How to Do Supervised Learning in R?

Yewon Kim

Contents

What is Supervised Learning?

modeling a specific response variable as a function of some explanatory variables

- regression

- classification

Ex) Finding relationship between height and weight, predicting height based on weight : regression

Ex) Predicting if a person is a man or a woman based on height and weight : classification

What is Decision Tree?

Decision Tree is a tree-like flowchart that assigns class labels to individual observations

Daughter nodes are branched by a decision involving a single variable.

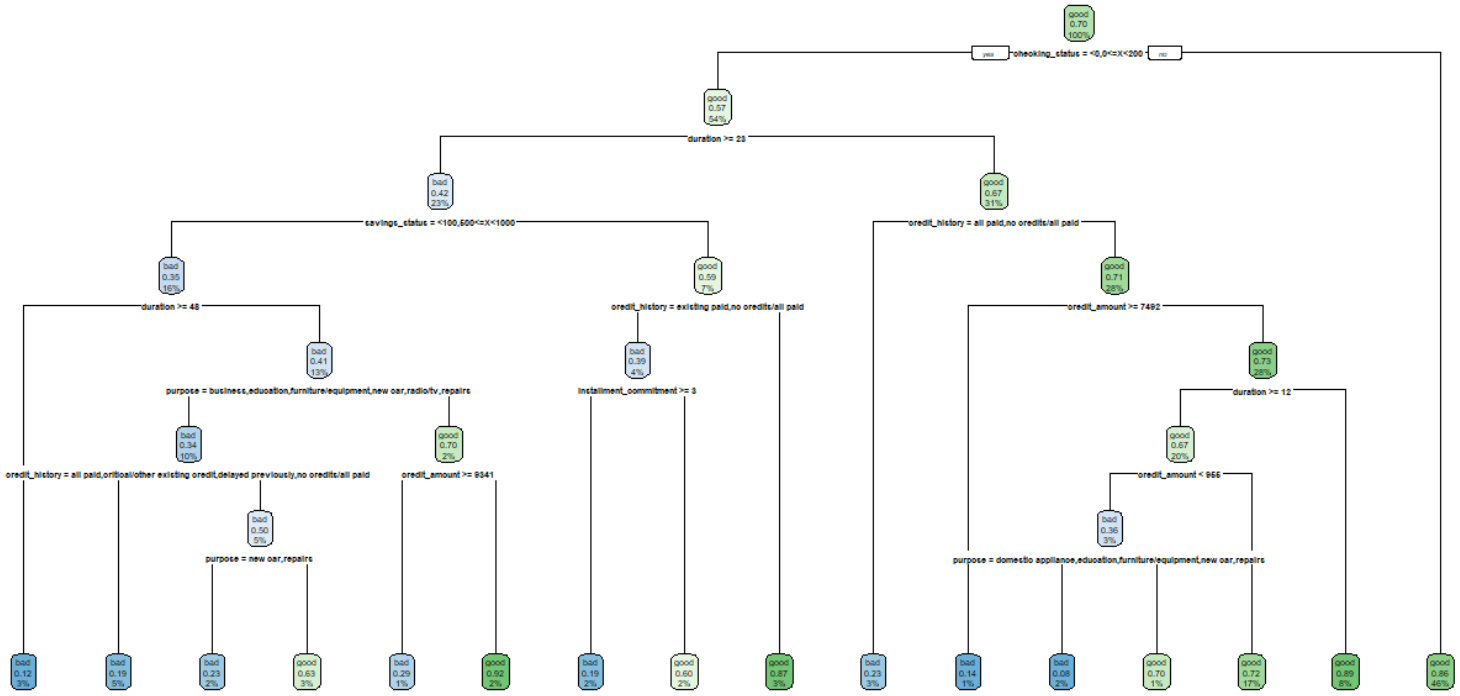
Each node shows:

the predicted class, the predicted probability, the percentage of observations in the node

Example of Decision Tree (Credit Risk Analysis)

variables used in this decision tree:

- checking_status
- duration
- saving_status
- credit_history
- etc



How to do Decision Tree in R?

We will use R package called ‘rpart’.

Also, R package called ‘rpart.plot’ will be used to draw a decision tree.

```
set.seed(123)

credit_tree<-rpart(form_full_credit,train_credit)
rpart.plot(credit_tree)
```

In this code,
- ‘set.seed’ is used to create random numbers that can be reproduced. You can put any number.
- By using rpart function, we will create decision tree.
  Usage : rpart(formula, data, weights, subset, na.action = na.rpart, method,
  model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)
- To draw decision tree : rpart.plot(decision tree)

What is Logistic Regression?

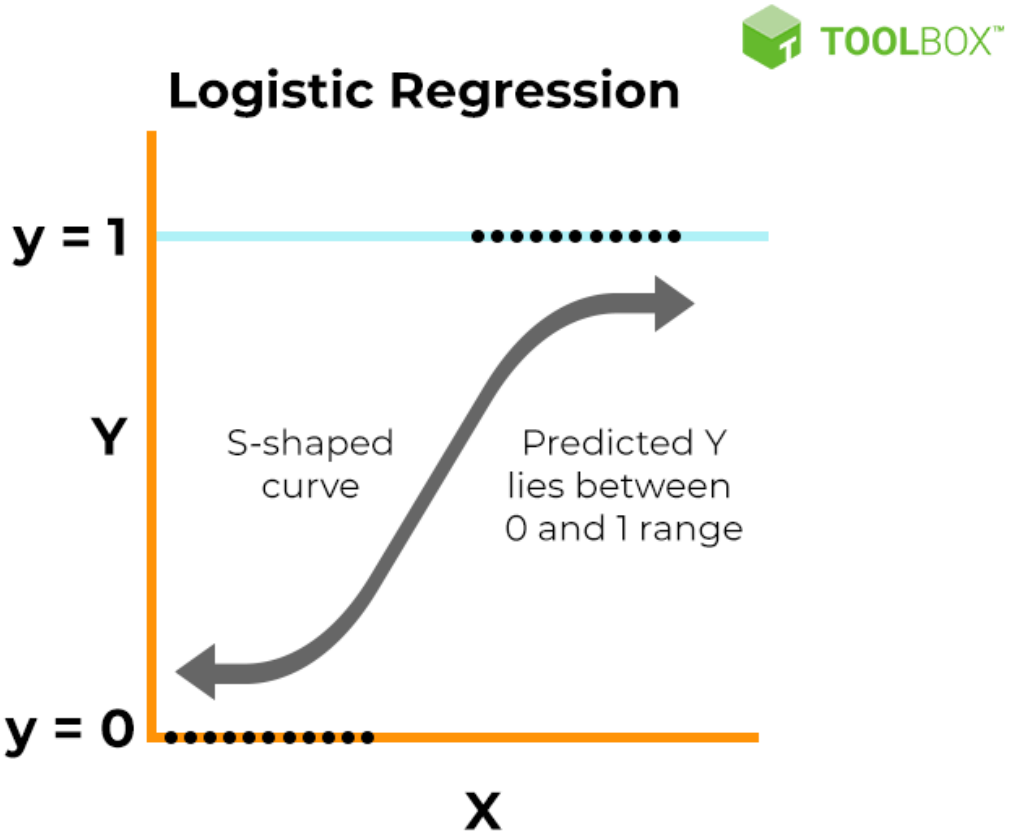
Logistic Regression is a model that estimates and predicts the probability of Y=1, given the values x.

In this case, Y has values of 0 or 1.

For prediction of Y, we will round $\hat{p}$ to either 0 or 1.

In particular, for $z = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$

$$p = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$



Example of Logistic Regression

```
Call:
glm(formula = train_credit$class ~ ., family = "binomial", data = train_credit)

Coefficients:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                              2.641e+00  1.416e+00   1.865  0.06212 .
duration                                -3.175e-02  1.078e-02  -2.947  0.00321 **
credit_amount                           -1.349e-04  4.961e-05  -2.719  0.00655 **
installment_commitment                  -3.211e-01  9.834e-02  -3.265  0.00109 **
residence_since                          3.748e-02  9.745e-02   0.385  0.70054
age                                      1.521e-02  1.064e-02   1.430  0.15276
existing_credits                        -2.795e-01  2.140e-01  -1.306  0.19152
num_dependents                          -2.530e-01  2.858e-01  -0.885  0.37601
checking_status>=200                     7.720e-01  3.938e-01   1.960  0.04996 *
checking_status0<=X<200                  4.477e-01  2.480e-01   1.805  0.07103 .
checking_statusno checking               1.712e+00  2.606e-01   6.569 5.08e-11 ***
credit_historycritical/other existing credit  1.641e+00  5.170e-01   3.174  0.00150 **
credit_historydelayed previously         8.363e-01  5.500e-01   1.521  0.12837
credit_historyexisting paid              9.083e-01  4.670e-01   1.945  0.05181 .
credit_historyno credits/all paid       -2.317e-01  6.461e-01  -0.359  0.71985
purposedomestic appliance               -1.056e+00  9.942e-01  -1.062  0.28835
purposeeducation                        -7.230e-01  5.329e-01  -1.357  0.17489
purposefurniture/equipment              -4.167e-01  4.075e-01  -1.023  0.30640
purposenew car                          -9.841e-01  3.819e-01  -2.577  0.00998 **
purposeother                             9.820e-01  1.058e+00   0.928  0.35324
purposeradio/tv                         -1.154e-01  3.795e-01  -0.304  0.76119
purposerepairs                          -1.326e+00  6.885e-01  -1.926  0.05407 .
purposeretraining                        1.448e+01  4.539e+02   0.032  0.97454
purposeused car                          4.960e-01  4.882e-01   1.016  0.30960
savings_status>=1000                     1.479e+00  5.768e-01   2.564  0.01034 *
savings_status100<=X<500                 3.322e-01  3.265e-01   1.017  0.30897
savings_status500<=X<1000                2.171e-01  4.368e-01   0.497  0.61914
savings_statusno known savings           9.465e-01  2.903e-01   3.260  0.00111 **
employment>=7                            3.375e-01  3.338e-01   1.011  0.31196
employment1<=X<4                         1.680e-01  2.750e-01   0.611  0.54131
employment4<=X<7                         7.732e-01  3.347e-01   2.310  0.02090 *
employmentunemployed                     2.203e-01  4.775e-01   0.461  0.64459
personal_statusmale div/sep             -7.019e-01  4.850e-01  -1.447  0.14783
personal_statusmale mar/wid             -8.283e-02  3.521e-01  -0.235  0.81403
personal_statusmale single               4.930e-01  2.357e-01   2.092  0.03644 *
other_partiesguarantor                   1.503e+00  6.669e-01   2.253  0.02426 *
other_partiesnone                        4.005e-01  4.503e-01   0.889  0.37379
property_magnitudelife insurance        -3.516e-01  2.623e-01  -1.341  0.18008
property_magnitudeno known property     -8.856e-01  4.601e-01  -1.925  0.05428 .
property_magnitudereal estate            8.814e-02  2.662e-01   0.331  0.74059
other_payment_plansnone                  6.875e-01  2.678e-01   2.567  0.01026 *
other_payment_plansstores                7.089e-01  5.133e-01   1.381  0.16724
housingown                              -3.761e-01  5.070e-01  -0.742  0.45816
housingrent                             -7.888e-01  5.379e-01  -1.467  0.14249
jobskilled                              -4.223e-01  3.304e-01  -1.278  0.20113
jobunemp/unskilled non res              -3.799e-01  7.885e-01  -0.482  0.62996
jobunskilled resident                   -4.750e-01  4.071e-01  -1.167  0.24323
own_telephoneyes                         2.091e-01  2.245e-01   0.931  0.35161
foreign_workeryes                       -1.860e+00  7.335e-01  -2.536  0.01121 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 973.97  on 799  degrees of freedom
Residual deviance: 706.00  on 751  degrees of freedom
AIC: 804
```

Significant variables(in terms of p-value):duration, credit_amount, installment_commitment, …

How to do Logistic Regression in R?

We will use R package called ‘glmnet’, and a function glm().

```
credit_glm <- glm(formula = train_credit$class ~.,data=train_credit, family='binomial')
summary(credit_glm)
```

Usage of glm():

glm(formula, family = gaussian, data, weights, subset,

na.action, start = NULL, etastart, mustart, offset,

control = list(…), model = TRUE, method = "glm.fit",

x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, …)

What is Random Forest?

A random forest is collection of bootstrapped decision trees.

A random forest is constructed by:

- Choosing the number of decision trees to grow and the number of variables to consider at each split

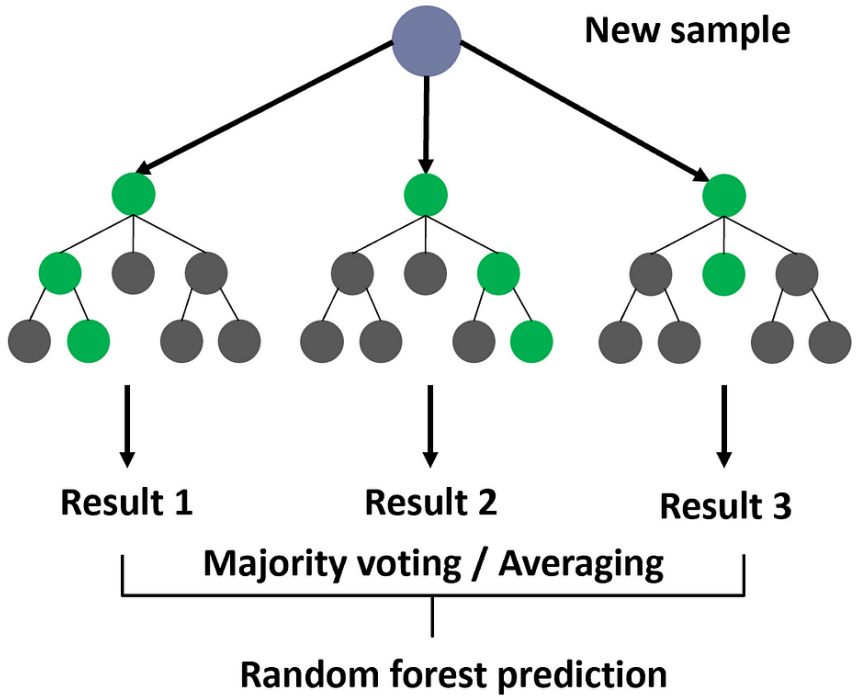
- Randomly selecting the rows of the data frame with replacement

- Randomly selecting variables from the data frame at each split

- Building a decision tree on the resulting data set

Example of Random Forest

credit_rf	list [19] (S3: randomForest.formu	List of length 19
call	language	randomForest(formula = class ~ ., data = train_cr
type	character [1]	'classification'
predicted	factor	Factor with 2 levels: "bad", "good"
err.rate	double [1000 x 3]	0.293 0.297 0.308 0.322 0.326 0.308 0.560 0.489 (
confusion	double [2 x 3]	89.0000 44.0000 149.0000 518.0000 0.6261 0.078
votes	double [800 x 2] (S3: matrix, arra	0.2332 0.1376 0.3204 0.5884 0.1387 0.0924 0.766
oob.times	double [800]	386 356 387 362 382 357 ...
classes	character [2]	'bad' 'good'
importance	double [20 x 4]	0.015168 0.011188 0.003018 0.001530 0.008382 ·
importanceSD	double [20 x 3]	0.001370 0.001326 0.000802 0.000776 0.001160 (
localImportance	NULL	Pairlist of length 0
proximity	NULL	Pairlist of length 0
ntree	double [1]	1000
mtry	double [1]	4
forest	list [14]	List of length 14
y	factor	Factor with 2 levels: "bad", "good"
test	NULL	Pairlist of length 0
inbag	NULL	Pairlist of length 0
terms	formula	class ~ duration + credit_amount + installment_c



Example of random forest in credit risk analysis

How to do Random Forest in R?

There is a package called ‘randomForest’.

Using this package, we can easily build a random forest model!

```
set.seed(123)
credit_rf<-randomForest(class~.,train_credit,ntree=1000,importance=TRUE)
varImpPlot(credit_rf, sort=TRUE, main="Variable Importance Plot")
```

varImpPlot is a function that create a plot that shows the importance of each variable, in terms of decreased accuracy and gini(measurement of purity).