

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

2021. 12. 02. (목)

김지원

박윤화

김태용





Contents

1. Introduction
2. Data Import
 - (1) 데이터 가져오기
 - (2) 데이터 전처리
 - (3) 데이터 분리
3. 모형 개발 준비
 - (1) Controller
 - (2) Feature Engineering
 - (3) 독립 변수와 종속 변수의 정의
4. 모형 개발
 - (1) 로지스틱 회귀분석
 - (2) 의사결정나무
 - (3) 랜덤 포레스트
5. 모형 Resampling
6. 최종 모형 선정 및 모형 평가
 - (1) Confusion Matrix
 - (2) ROC Curve & AUC

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

Introduction

담당자: 김지원

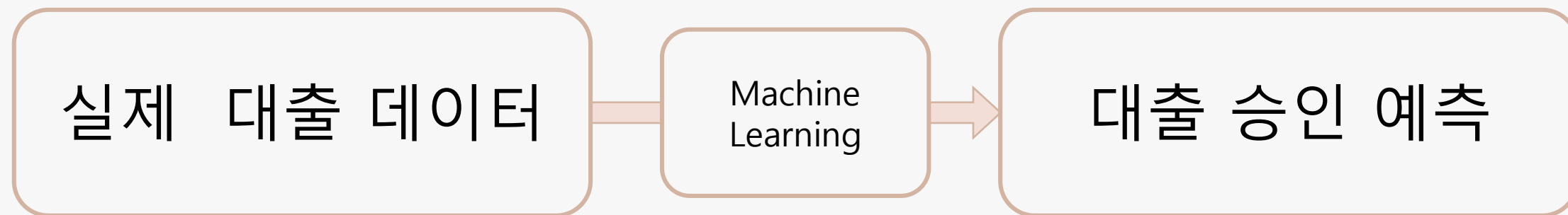
담당자: 김지원



Introduction : Contest

Analytics Vidhya

- [Loan Prediction Practice Problem](#)



Introduction : Language & IDE

Programming Language
- R



IDE
- RStudio



Introduction : R Libraries

caret

- Classification **A**nd **R**egression **T**raining

1. train/test 효율적 분할: createDataPartition()
2. 간편한 전처리: preProcess()
3. 손쉬운 모델 훈련 컨트롤: trainControl()
4. 튜닝 기본 제공 + 추가적 튜닝의 편의성: tuneGrid, tunelength 등
5. 대부분의 모델 지원

doParallel

- Provides a parallel backend

Parallel: for processing speed

tidyverse

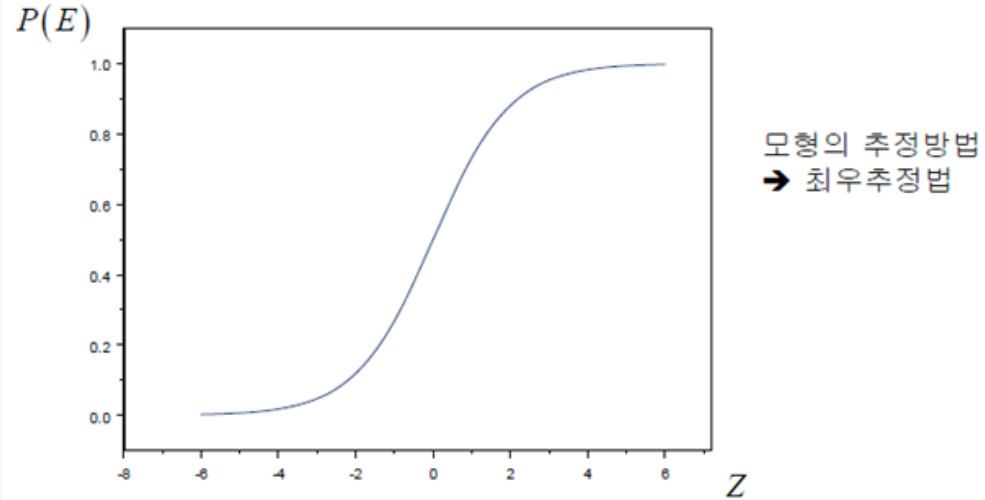
- collection of R packages(for data science)

ROCR

- visualizing classifier performance in R

Introduction : Logistic Regression

사건E의 발생확률과 독립변수 선형 결합간의 관계



[이항 로지스틱회귀 모형]

집단에 속할 확률

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

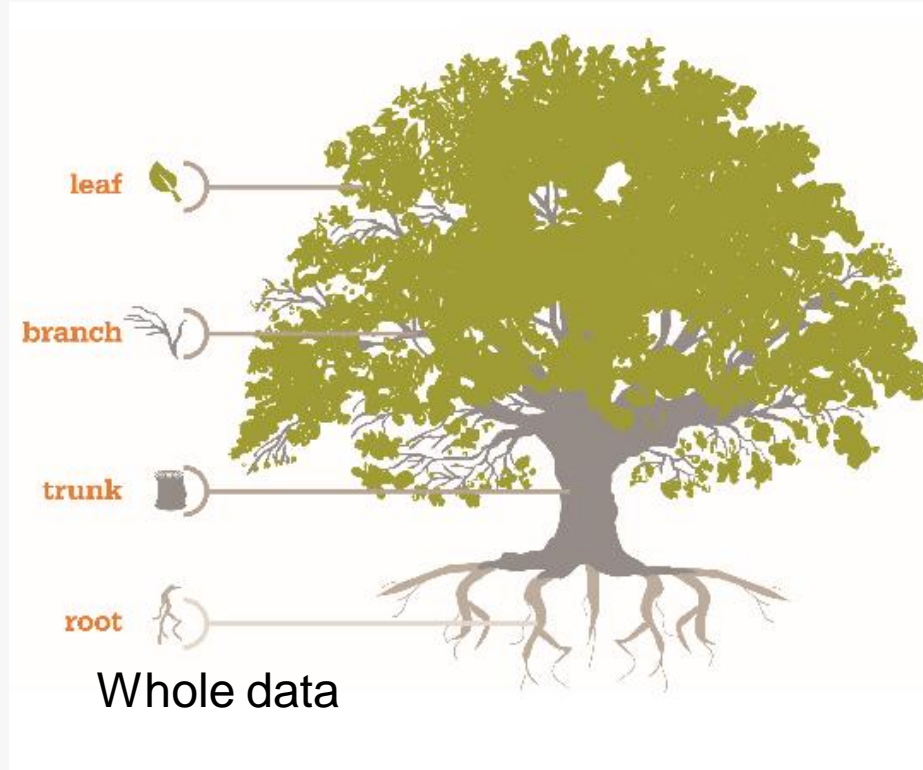
logit

속하지 않을 확률

분석하고자 하는 대상들이 두개 이상의 집단(다변수 데이터)으로 나누어진 경우, 개별 관측치들이 어느 집단으로 분류될 수 있는가를 분석하고 이를 예측하는 모형 개발을 위한 통계 알고리즘

- 이항 로지스틱 분석 : 분석 대상 2그룹
- 다항로지스틱 분석 : 분석 대상 2그룹 이상

Introduction : Decision Tree



의사결정나무(Decision Tree)는 의사결정규칙(Decision Rule)을 나무 구조로 도표화하여 분류와 예측을 수행하는 분석 방법

Decision Tree Analysis step

1

의사결정나무의 형성

분석의 목적과 자료구조에 따라 적절한 분리기준(Split)과 정지규칙(Stopping Rule)을 지정하여 의사결정나무를 얻는다.

2

가지치기

분류오류(Classification Error)를 크게 할 위험(Risk)이 높거나 부적절한 추론규칙(Induction Rule)을 가지고 있는 가지(Branch)를 제거한다

3

타당성 평가

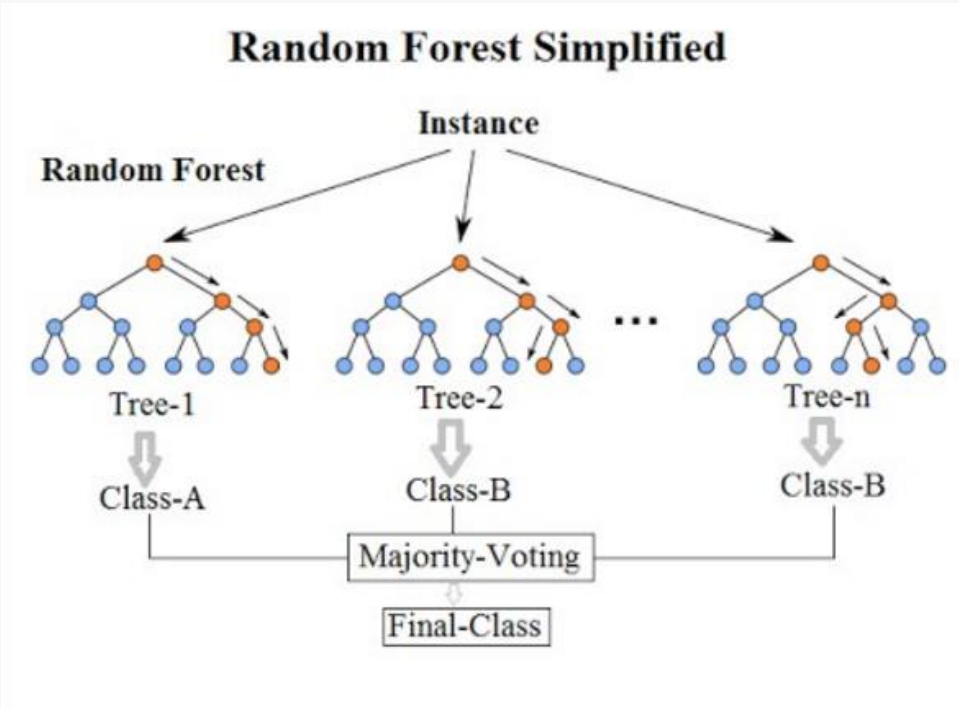
이익도표(Gains Chart)나 위험도표(Risk Chart)또는 검증용 자료(Test Data)에 의한 교차타당성(Cross Validation) : 등을 이용하여 의사결정나무를 평가

4

해석 및 예측

의사결정나무를 해석하고 예측 모형을 설정

Introduction : Random forest regression



A type of machine learning that randomly creates and compares characteristics when creating a Decision Tree.

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

Data Import

담당자: 김태용



Data

read csv data file

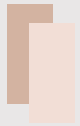
- cleaned_loan_data.csv

```
loan_data <- read.csv( "data/cleaned_loan_data.csv" , stringsAsFactors = FALSE)
```

```
## [1] 29091      8
```

Total : 29091ea, Column :8

loan_status	Loan approved (0 – false / 1 – true)
loan_amnt	Loan amount (\$)
grade	Credit rating
home_ownership	Holding type of house
annual_inc	Annual income (\$)
age	Age who applying loan
emp_cat	Employed year
ir_cat	Interest rate (%)



Data pre-processing

Remove non-unique rows

- distinct

```
loan_data2 <- loan_data %>% distinct()
```

Change column type

- columns type & data value

```
loan_data2$loan_status <- factor(loan_data2$loan_status, levels = c(0, 1), labels = c("default", "Approval"))  
loan_data2$grade <- as.factor(loan_data2$grade)  
loan_data2$home_ownership <- as.factor(loan_data2$home_ownership)
```

Data glimpse

Data: before pre-processing

- loan_data

```
glimpse(loan_data)
```

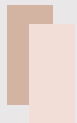
```
## Rows: 29,091
## Columns: 8
## $ loan_status    <int> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0~
## $ loan_amnt      <int> 5000, 2400, 10000, 5000, 3000, 12000, 9000, 3000, 10000~
## $ grade          <chr> "B", "C", "C", "A", "E", "B", "C", "B", "B", "D", "C", ~
## $ home_ownership <chr> "RENT", "RENT", "RENT", "RENT", "RENT", "OWN", "RENT", ~
## $ annual_inc     <dbl> 24000.00, 12252.00, 49200.00, 36000.00, 48000.00, 75000~
## $ age            <int> 33, 31, 24, 39, 24, 28, 22, 22, 28, 22, 23, 27, 30, 24,~
## $ emp_cat        <chr> "0-15", "15-30", "0-15", "0-15", "0-15", "0-15", "0-15"~
## $ ir_cat         <chr> "08월 11일", "Missing", "11-13.5", "Missing", "Missing"~
```

Data: after pre-processing

- loan_data2

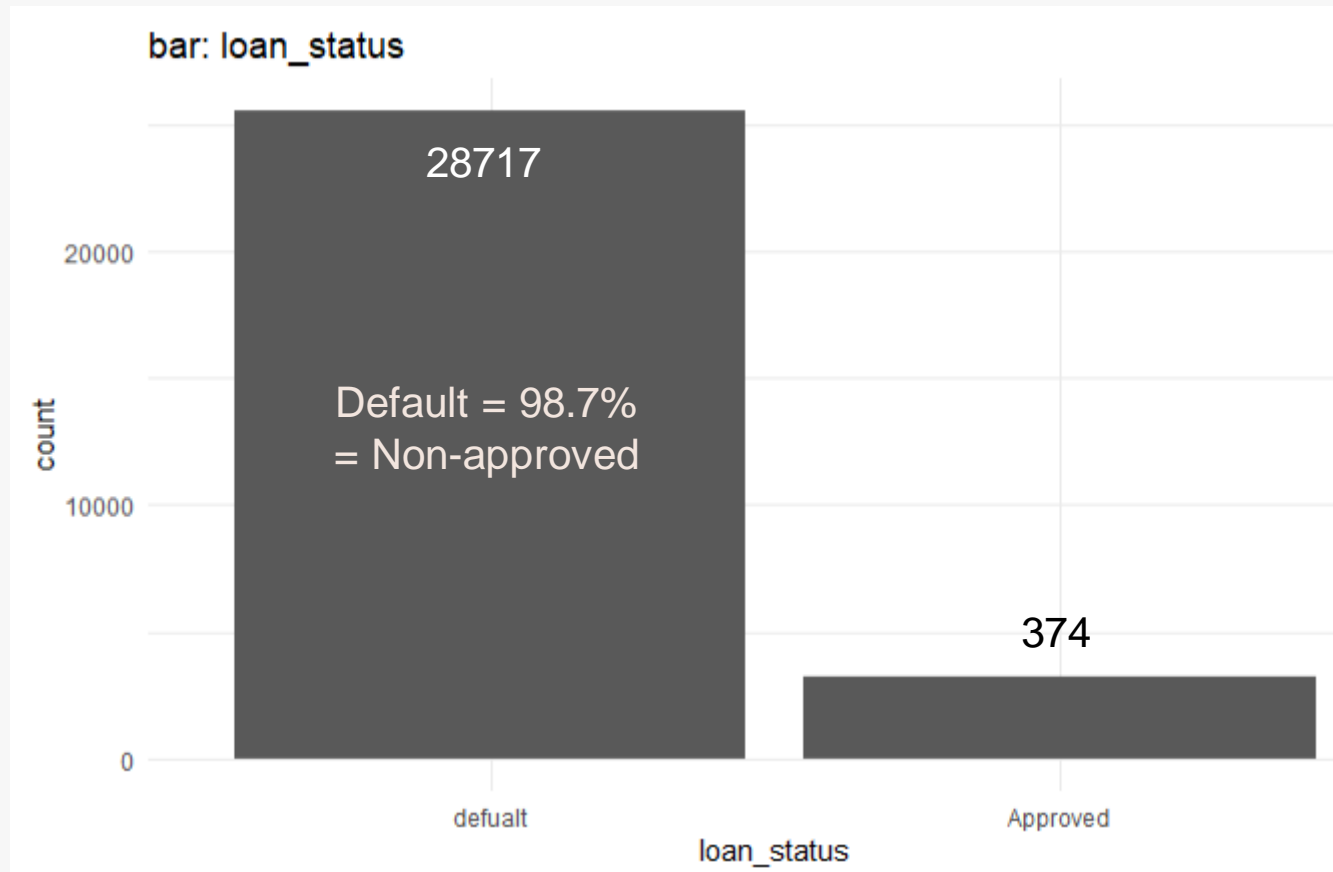
```
glimpse(loan_data2)
```

```
## Rows: 28,717
## Columns: 8
## $ loan_status    <fct> default, default, default, default, default, default, A~
## $ loan_amnt      <int> 5000, 2400, 10000, 5000, 3000, 12000, 9000, 3000, 10000~
## $ grade          <fct> B, C, C, A, E, B, C, B, B, D, C, A, B, A, B, B, B, B~
## $ home_ownership <fct> RENT, RENT, RENT, RENT, RENT, OWN, RENT, RENT, RENT, RE~
## $ annual_inc     <dbl> 24000.00, 12252.00, 49200.00, 36000.00, 48000.00, 75000~
## $ age            <int> 33, 31, 24, 39, 24, 28, 22, 22, 28, 22, 23, 27, 30, 24,~
## $ emp_cat        <chr> "0-15", "15-30", "0-15", "0-15", "0-15", "0-15", "0-15"~
## $ ir_cat         <chr> "08월 11일", "Missing", "11-13.5", "Missing", "Missing"~
```



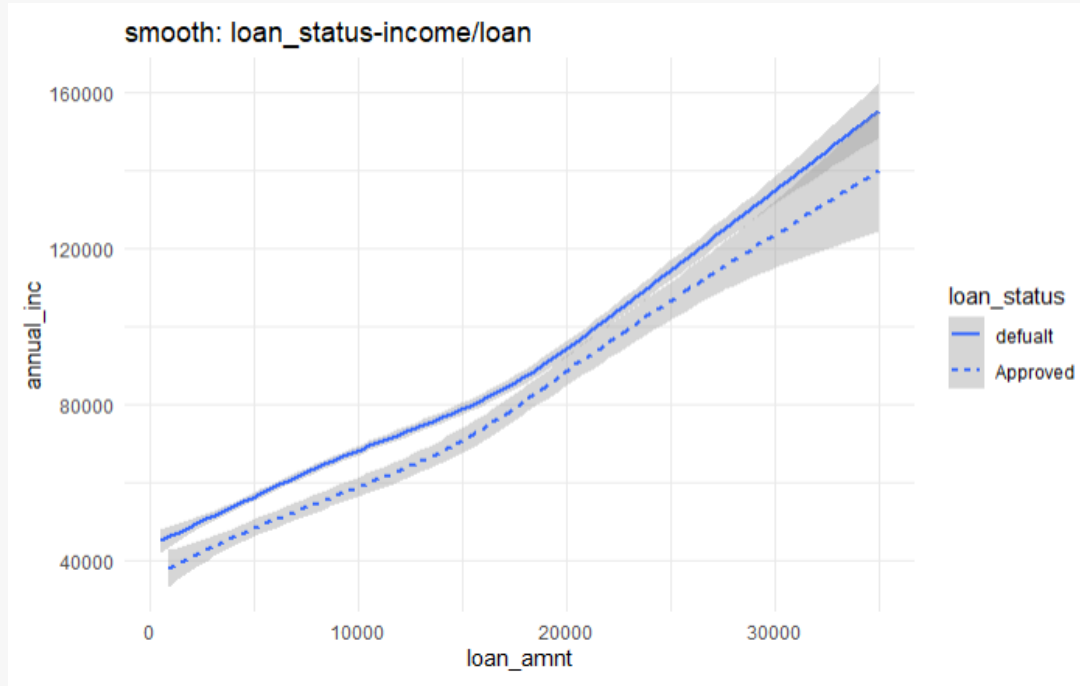
Data EDA

대출 승인과 대출 승인이 나지 않은 column 확인

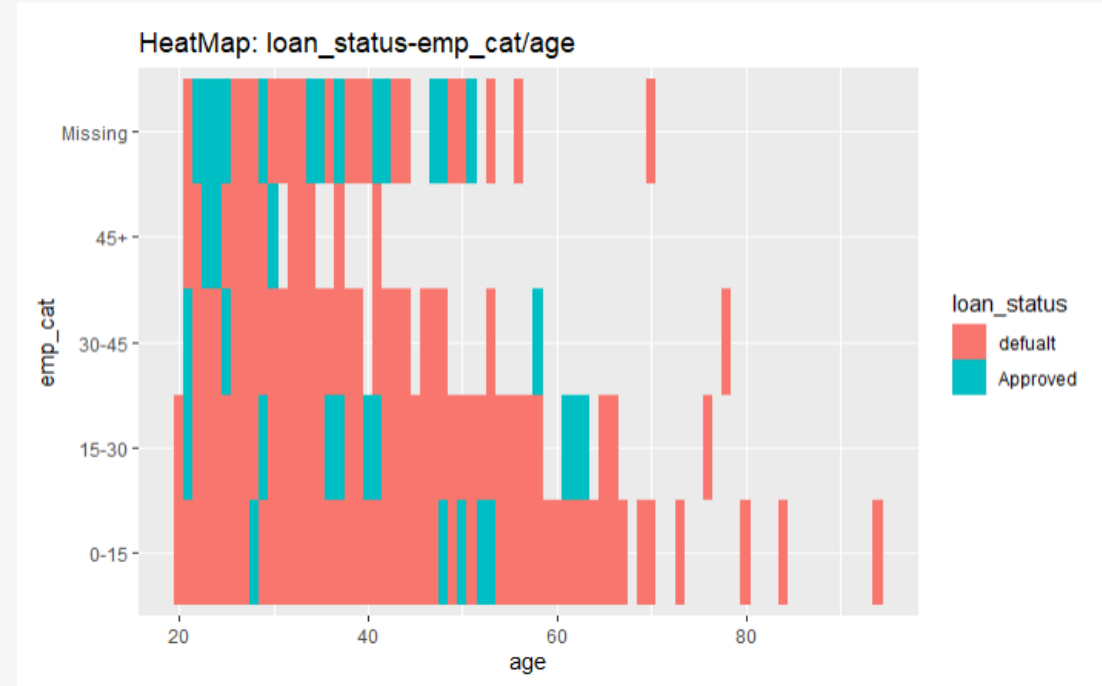


Data EDA

대출 승인과 age :



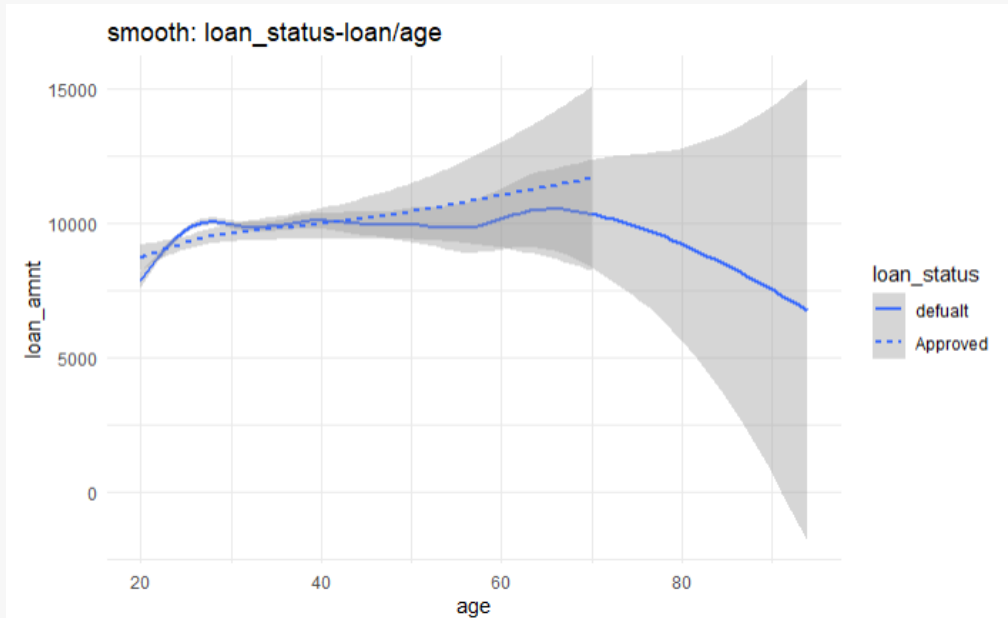
연봉이 높다고 대출 승인이 잘 나는 것이 아니다.
연봉이 높으면, 많은 양의 대출을 할 수 있다.



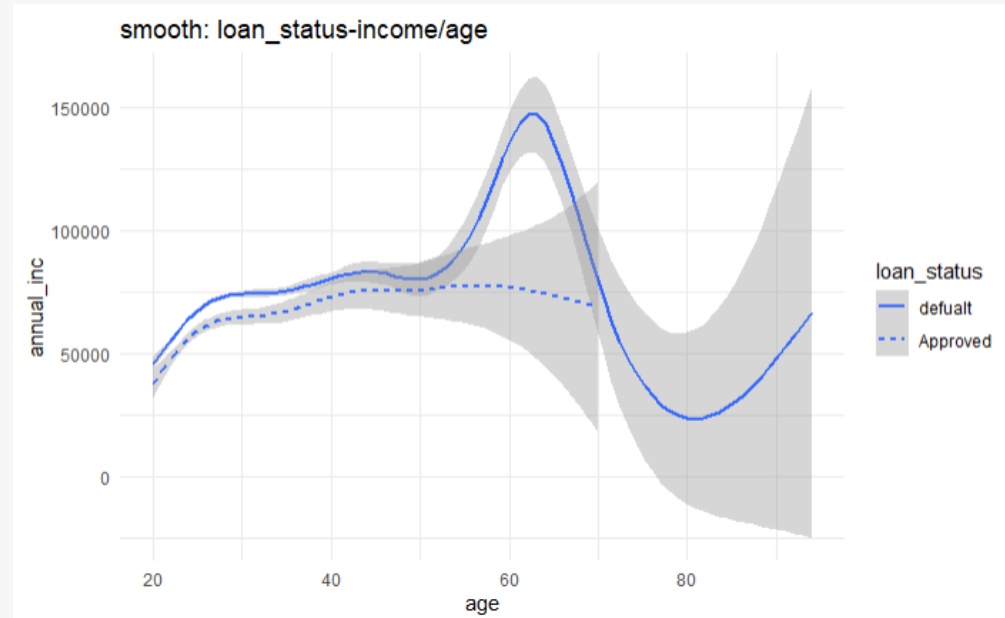
근속 연수가 높다고 대출 승인이 잘 되는 것은 아니다.

Data EDA

대출 승인과 age :



70세 이상은 대출 승인이 나지 않고, 대출의 양의 경우 나이가 증가 할 수록 분산이 증가 하는 것을 볼 수 있다.

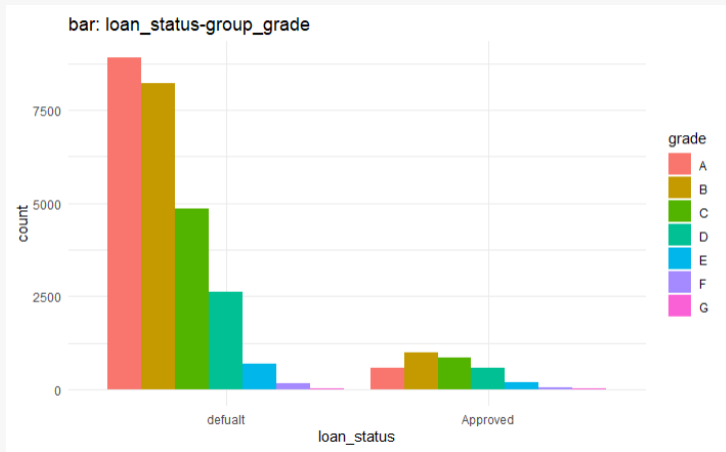


연봉이 높다고 대출이 잘 되는 것은 아니다.

Data EDA

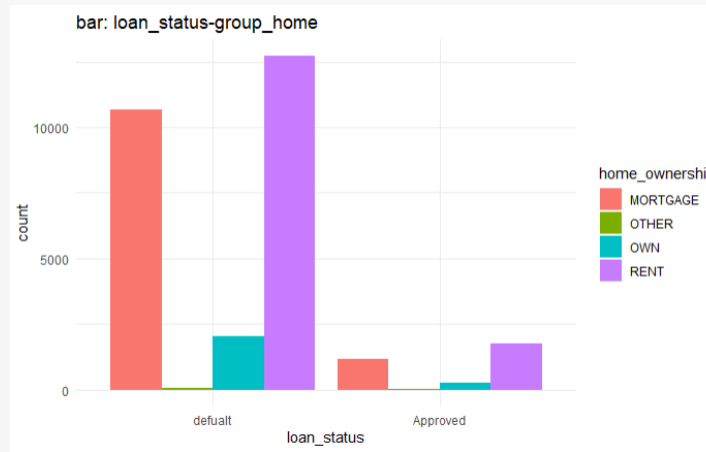
대출 승인과 기타 요인들 :

대출승인-신용등급



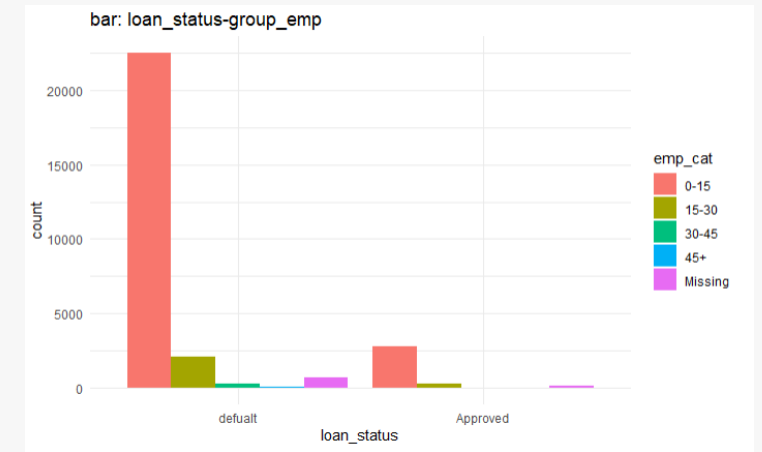
신용등급이 높다고 대출 승인이 잘 되는 것은 아니다. (B, C 가 A 보다 높다.)

대출승인-주거상태



집을 가지고 있다고 대출 승인이 잘 나는 것은 아니다. (OWN < RENT)

대출승인-고용기간



고용기간의 경우 역시 기간이 높다고 해서 승인이 잘 나는 것은 아니다.

Data pre-treatments

```
sapply(loan_data, function(x) sum(is.na(x)))
```

```
##   loan_status   loan_amnt      grade home_ownership   annual_inc  
##         0         0         0         0         0  
##      age      emp_cat      ir_cat  
##         0         0         0
```

- 데이터 타입을 확인한다.

```
loan_data %>% duplicated() %>% sum() # 374개 확인
```

```
## [1] 374
```

```
loan_data2 <- loan_data %>% distinct()
```

- 데이터 타입을 확인한다.

```
glimpse(loan_data2)
```

```
## Rows: 28,717  
## Columns: 8  
## $ loan_status   <int> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0~  
## $ loan_amnt     <int> 5000, 2400, 10000, 5000, 3000, 12000, 9000, 3000, 10000~  
## $ grade         <chr> "B", "C", "C", "A", "E", "B", "C", "B", "B", "D", "C", ~  
## $ home_ownership <chr> "RENT", "RENT", "RENT", "RENT", "RENT", "OWN", "RENT", ~  
## $ annual_inc    <dbl> 24000.00, 12252.00, 49200.00, 36000.00, 48000.00, 75000~  
## $ age          <int> 33, 31, 24, 39, 24, 28, 22, 22, 28, 22, 23, 27, 30, 24,~  
## $ emp_cat       <chr> "0-15", "15-30", "0-15", "0-15", "0-15", "0-15", "0-15"~  
## $ ir_cat        <chr> "8-11", "Missing", "11-13.5", "Missing", "Missing", "11~
```

경로를 확인한 뒤 데이터를 가져온다.

먼저 중복 값을 확인한다

Chr factors가 보인다.

➤ grade, home_ownership, emp_cat, ir_cat

Data pre-treatments

- 우선 타겟 데이터는 영어로 표현한다.

```
loan_data2$loan_status <- factor(loan_data2$loan_status, levels = c(0, 1), labels = c("non_default", "default"))
loan_data2$grade <- as.factor(loan_data2$grade)
loan_data2$home_ownership <- as.factor(loan_data2$home_ownership)
```

- 만약 한꺼번에 하고 싶다면 다음과 같이 할 수 있다.

```
loan_data2 <- loan_data2 %>%
  mutate_if(is.character, as.factor)
```

- chr 데이터가 모두 factor로 바뀌었는지 확인한다.

```
glimpse(loan_data2)
```

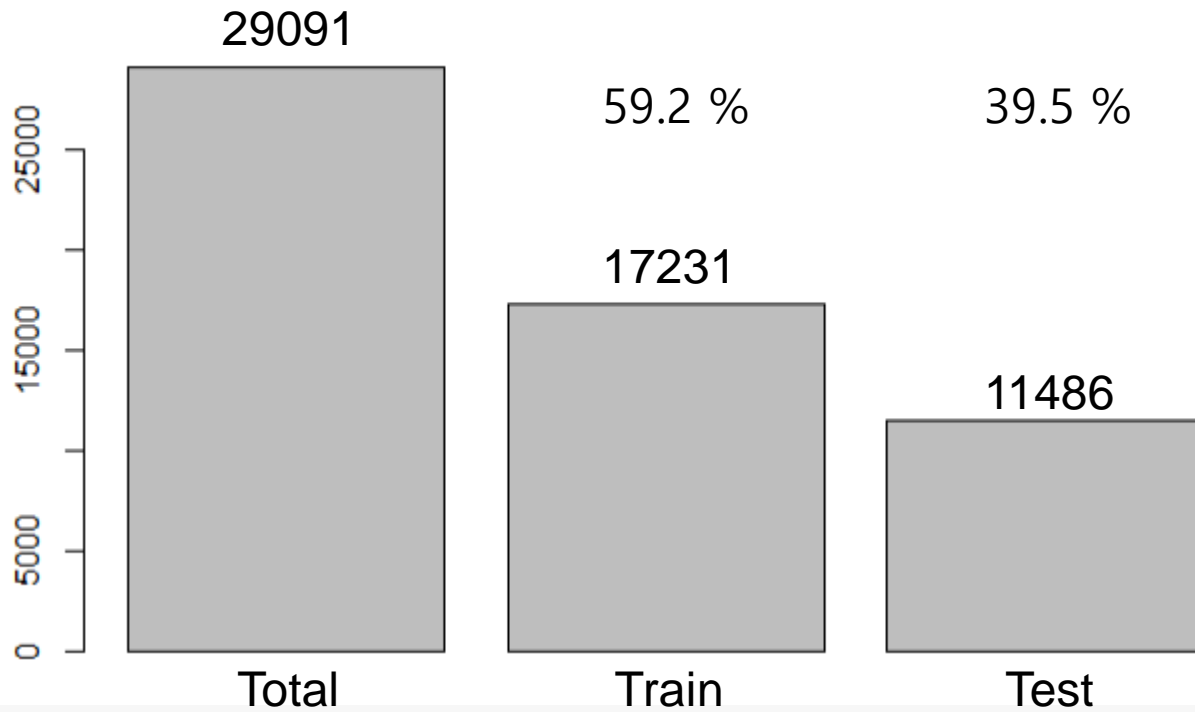
```
## Rows: 28,717
## Columns: 8
## $ loan_status    <fct> non_default, non_default, non_default, non_default, non~
## $ loan_amnt      <int> 5000, 2400, 10000, 5000, 3000, 12000, 9000, 3000, 10000~
## $ grade          <fct> B, C, C, A, E, B, C, B, B, D, C, A, B, A, B, B, B, B~
## $ home_ownership <fct> RENT, RENT, RENT, RENT, RENT, OWN, RENT, RENT, RENT, RE~
## $ annual_inc     <dbl> 24000.00, 12252.00, 49200.00, 36000.00, 48000.00, 75000~
## $ age            <int> 33, 31, 24, 39, 24, 28, 22, 22, 28, 22, 23, 27, 30, 24,~
## $ emp_cat        <fct> 0-15, 15-30, 0-15, 0-15, 0-15, 0-15, 0-15, 0-15, 0-15, ~
## $ ir_cat         <fct> 8-11, Missing, 11-13.5, Missing, Missing, 11-13.5, 11-1~
```

Chr data를 factor로 바꿔준다.

Data pre-treatments

- 훈련 데이터와 테스트 데이터로 분리한다.

```
set.seed(2021)
inx  <- createDataPartition(loan_data2$loan_status, p = 0.6, list = F)
train <- loan_data2[ inx, ]
test  <- loan_data2[-inx, ]
```



전체 data를 6:4 정도의 비율로 나누어 머신러닝 진행

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

머신러닝 모형 개발 준비

담당자: 박윤화



머신러닝 모형 개발 _Controller

TrainControl 함수를 활용하여 기본 세팅을 진행 한다.

✓ trainControl 함수: 모델을 생성하기위한 매개 변수들을 생성

3개의 분리된
10배 교차검증

```
control <- trainControl(  
  method = "repeatedcv",  
  number = 10, # 10겹  
  repeats = 3, # 3번  
  search = "grid",  
  classProbs = TRUE)
```

method:

the resampling method : repeatedcv

"boot", "cv", "LOOCV", "LGOCV", "repeatedcv", "timeslice", "none" and "oob"

Number:

K-폴드 교차 검증의 접힘 횟수 또는 부트스트래핑 및 그룹 아웃 교차 검증을 위한 리샘플링 반복 횟수

Repeats:

반복 k-접힘 교차 검증 전용: 계산할 전체 접힘 집합 수

Search:

the Grid Search (Random 방법도 있다.)

직접 Hyperparameter의 범위를 지정 해 주는 방법

ClassProbs:

True

각 대표본에서 (예측값과 함께) 분류 모델에 대해 클래스 확률을 계산해야 하는가?에대한 대답

머신러닝 모형 개발 _Feature Engineering

통계처리를 진행한다. for Normalization

```
preProc <- c("BoxCox",  
Data를 0~1에 놓이게 하여  
Normalize 해줌. "center",  
"scale",  
"spatialSign",  
"corr",  
"zv")
```

정규 분포와 매우 유사하도록 데이터를 변환, 신뢰 구간을 구성하고 가설 검정을 수행 가능

각 값을 평균으로 빼 값

각 값을 평균으로 빼 준 후 표준 편차로 나눠 준 값 (center = T, scale = T 일 때.)

공간 계수 계산 (데이터 벡터를 단위 길이 원에 투영)

가중 형태의 상관 계수를 계산합니다.

zero variance predictors : 그룹 내에서 희소성이 있는 x 열을 식별

독립변수와 종속 변수를 정의한다.

```
frml <- loan_status ~ loan_amnt + grade + home_ownership + annual_inc + age + emp_cat + ir_cat
```

독립변수

종속변수

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

머신러닝 모형 개발

담당자: 박윤화



Logistic Regression

개발 준비가 끝났다면, 다양한 모델을 개발하도록 한다.

```
logis <- train(
  frm1,           앞에서 정해진 독립변수와 종속변수
  data = train,   Data set
  method = "glm", generalized linear model : 텍스트와 숫자가 섞여 있는 구조에서 회기 분석
  metric = "Accuracy", 최적의 모델을 선택하는 데 사용할 요약 Metric을 지정
  trControl = control, 앞에서 한 기본 세팅 로딩
  preProcess = preProc, 앞에서 한 Normalization
)
```

logis

regression : "RMSE " and "Rsquared"
classification : "Accuracy" (1에 가까울 수록 좋음) and "Kappa" (일치도 계산)

```
## Generalized Linear Model
##
## 17231 samples
##      7 predictor
##      2 classes: 'non_default', 'default'
##
## Pre-processing: Box-Cox transformation (3), centered (20), scaled (20),
## spatial sign transformation (20)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 15509, 15508, 15508, 15507, 15508, 15507, ...
## Resampling results:
##
##      Accuracy   Kappa
##      0.8878377  -0.0004200657
```

Accuracy : 예측한 값 중 정확히 예측한 값의 비율

의사결정나무

의사결정 나무에서 hyperParameter를 정의한다.

hyperParameter : machineLearning에서 Learning procee를 control 하는데 사용되는 값을 갖는 매개 변수로써 가장 좋은 변수

Data Frame 생성

```
rpartGrid <- expand.grid(cp = c(0.001, 0.005, 0.01))  
modelLookup("rpart")
```

모델 및 패키지에 대한 정보

##	model	parameter	label	forReg	forClass	probModel
## 1	rpart	cp	Complexity Parameter	TRUE	TRUE	TRUE

✓ **rpart Model:** Na 값을 대리 변수(surrogate variable)로 처리 해줌 (확률적으로 높은 변수로)

rpart는 잘알려진 CART(Classification and Regression Trees)의 아이디어를 구현한 패키지

원래는 Random hyperparameter로 찾아야 하지만, 지금은 연습이기 때문에 범위를 지정하여 가장 좋은 parameter를 찾는다.
<http://topepo.github.io/caret/random-hyperparameter-search.html>

머신러닝 모형 개발

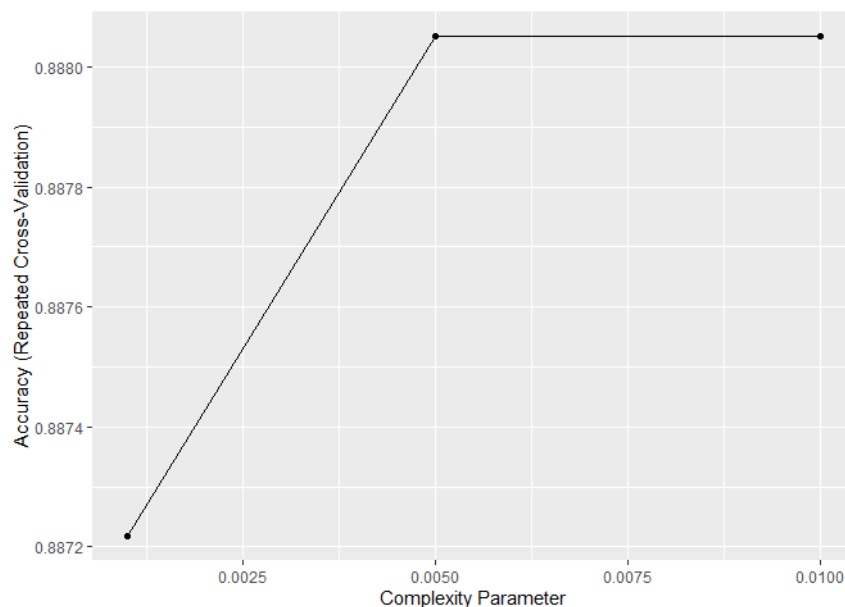
의사결정 나무 모델을 개발한다.

```
set.seed(2021) 유사 난수 생성
rpt <- train(
  frml,
  data = train,
  method = "rpart",
  metric = "Accuracy",
  trControl = control,
  preProcess = preProc,
  tuneGrid = rpartGrid) 각 값을 자동으로 넣어지는 tuneGrid 함수 사용
```

rpt

```
## CART
##
## 17231 samples
## 7 predictor
## 2 classes: 'non_default', 'default'
##
## Pre-processing: Box-Cox transformation (3), centered (20), scaled (20),
## spatial sign transformation (20)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 15508, 15507, 15508, 15508, 15508, 15508, ...
## Resampling results across tuning parameters:
##
##   cp      Accuracy      Kappa
##   ---      ---
## 0.001 0.8872189 0.008546392
## 0.005 0.8880506 0.000000000
## 0.010 0.8880506 0.000000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.01.
```

ggplot(rpt)



Random Forest

이번에는 Random Forest를 사용하기 위한 Hyperparameter를 정의한다.

Data Frame 생성

```
rfGrid <- expand.grid(mtry = c(3, 4, 5))  
modelLookup("rf")
```

##	model	parameter	label	forReg	forClass	probModel
## 1	rf	mtry #Randomly Selected Predictors	TRUE	TRUE	TRUE	

mtry는 각각의 tree마다 몇 개의 feature를 사용할 것인지를 정하는 것

Random Forest

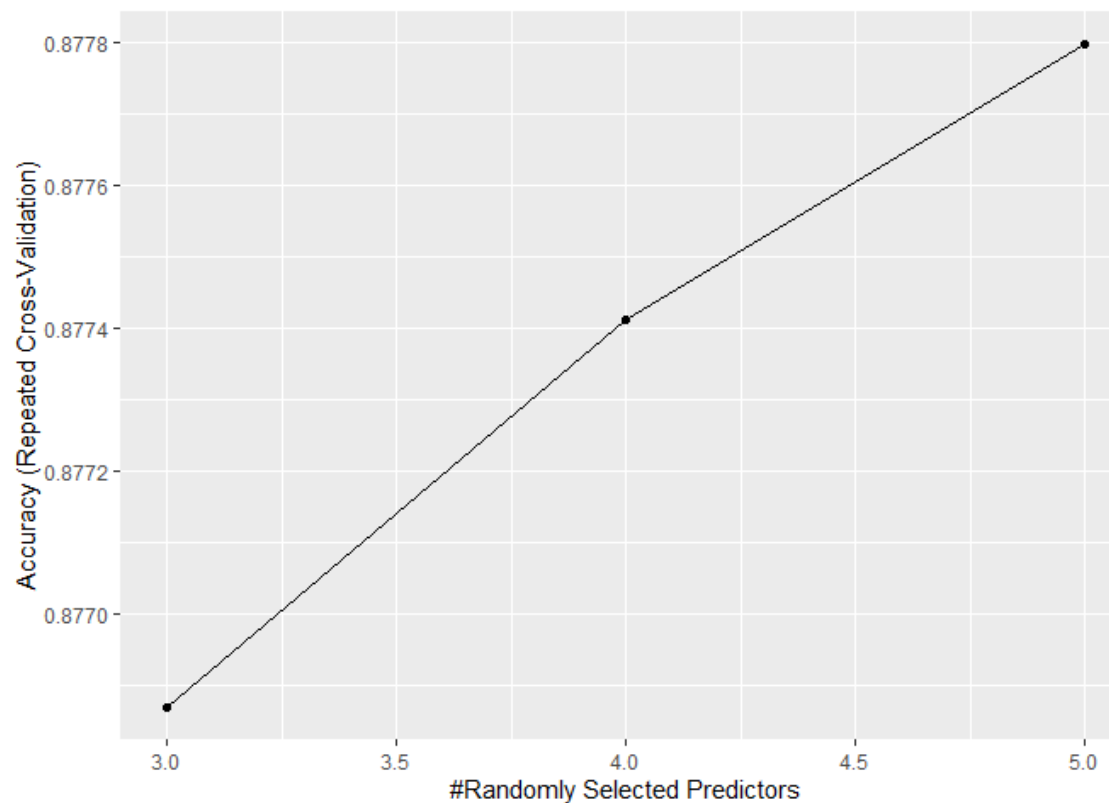
Random Forest 모델을 개발한다.

```
rf <- train(
  frml,
  data = train,
  method = "rf",
  metric = "Accuracy",
  trControl = control,
  preProcess = preProc,
  tuneGrid = rfGrid
)

rf
```

```
## Random Forest
##
## 17231 samples
##    7 predictor
##    2 classes: 'non_default', 'default'
##
## Pre-processing: Box-Cox transformation (3), centered (20), scaled (20),
## spatial sign transformation (20)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 15508, 15508, 15507, 15508, 15509, 15508, ...
## Resampling results across tuning parameters:
##
##    mtry  Accuracy  Kappa
##    3     0.8768692  0.005206944
##    4     0.8774109  0.005965678
##    5     0.8777978  0.008642398
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 5.
```

ggplot(rf)



Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

Model Resampling

담당자: 김태용



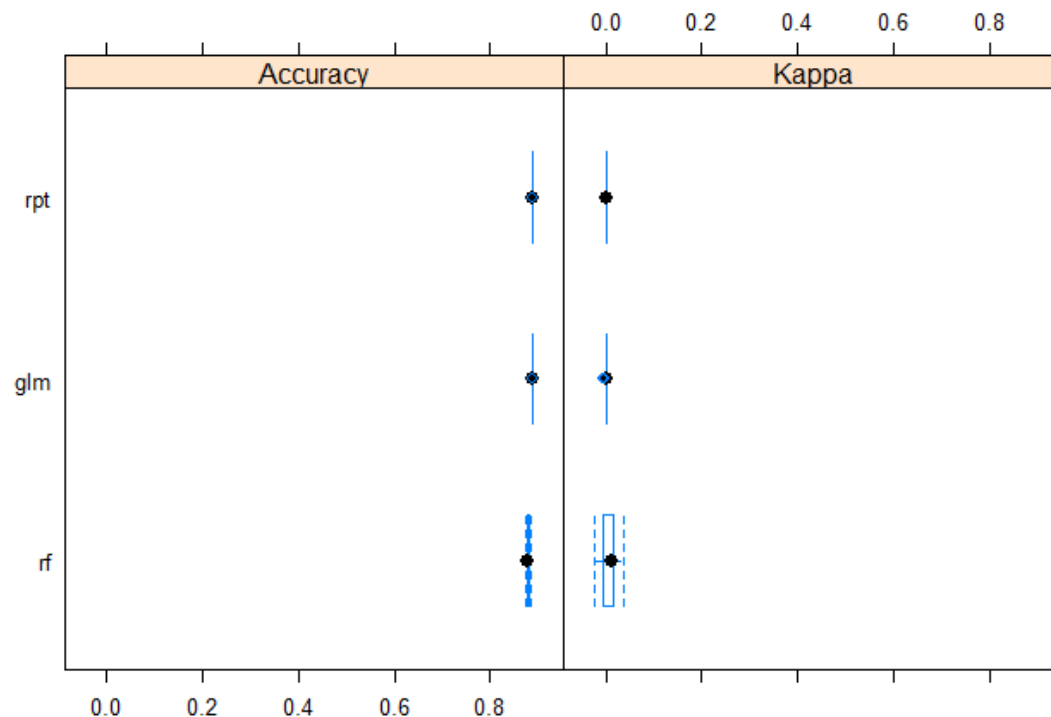
모형 비교

3개의 모형을 비교하도록 한다.

```
resamps <- resamples(  
  list(glm = logis,  
        rpt = rpt,  
        rf = rf))  
  
summary(resamps)
```

```
##  
## Call:  
## summary.resamples(object = resamps)  
##  
## Models: glm, rpt, rf  
## Number of resamples: 30  
##  
## Accuracy  
##      Min.    1st Qu.  Median    Mean   3rd Qu.    Max. NA's  
## glm 0.8861789 0.8879861 0.8879861 0.8878377 0.8879861 0.8885017  0  
## rpt 0.8879861 0.8879861 0.8879861 0.8880506 0.8879861 0.8885665  0  
## rf  0.8723157 0.8758701 0.8772134 0.8777978 0.8797156 0.8839234  0  
##  
## Kappa  
##      Min.    1st Qu.  Median    Mean   3rd Qu.    Max.  
## glm -0.004571755 0.000000000 0.0000000 -0.0004200657 0.00000000 0.00000000  
## rpt 0.000000000 0.000000000 0.0000000 0.0000000000 0.00000000 0.00000000  
## rf -0.024423256 -0.003628695 0.0129503 0.0086423981 0.01672265 0.03835002  
## NA's  
## glm 0  
## rpt 0  
## rf  0
```

```
bwplot(resamps, layout = c(2, 1))
```



glm, rpt가 Accuracy 값은 높지만, Kappa median 값이 0 이기 때문에, rf를 가져가되, Accuracy 값을 조정 해 보자.

Loan Risk Analysis with Machine Learning Classification

Logistic Regression & Decision Tree & random forest

최종모형 선정 및 모형평가

담당자: 김지원



(1) Confusion Matrix(오차 행렬)

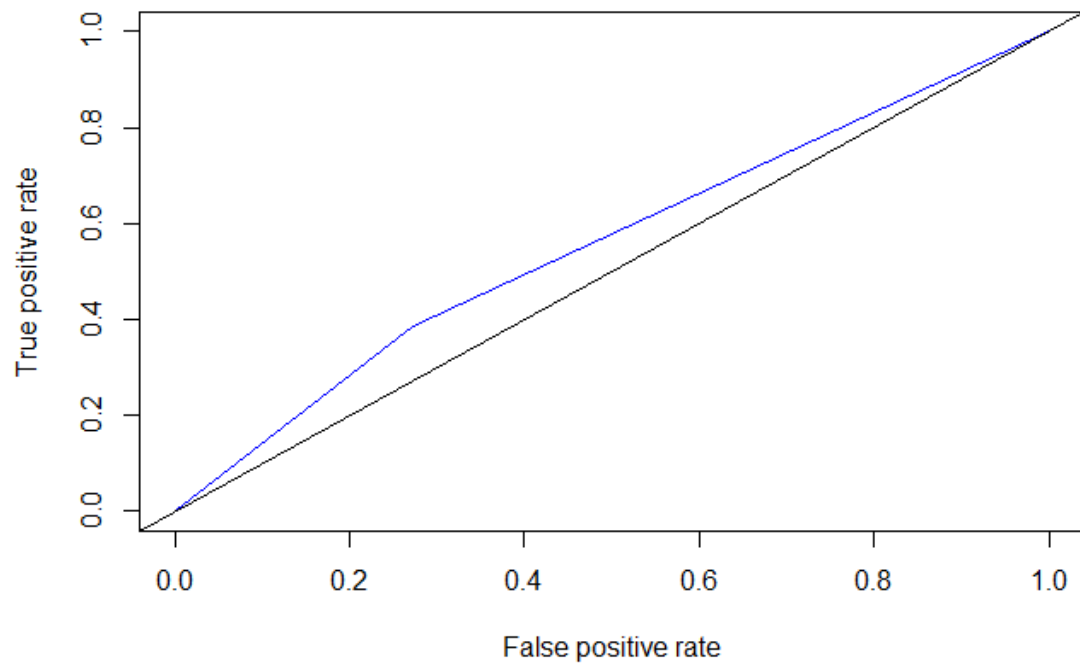
```
pred_rpt <- predict(rf, test, type = "prob")
pred_rpt$loan_status <- ifelse(pred_rpt$non_default > 0.85, 0, 1) # cut-off를 조정하며 맞춰보자
pred_rpt$loan_status <- factor(pred_rpt$loan_status, levels = c(0, 1), labels = c("non_default", "default"))
confusionMatrix(pred_rpt$loan_status, test$loan_status, positive = "non_default")
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  non_default default
## non_default      7417      791
## default         2783      495
##
##              Accuracy : 0.6888
##              95% CI : (0.6803, 0.6973)
##      No Information Rate : 0.888
##      P-Value [Acc > NIR] : 1
##
##              Kappa : 0.0668
##
## Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.7272
##              Specificity : 0.3849
##      Pos Pred Value : 0.9036
##      Neg Pred Value : 0.1510
##      Prevalence : 0.8880
##      Detection Rate : 0.6457
##      Detection Prevalence : 0.7146
##      Balanced Accuracy : 0.5560
##
##      'Positive' Class : non_default
##
```

더 안 좋아졌다.

(2) ROC Curve & AUC

```
library(ROCR)
pr <- prediction(as.numeric(pred_rpt$loan_status) - 1, as.numeric(test$loan_status) - 1)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, col = "blue")
abline(a = 0, b = 1)
```



```
# AUC = Area Under Curve의 뜻으로
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.5560357
```

참고문헌

Introduction

doParallel <https://freshrimpsushi.tistory.com/1266>

의사 결정 나무 <https://wikidocs.net/73510>
<https://sungwookkang.com/275>

Logistic regression <https://stats.idre.ucla.edu/r/dae/logit-regression/>

Controller <https://www.rdocumentation.org/packages/caret/versions/6.0-90/topics/trainControl>

통계 https://vuquangnguyen2016.files.wordpress.com/2018/03/applied-predictive-modeling-max-kuhn-kjell-johnson_1518.pdf

Glm <https://agronomy4future.com/2021/08/09/r%EC%9D%84-%EC%9D%B4%EC%9A%A9%ED%95%B4%EC%84%9C-general-linear-model-glm-%EC%9D%BC%EB%B0%98%EC%84%A0%ED%98%95%EB%AA%A8%EB%8D%B8%EC%9D%84-%EB%B6%84%EC%84%9D%ED%95%B4-%EB%B3%B4%EC%9E%90-2-2/>

머신러닝 모형개발

Logistic Rgression <https://www.rdocumentation.org/packages/caret/versions/4.47/topics/train>

Expend grid <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/expand.grid>

hyperParameter http://r4pda.co.kr/pdf/r4pda_2014_03_02.pdf

Set Seed in R <https://r-coder.com/set-seed-r/>

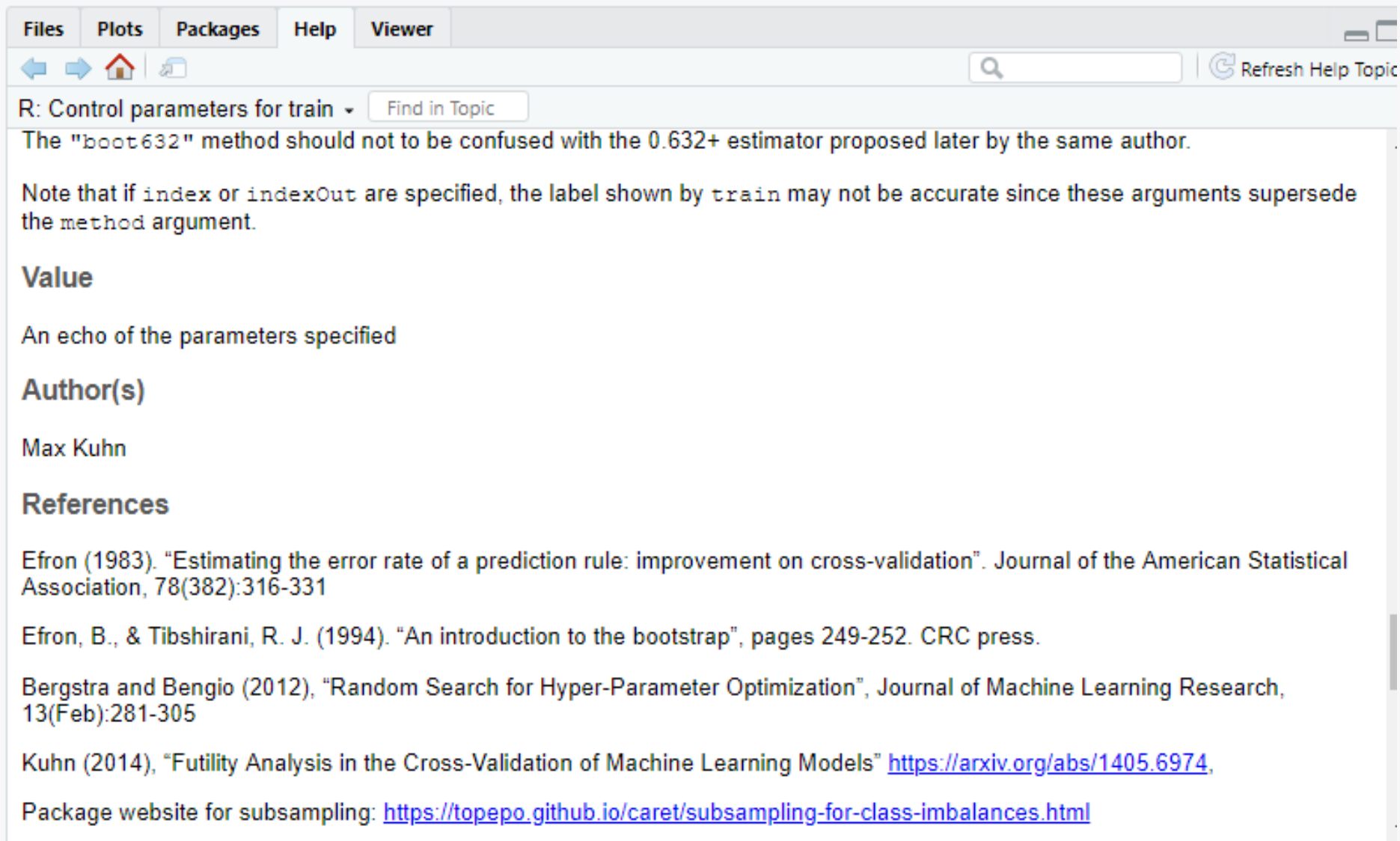
Cran.r <https://cran.r-project.org/web/packages/caret/caret.pdf>

데이터 탐색

참고문헌

```
00 rpart.plot(rpt$f
01 ?trainControl
02 -
```

이렇게 찾아보면 더 쉽다.



The screenshot shows the R help viewer interface. The top bar includes tabs for Files, Plots, Packages, Help, and Viewer. Below the tabs is a search bar and a 'Refresh Help Topic' button. The main content area displays the help text for the `trainControl` function. The title is 'R: Control parameters for train'. A note states: 'The "boot632" method should not be confused with the 0.632+ estimator proposed later by the same author.' Another note says: 'Note that if `index` or `indexOut` are specified, the label shown by `train` may not be accurate since these arguments supersede the `method` argument.' The 'Value' section describes it as 'An echo of the parameters specified'. The 'Author(s)' section lists 'Max Kuhn'. The 'References' section lists several academic papers: Efron (1983), Efron & Tibshirani (1994), Bergstra and Bengio (2012), and Kuhn (2014). At the bottom, it provides a package website for subsampling.

R: Control parameters for train Find in Topic

The "boot632" method should not be confused with the 0.632+ estimator proposed later by the same author.

Note that if `index` or `indexOut` are specified, the label shown by `train` may not be accurate since these arguments supersede the `method` argument.

Value

An echo of the parameters specified

Author(s)

Max Kuhn

References

Efron (1983). "Estimating the error rate of a prediction rule: improvement on cross-validation". Journal of the American Statistical Association, 78(382):316-331

Efron, B., & Tibshirani, R. J. (1994). "An introduction to the bootstrap", pages 249-252. CRC press.

Bergstra and Bengio (2012), "Random Search for Hyper-Parameter Optimization", Journal of Machine Learning Research, 13(Feb):281-305

Kuhn (2014), "Futility Analysis in the Cross-Validation of Machine Learning Models" <https://arxiv.org/abs/1405.6974>,

Package website for subsampling: <https://topepo.github.io/caret/subsampling-for-class-imbalances.html>