NA NA

pneumonia -1.5240030 0.0321392 -47.419 < 2e-16 \*\*\*

age 0.0283538 0.0007523 37.692 < 2e-16 \*\*\*

diabetes -0.2805133 0.0275182 -10.194 < 2e-16 \*\*\*

copd 0.0964173 0.0547015 1.763 0.07797 .

asthma 0.0621049 0.0706011 0.880 0.37904

inmsupr -0.0877809 0.0653804 -1.343 0.17940

hypertension -0.1291804 0.0284484 -4.541 5.60e-06 \*\*\*

other\_disease -0.1554985 0.0618984 -2.512 0.01200 \*

cardiovascular -0.0007630 0.0528927 -0.014 0.98849

obesity -0.1824579 0.0283143 -6.444 1.16e-10 \*\*\*

renal\_chronic -0.3180567 0.0539932 -5.891 3.85e-09 \*\*\*

tobacco 0.1576789 0.0605036 2.606 0.00916 \*\*

contact\_other\_covid 0.3335346 0.0280642 11.885 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 44855 on 32355 degrees of freedom

Residual deviance: 38157 on 32342 degrees of freedom

AIC: 38185

Number of Fisher Scoring iterations: 4

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, age

[1] "test 건수 : 5920"

[1] "사망여부 예측성공 건수 : 3606"

[1] "사망여부 예측 정확도 : 0.609121621621622"

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 0.30 | 0.20 |
| Actual Negative | 0.19 | 0.31 |

logisticFit\_dead = glm(day\_cnt~., data =covid\_dead\_train, family = gaussian );

Call:

glm(formula = day\_cnt ~ ., family = gaussian, data = covid\_dead\_train)

Coefficients: (2 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.789655 2.815601 2.411 0.01594 \*

sex NA NA NA NA

patient\_type NA NA NA NA

pneumonia -1.146756 0.420694 -2.726 0.00644 \*\*

age -0.005432 0.008215 -0.661 0.50855

diabetes 0.371892 0.264079 1.408 0.15914

copd 0.748439 0.510778 1.465 0.14293

asthma 0.089652 0.765361 0.117 0.90676

inmsupr 1.076194 0.672527 1.600 0.10964

hypertension 0.197530 0.272466 0.725 0.46852

other\_disease -1.026140 0.606404 -1.692 0.09070 .

cardiovascular 0.734718 0.490061 1.499 0.13390

obesity -0.024487 0.273988 -0.089 0.92879

renal\_chronic 1.587909 0.500762 3.171 0.00153 \*\*

tobacco -0.244416 0.633829 -0.386 0.69980

contact\_other\_covid -0.477998 0.292047 -1.637 0.10178

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 55.09196)

Null deviance: 199711 on 3599 degrees of freedom

Residual deviance: 197560 on 3586 degrees of freedom

AIC: 24665

Number of Fisher Scoring iterations: 2

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

중요변수 : pneumonia, renal\_chronic

[1] "test 건수 : 420"

[1] "투병일수 예측성공 건수( 10 일) : 387"

[1] "투병일수 예측 정확도( 10 일) : 0.921428571428571"

# **분석결과**

## 1. 사망여부 예측

|  |  |  |
| --- | --- | --- |
| 모델 | 정확도 | 중요변수 |
| Tree | 0.617 | Pneumonia, age |
| rpart | 0.621 | Pneumonia, age |
| C4.5 | 0.914 | Age |
| C5.0 | 0.703 | Pneumonia, age |
| Chaid | 0.604 | Pneumonia, hypertension |
| DT\_ensemble | 0.616 | Pneumonia, age, diabetes |
| Random Forest | 0.617 | Pneumonia, age |
| Boosting | 0.714 | Pneumonia, age |
| MBR | 0.743 | - |
| Neural network | 0.614 | - |
| Regression | 0.610 | Pneumonia, age |
| Logistic regression | 0.609 | Pneumonia, age |

C4.5 모델이 예측 정확도가 가장 높으며, pneumonia, age 변수가 사망여부에 가장 큰 영향을 미침

## 2. 생존일수 예측

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모델 | 정확도(5일) | 정확도(7일) | 정확도(10일) | 중요변수 |
| Tree | 0.521 | 0.726 | 0.900 | Age, renal\_chronic |
| rpart | 0.666 | 0.792 | 0.888 | Age, inmsupr, cardiovascular |
| C4.5 |  |  |  |  |
| C5.0 |  |  |  |  |
| Chaid |  |  |  |  |
| DT\_ensemble | 0.492 | 0.697 | 0.921 | Age |
| Random Forest | 0.504 | 0.723 | 0.923 | Age, renal\_chronic |
| Boosting | 0.540 | 0.742 | 0.914 | Age, other\_disease |
| MBR | 0.545 | 0.650 | 0.800 | - |
| Neural network |  |  |  |  |
| Regression | 0.516 | 0.726 | 0.921 | Pneumonia, renal\_chronic |
| Logistic regression | 0.517 | 0.726 | 0.921 | Pneumonia, renal\_chronic |

생존일수의 경우, 임계값에 따라 예측정확도가 달라지며, age, renal\_chronic 변수가 생존일수에 가장 큰 영향을 미침

# **한계점 및 개선사항**

## 1. Oversampling 관련 데이터 생성 문제

SMOTE 알고리즘 통한 오버샘플링 진행 시, 생성된 데이터가 기존 데이터와 상이한 문제 발생

* Age 가 정수가 아닌 유리수로 생성됨
* Age 가 음수인 경우 발생

따라서 오버샘플링 과정에서 추가적인 작업을 통한 데이터 처리가 필요