**<비즈니스인텔리전스와 데이터마이닝>**

**빅데이터 분석학과 정찬미, 노윤지**

**자동차 부품의 생산데이터로 불량품과 양품 예측하기**

**[목차]**

1. 프로젝트 배경 및 목적
2. 선행연구
3. 분석대상 데이터와 적용할 분석기법
4. 분석결과에 대한 해석 및 요약
5. 분석결과의 활용방안 및 의의
6. 분석에 사용된 R 코드 전문
7. **프로젝트 배경 및 목적**
8. 프로젝트 배경

인공신경망과 의사결정나무 모델을 개발하여 기업의 부도 여부를 예측했던 경험을 토대로 자동차 부품 데이터를 사용하여 불량품과 양품을 예측하는 모형을 구축하고자 한다.

1. 프로젝트 목적
2. 품질 관리 분야에서의 데이터마이닝 적용 가능 여부 확인

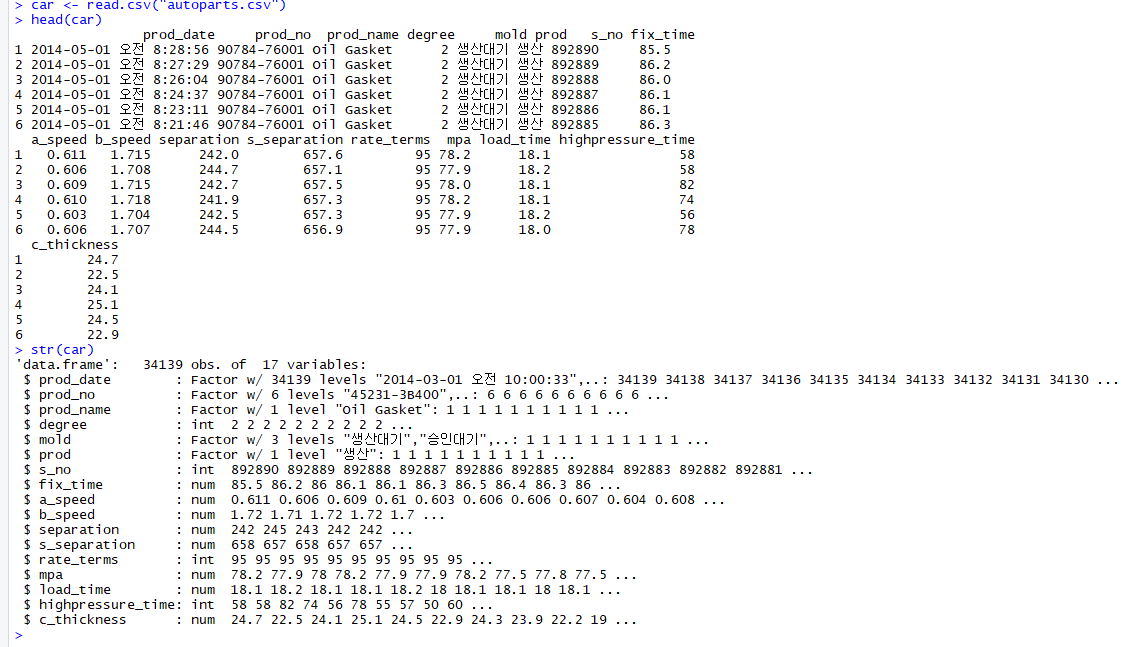
데이터마이닝 기법들을 활용하여 불량의 원인을 도출해낼 수 있다면 비용 절감 및 생산성 향상을 통한 가격 경쟁력을 높일 수 있다.

1. 세 가지 분석 기법들의 모형 성능 평가

로지스틱 회귀분석, 인공신경망, 의사결정나무 모형 분석을 통해 불량/양품에 영향을 미친 부품을 판별한다. 이를 통해 세 가지 모형의 성능을 비교, 평가할 수 있다.

1. **선행연구**
2. 로지스틱 회귀분석 연구 사례
3. 논문 제목 : 다양한 데이터마이닝 알고리즘 및 Missing Value 처리를 통항 자동차 부품 생산 라인에서 불량품 예측
4. 예측 목적 : 최종 부품 상태가 정상인지 비정상인지 예측
5. 데이터 : Bosch사가 제공한 데이터
6. 변수 : 제품 ID, 각 공정 라인 별 측정된 센서 값이 정상이면 0, 불량이면 1로 분류
7. 분석방법 : 로지스틱 회귀분석을 사용했을 때와 부스팅을 사용했을 때의 차이점을 제시
8. 인공신경망 연구 사례
9. 논문 제목 : 데이터마이닝 기법을 이용한 제조 공정내의 불량항목별 예측방법
10. 예측 목적 : 공정 이상을 해결하는 과정에 소요되는 시간을 단축
11. 데이터 : 자동차 내.외장품 생산업체인 국내 D기업의 범퍼공장 데이터
12. 변수 : 각 공정의 온도와 습도 및 교반시간
13. 분석방법 : 사전처리, 불량항목별 분석, 불량항목별 예측
14. 의사결정나무 연구 사례
15. 논문 제목 : 의사결정트리 기법을 이용한 스틸 파이프 생산 수율 및 불량률 측정에 관한 연구
16. 예측 목적 : 주요 특성들이 수율과 불량률에 미치는 영향 정도를 추정
17. 데이터 : 국내 스틸 파이프업계에서 실제 생산된 스틸 파이프에 대한 데이터
18. 변수 : 스틸 파이프의 주요 인자로 선정된 8가지 특성을 입력 변수로 활용
19. 분석 방법 : 주요 특성별 수율과 불량률의 분포를 분석
20. **분석 대상 데이터와 적용할 분석 기법**
21. 분석 대상 데이터 설명

R에서 head()와 str()를 통해 살펴본 데이터의 기본적인 특성은 다음의 Figure 1과 같다. 현재 car 데이터에는 17개의 변수와 34,139개의 관측치가 존재한다.



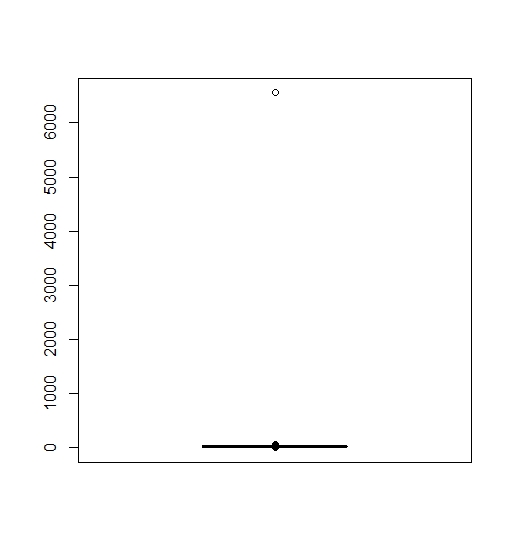
< Figure 1 > 데이터의 head와 구조

1. 입력변수/ 종속변수 선정

총 17개 변수들 중, 생산날짜와 같이 불필요한 변수들을 제외하고 탕구 두께에 영향을 줄 것으로 생각되는 공정변수 고정시간, A속도, B속도, 실압력, 하중시간, 고압시간을 입력변수로 선정하였다.

1. 입력변수: 고정시간, A속도, B속도, 실압력, 하중시간, 고압시간
2. 종속변수: 탕구 두께
3. 종속 변수 불량/ 양품 기준 선정

종속 변수로 사용될 탕구 두께 변수의 boxplot은 아래 Figure 2와 같다. 용이한 분석을 위해 전처리 단계에서 탕구 두께 자료 중 1000을 넘는 이상치들은 제외하였다. 그 후 생산관리의 공정불량률의 평균이 전체 생산량의 1% 미만이라는 연구결과를 토대로 탕구 두께 변수의 신뢰구간 99%를 구했다. t.test() 함수를 통해 구해진 99% 신뢰구간은 [24.00817, 24.10827] 이었다. 하지만 이를 불량품의 구간으로 사용하기에는 너무 적은 수의 데이터 수가 사용되어 유의미한 분석결과를 기대하기 힘드므로, quantile() 함수를 사용하여 도출된 구간으로 불량의 범위를 지정하였다. 즉, 종속 변수 데이터의 위 아래 0.05%씩을 제외한 구간에 해당하는 데이터를 양품으로, 그 외의 구간에 해당하는 물품들은 불량품으로 지정한 것이다. quantile()함수를 사용했을때 구해진 양품 탕구 두께 구간은 12.5 이상 38.6 미만이었다. 여기서 기준이 되는 각 수치를 1의 자리까지 반올림하여 얻어진 구간 [13,39]을 넘어서는 탕구 두께는 불량품으로 지정하였다. 이와 같은 기준으로 나눠어진 데이터들 중 양품의 개수는 33,792개, 불량품의 개수는 335개이다.



< Figure 2 > 탕구 두께 변수 boxplot

1. 분석 기법

탕구 두께가 [13,39] 구간에 속하는지 여부를 반응 변수(불량품 또는 양품)로 하여 공정변수(입력변수)들이 어떤 영향을 미치는지 분석하였다. 즉, 반응 변수를 연속형이 아닌 이산형으로 변환하여 분류 문제로 설정한 것이다. 일반화선형모형으로 로지스틱 회귀모형을 선정하였으며, 비선형 모형으로 의사결정나무 모형 및 신경망 모형을 활용하여 분석해 보았다. ROC curve로 도출된 AUC 통계량과 ecdf 그래프 등을 사용하여 모형의 성능을 비교하였다.

1. 모형 성능 평가 기준
2. ROC curve

ROC(Receiver Operating Charactersic)곡선은 cutoff 값의 변화에 따른 민감도와 특이도의 관계를 표현한 그래프이다. True Positive Rate(Sensitivity, 민감도)는 실제값이 1이라면 예측을 1로 분류한 비율이고, True Negative Rate(Specificity, 특이도, 1-민감도)는 실제 0인 값을 0으로 얼마나 잘 예측했는지 나타내는 비율이다. ROC curve 값은 x축은 FPR, Y축은 TPR 그려진다. ROC curve가 사각형을 가득 채울수록 분류율이 높다고 볼 수 있다.

(2) AUC(Area Under Curve)

ROC curve의 아래 면적으로 최대값은 1이다. AUC가 1에 가까울수록 Sensitivity와 Specificity가 모두 높아 우수한 분류 모형으로 볼 수 있다. AUC가 0.5 아래로 떨어지면 성능이 매우 안 좋은 모형이다.

(3) ECDF

통계 패키지 R에서는 ecdf()함수를 이용하여 각 그룹의 예측확률의 분포함수를 추정할 수 있다. 본 분석에서 양품과 불량품 각각에 대한 분류 예측률에 대하여 비교해볼 수 있는 그래프를 나타낸다.

1. **분석결과에 대한 해석 및 요약**
2. Train set / Test set으로 할당

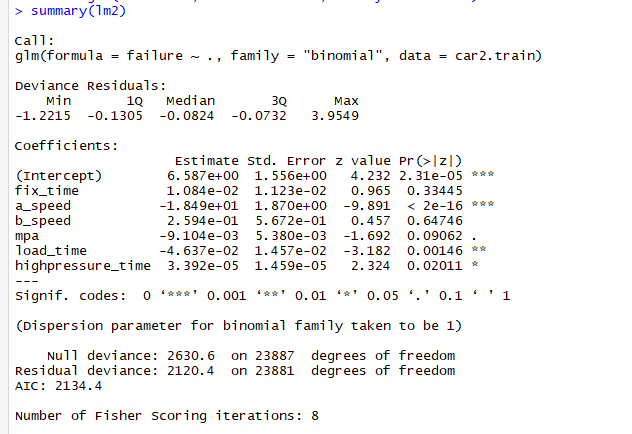
Train 데이터와 Test 데이터를 7:3의 비율로 할당하였다. 불량품과 양품 각각의 분류예측을 용이하게 하기 위하여, 둘 다 동등하게 7:3으로 Train 데이터와 Test 데이터로 들어갈 수 있도록 할당하였다. 할당된 결과는 아래 표1과 같다.

|  |  |  |
| --- | --- | --- |
|  | 양품 | 불량품 |
| Train data | 23654 | 234 |
| Test data | 10138 | 100 |

< 표 1 > train data와 test set에 할당된 결과

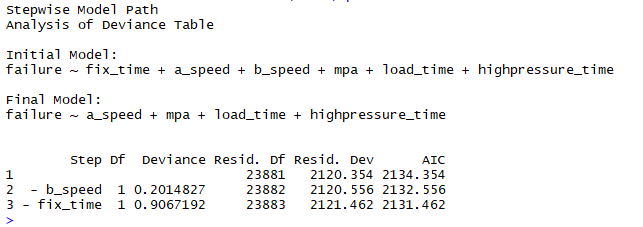
1. 로지스틱 회귀분석

로지스틱 회귀분석을 위해 glm()함수를 사용했다. glm()함수는 일반화선행모형 적합 함수로 binomial 설정으로 로지스틱 회귀모형에 적합하다. 모든 변수들을 입력 변수로 활용하였으며, 사용된 데이터는 Train 데이터이다. Summary()함수를 통해 살펴본 모형의 결과는 다음과 같다.



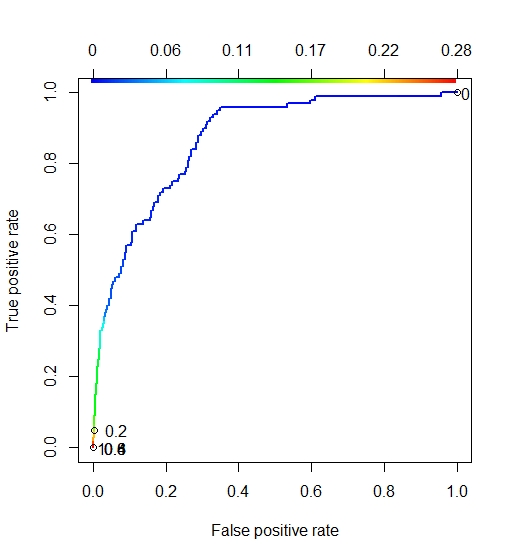
< Figure 3 > 로지스틱 회귀분석 실행 결과

모형의 간결함과 적중률을 높이기 위해 stepAIC() 함수를 사용하여 변수를 선택하였다. 로지스틱 회귀분석 모형에서 변수 선택법에는 3가지가 있는데, 그 중에서도 단계적 변수 선택 방법을 택하였다. StepAIC함수를 통해 나온 모형을 anova로 간략히 살펴보면, 유의수준 0.05에서 고정시간, b속도가 유의하지 않은 것으로 나타났다. 아래의 Figure 4를 보면 AIC에 의한 변수 선택 결과 위 2개의 변수를 제외한 모형이 최적화 된 것을 알 수 있다.



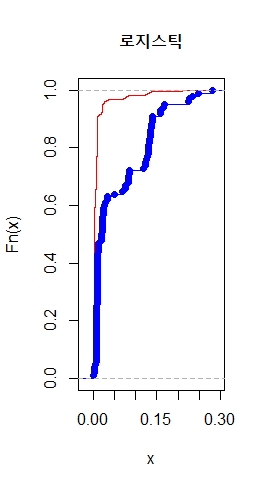
< Figure 4 > 로지스틱 회귀모형 AIC 변수선택 결과

로지스틱 회귀모형의 ROC 커브는 아래의 Figure 5와 같다. 1에 가까운 모양이고 AUC 값은 86.8 으로, 분류 예측을 비교적 잘 수행했다고 볼 수 있다.



< Figure 5 > 로지스틱 회귀모형의 ROC 커브

양품과 불량의심제품의 두 그룹을 나누어 각각 불량예측확률을 저장하였다. 아래의Figure 6은 Ecdf() 함수를 이용하여 각 그룹의 예측 확률의 분포 함수를 plot()함수로 그린 그래프이다. 빨간색인 양품은 0 근처에 예측 확률이 집중되어 있으나, 불량의심품은 0 근처와 1 근처에 고르게 예측 확률이 분포한다. 즉, 불량의심품 분류 예측의 정확도가 떨어진다는 것을 알 수 있다.



< Figure 6 > 로지스틱 회귀모형의 ecdf 그래프

(빨간색 : 양품, 파란색 : 불량품)

1. 인공신경망

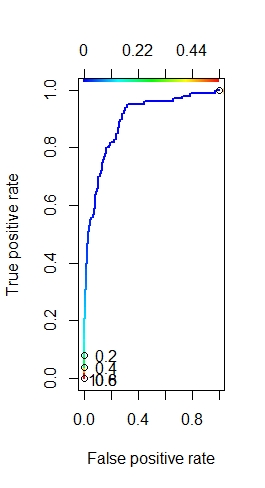
신경망모형은 생물학의 신경망을 토대로 만든 학습 알고리즘으로 비선형 모형이다. 신경망모형의 적합을 위해서 설명변수들을 표준화(normalization)하였다. R의 패키지 중 RSNNS 패키지를 사용하였고, normalizeDate()를 이용하여 변수들을 표준화하였다.

은닉층의 개수를 10개로 설정하고 학습을 위한 최대 반복수를 170으로 설정하였다. 그 이유는 은닉층의 개수와 학습 반복수에 따라 AUC 값을 측정한 결과, 은닉층 개수 10개, 최대 반복수 170개 일 때가 AUC 값이 가장 크게 나타났기 때문이다.

|  |  |  |
| --- | --- | --- |
| 은닉층 수 | 학습 반복 횟수 | AUC |
| 5 | 50 | 0.8944664 |
| 5 | 100 | 0.8987207 |
| 10 | 100 | 0.898697 |
| 10 | 200 | 0.8940896 |
| 10 | 150 | 0.9004281 |
| 20 | 150 | 0.8985323 |
| 10 | 170 | 0.9029375 |
| 10 | 175 | 0.8972835 |
| 20 | 340 | 0.8928221 |
| 100 | 1700 | 0.8894161 |

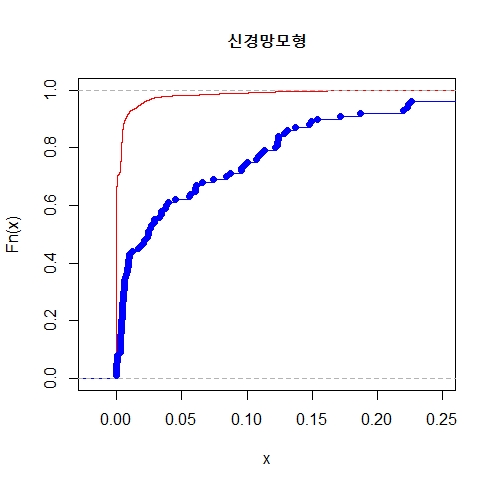
< 표 2 > 은닉층 수와 학습 반복 횟수에 따른 AUC 값

로지스틱 회귀분석과 마찬가지로 인공신경망 모형의 ROC curve 그래프를 그리고 AUC 통계량을 구하였다. AUC 통계량은 88.99으로, 로지스틱 회귀모형보다 조금 더 높은 수치를 보였다. Figure 7에 나타난 ROC curve 그래프를 보면 로지스틱 회귀모형보다 조금 더 1에 가까운 모양임을 알 수 있다.



< Figure 7 > 인공신경망 모델의 ROC curve

다음의 Figure 8 에서 양품과 불량품 분류 ecdf 그래프를 신경망모형 분류 예측결과를 알 수 있다. 로지스틱 회귀분석 모형과 마찬가지로 양품과 불량의심제품의 두 그룹을 나누어 각각 불량예측확률을 저장하였다. 로지스틱 회귀분석과 마찬가지로 양품을 양품으로 잘 분류하였지만, 불량의심제품 분류 예측 정확도는 그에 못미치고 있었다.



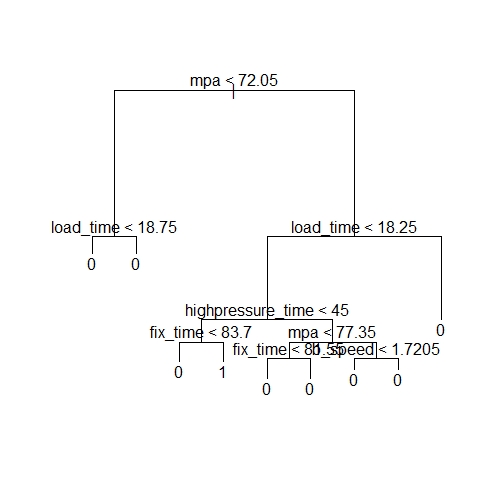
< Figure 8 > 인공신경망 ecdf 그래프

(빨간색 : 양품, 파란색 : 불량품)

1. 의사결정나무

의사결정나무는 분류 규칙에 의해 자료를 이원화하는 과정을 반복하는 것으로 하위 노드의 불순도가 가능하면 커지는 방향으로 분류 규칙이 결정된다. 로지스틱 모형과 달리 비선형모형이다. 현 자료의 경우 공정변수들의 값이 이원화되어 있는 경우가 많았으므로 로지스틱 모형보다는 의사결정나무 모형에 의한 분류 성능이 더 좋을 것으로 기대해 볼 수 있다. 직관적으로 모형의 이해가 가능하고 해석하기 좋으며 시각화가 잘 된다는 장점이 있다.

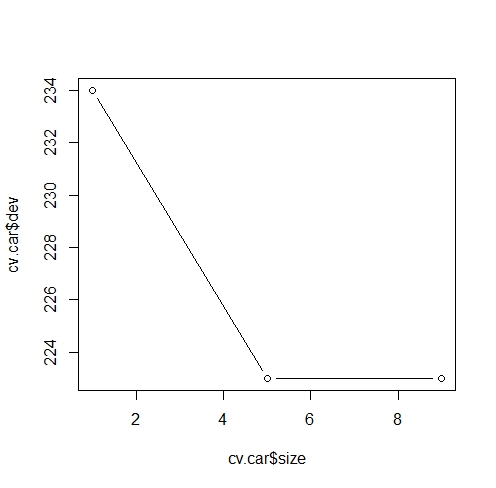
아래의 Figure 9에 Tree()함수를 사용하여 의사결정나무를 적합한 결과가 나타나 있다. 의사결정나무 적합 결과 실제 나무 적합을 위해 쓰인 변수는 총 6가지 중에서 a속도를 제외한 mpa(실압력), load\_time(하중시간), fix\_time(고정시간), highpressure\_time(고압시간), b\_speed(b 속도)이다.



< Figure 9 > 의사결정나무 변수 적합 결과

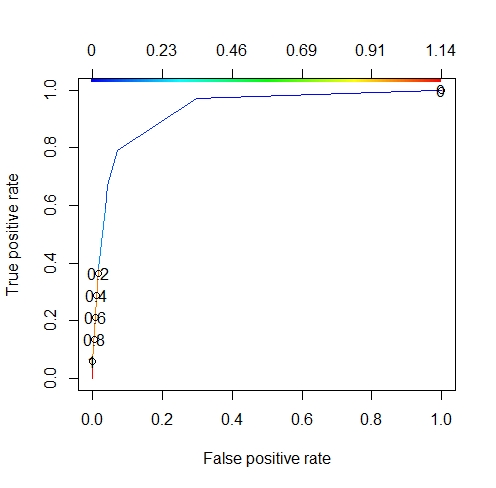
보통 큰 나무모형은 과적합(overfitting) 문제가 있는 것으로 알려져 있다. 이를 미연에방지하기 위해 가지치기(pruning)을 통해 나무의 크기를 줄여 예측 정확도를 향상시켰다.

가지치기를 위해 cross-validation에 의해 deviance를 최소로 하는 terminal node의 개수를 찾아보았다. 여기서 deviance는 평균제곱오차(MSE)의 확장된 개념으로 일종의 오차이다. 아래의 Figure 10을 보면 크기가 5부터 오류가 가장 작아지기 시작하여 11까지 변함이 없는 것을 알 수 있다. 따라서 원래 모형이었던 11개의 노드 모형을 사용하여도 무방하다고 판단된다.



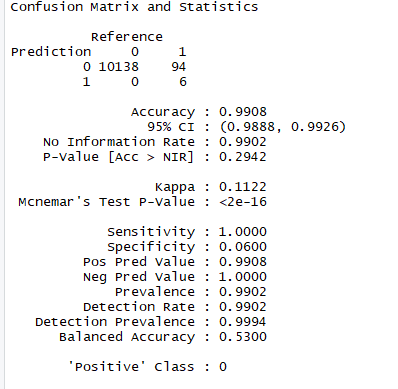
< Figure 10 > deviance를 최소로 하는 terminal node의 개수

앞선 두 모형과 동일하게 의사결정나무 모델로도 ROC curve를 통해 AUC 통계량을 구해 보았다. 의사결정나무의 AUC 통계량은 92.95로 세 가지 모형 중 가장 좋은 성능을 보였다.



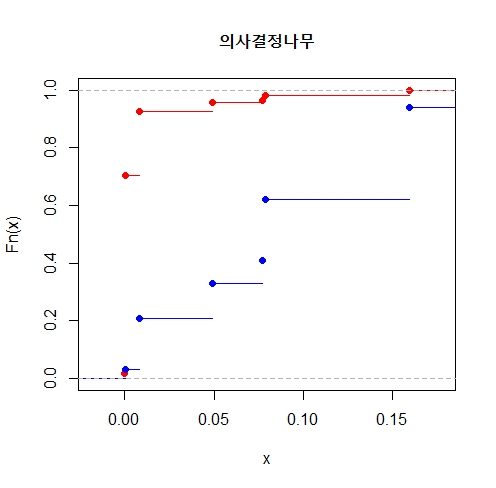
< Figure 11 > 의사결정나무의 ROC curve

confusion Matrix으로 의사결정나무 모형의 정확도를 도출한 결과가 아래의 Figure 12에 나타나 있다. 99.08%로 꽤 높은 수치를 보였지만, confusion Matrix를 통해 보면 불량의심제품의 분류 정확도가 굉장히 떨어지는 것으로 알 수 있다. 양품은 100% 정확히 분류해냈으나 불량의심품은 대부분 맞추지 못하였다.



< Figure 12 > 의사결정나무의 confusion matrix

아래의 Figure 13을 보면, 불량의심품의 분류 정확도는 떨어지고, 양품의 분류 정확도는 꽤 높게 나타났다. 이는 위의 confusion Matrix를 통해서도 확인한 결과이다.



< Figure 13 > 의사결정나무 ecdf 그래프

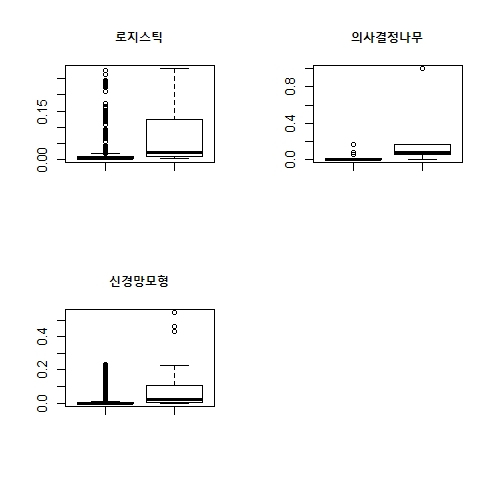
(빨간색 : 양품, 파란색 : 불량품)

1. 모형 성능 비교

아래의 Figure 14는 양품(좌), 불량의심품(우) 그룹에서 각 모형에 의한 예측 확률을 box plot으로 표현한 것이다. 좌측 상자는 0에 가깝고 우측상자는 1에 가까울수록, 한 두 상자의 폭이 좁고 서로 멀리 떨어져 있을수록 분류가 잘 된 것으로 볼 수 있다. 좌측 상자들을 보면, 모형들마다 양품은 불량확률을 비교적 잘 예측하고 있음을 알 수 있다. 세 가지 모형 중에서도 의사결정나무 모형이 양품을 가장 잘 예측하였다.

하지만 불량의심품의 경우에는 모든 분류 모형이 저하된 성능을 보였다. 모든 모형이 비슷하게 좋지 않은 성능을 보이고 있다. 이는 불량의심품의 개수가 300개 밖에 되지 않아 제대로 된 학습을 하지 못하여 모형 결과가 이와 같이 나온 것으로 보인다.

세 가지 모델 중에서 본 분석목표에 가장 적합한 모델을 선택하기 위해서는 어느 모델이 양품 예측보다 불량품 예측성능이 상대적으로 우수한 지를 보아야 한다. 따라서, 불량품 예측을 가장 우수하게 해낸 로지스틱 회귀분석을 적합한 모델로 선택할 수 있다.



< Figure 14 > 각 모델의 성능 비교

1. **분석결과의 활용방안 및 의의**

본 연구에서는 제조공정에서 발생하는 공정데이터를 이용하여 불량항목별 예측을 위한 분석절차와 모형을 제시하였다. 대부분 기업에서는 검사단계에서 불량검사가 이루어지며 검사단계에서 불량이라고 판명이 난 경우는 이미 불량이 만들어지고 난 이후가 된다. 본 연구에서 제시하는 분석절차와 모형을 통해 불량이 만들어지기 전에 공정단계에서 불량을 미리 예측하여 공정이상원인을 해결하는 과정에 소요되는 시간을 단축시킬 수 있을 것이다. 로지스틱 회귀분석에서 선택된 변수는 a속도, 하중시간, 고압시간이고 의사결정나무의 중요 변수로는 mpa, load\_time이라는 것을 알게 되었다.

각 모델들을 평가해 본다면, 상대적으로 더 중요한 불량품 예측율을 기준으로 비교했을 때 로지스틱 회귀분석이 분석목표에 가장 적합한 모델이라고 볼 수 있다.

본 분석결과는 한계는 불량품 데이터의 수가 적어서 정확한 결과를 이끌어 내기에 적절하지 않았다는 것이다. 향후에 다시 분석한다면 불량품 수가 더 많은 데이터로 모델링해서 불량의심품 분류 정확도를 향상시킬 수 있을 것이다.

**6. 분석에 사용된 R 코드 전문**

# 데이터 불러오기

getwd()

car <- read.csv("autoparts.csv")

head(car)

str(car)

# 불량구간 정하기

boxplot.stats(car$c\_thickness)

car <- car[(car$c\_thickness > 0)\*(car$c\_thickness < 1000) ==1,]

quantile(car$c\_thickness, probs=c(0.005, 0.995))

# 분류 준비

car1 <- car[,c(8,9,10,14,15,16,17)]

car1$failure <- as.factor((car1$c\_thickness >39)+(car1$c\_thickness <13))

car1$failure

str(car1$failure)

summary(car1$failure)

# 기존 변수 삭제

car2 <- car1[,-7]

head(car2)

table(car2$failure)

# 훈련자료와 테스트자료로 분류

install.packages("doBy")

library(doBy)

car2.train <- sampleBy(~ failure, frac = 0.7, replace = FALSE, data = car2)

car2.test<-sampleBy(~ failure, frac = 0.3, replace = FALSE, data = car2)

table(car2.train$failure)

table(car2.test$failure)

**## 로지스틱 회귀분석**

install.packages("e1071")

library("e1071")

lm2 <- glm(failure~.,data=car2.train, family='binomial')

summary(lm2)

library(MASS)

vs3 <- stepAIC(lm2,direction="both")

vs3$anova

install.packages("ROCR")

library(ROCR)

fit.log <- predict.glm(vs3,newdata=car2.test,type="response")

pred.log <- prediction(fit.log, car2.test$failure)

perf.log <- performance(pred.log,"tpr","fpr")

plot(perf.log, colorize=T, colorkey.pos="top",

print.cutoffs.at=seq(0,1,by=0.2), text.cex=1,

text.adj=c(-0.5, 0.5), lwd=2)

auc <- performance(pred.log,"auc")

auc

Label0.log <- fit.log[car2.test$failure==0]

Label1.log <- fit.log[car2.test$failure==1]

plot(ecdf(Label0.log),col="red",main="로지스틱")

plot(ecdf(Label1.log),col="blue",add=T)

**## 인공신경망**

install.packages("RSNNS")

library("RSNNS")

Y <- decodeClassLabels(car2.train$failure)

X <- normalizeData(car2.train[,-7])

test.Y <- decodeClassLabels(car2.test$failure)

test.X <- normalizeData(car2.test[,-7])

nn.car <- mlp(X,Y,size=10,maxit=170)

fit.nn <- predict(nn.car,test.X)

pred.nn <- prediction(fit.nn[,2], car2.test$failure)

perf.nn <- performance(pred.nn,"tpr","fpr")

plot(perf.nn, colorize=T, colorkey.pos="top",

print.cutoffs.at=seq(0,1,by=0.2),

text.cex=1,text.adj=c(-0.5, 0.5), lwd=2)

perf.nn1 <- performance(pred.nn,"auc")

perf.nn1

Label0.nn <- fit.nn[,2][car2.test$failure==0]

Label1.nn <- fit.nn[,2][car2.test$failure==1]

plot(ecdf(Label0.nn),col="red",main="신경망모형")

plot(ecdf(Label1.nn),col="blue",add=T)

**## 의사결정나무**

install.packages("tree")

library(tree)

tree.car <- tree(failure~., data=car2.train)

summary(tree.car)

plot(tree.car)

text(tree.car)

set.seed(1)

cv.car<-cv.tree(tree.car, FUN=prune.misclass)

plot(cv.car$size, cv.car$dev, type="b")

library(ROCR)

fit.tree <-predict(tree.car,newdata=car2.test, type="vector")

pred.tree <-prediction(fit.tree[,2], car2.test$failure)

perf.tree <- performance(pred.tree, "tpr","fpr")

plot(perf.tree, colorize=T, colorkey.pos="top", print.cutoffs.at=seq(0,1,by=0.2,text.cex=1, text.adj=c(-0.5,0.5),lwd=2))

perf.tree1<-performance(pred.tree, "auc")

perf.tree1

install.packages("caret")

library("caret")

install.packages("e1071")

library("e1071")

confusionMatrix(factor(1\*(fit.tree[,2]>0.5)), factor(car2.test$failure))

Label1.tree<-fit.tree[,2][car2.test$failure==0]

Label2.tree<-fit.tree[,2][car2.test$failure==1]

plot(ecdf(Label1.tree),col="red", main="의사결정나무")

plot(ecdf(Label2.tree),col="blue", add=T)

**## 모형 성능 비교**

par(mfrow=c(2,2))

boxplot(Label0.log, Label1.log, main="로지스틱")

boxplot(Label1.tree, Label2.tree, main="의사결정나무")

boxplot(Label0.nn, Label1.nn, main="신경망모형")