Application of Decision Tree

Yingying Ji, 15220162202134

2019/05/03

Introduction

In this report, I will implement the methods used in the "Polish Attritioncy" case by another simulated data, employee attrition data. The methods include logistic regression, classification tree, cross-validation approach and boosting method. After analyzing the data by decision trees model, we could know which factors can be more important to the employee attrition and what factors the HR can change to prevent the loss of good people.

Data Description

This dataset contains various data points on an HR's employees. [1] And before I use it, I have already processed the original data with Excel by cleaning some unuseful data.

The next table shows the interpretation of some variables that are not clearly expressed only by their names.

Name of variables	Interpretation
Education	1: Below college; 2: College; 3: Bachelor; 4: Master; 5: Doctor
EnvironmentSatisfaction	1: Low; 2: Medium; 3: High; 4: Very High
JobInvolvement	1: Low; 2: Medium; 3: High; 4: Very High
JobSatisfaction	1: Low; 2: Medium; 3: High; 4: Very High
PerformanceRating	1: Low; 2: Good; 3: Excellent; 4: Outstanding
RelationshipSatisfaction	1: Low; 2: Medium; 3: High; 4: Very High
WorkLifeBalance	1: Bad; 2: Good; 3: Better; 4: Best
BusinessTravel	1: non-travel; 2: travel rarely; 3: travel frequently
Department	1: Human resources; 2: R&D; 3: Sales
Marital Status	1: Married; 2: Single; 3: Divorced

Codes and the Interpretation of Results[2]

Prepare the data

```
library(caret)
library(glmnet)
library(corrplot)
library(rpart)
library(rpart.plot)
library(randomForest)
library(nnet)
library(gbm)
library(AER)
rm(list=ls())
data <- read.csv("C:/Users/15068/Desktop/Microeconometrics/HW5/Employee.csv",header = T)
```

[1] The data are from ***Kaggle Dataset***.

[2] Since the code would have some problems running here, I put the whole codes in another file, where you can run to check the results.

See the attrition rate

```
data$Attrition = as.factor(data$Attrition)
dim(data)
sum(data$Attrition==1)/nrow(data) # Attrition rate in the data
set.seed(123)
train =  createDataPartition(data$Attrition,p=0.7,list=F)
data_train = data[train,]
data_test = data[-train,]
sum(data_train$Attrition==1)/nrow(data_train) # Attrition rate in the training data
sum(data_test$Attrition==1)/nrow(data_test) # Attrition rate in the test data
```

- Attrition rate in the data: 0.1612245
- Attrition rate in the training data: 0.161165
- Attrition rate in the test data: 0.161165

Logistic Regression

```
fit = glm(Attrition~.,data_train,family="binomial")
summary(fit)
result=coeftest(fit)
result[1:4,]
sum(result[,4]>0.05)
```

- The result is 16, which means 16 varibales are statistically significant.

Test error in logistic regression

```
ytrue = data_test$Attrition
phat = predict(fit,data_test,type="response")
yhat = as.numeric(phat > 0.5)
table(ytrue,yhat)
1-mean(yhat==ytrue) #misclassification error rate
```

- The result is 0.1159091, which means the misclassification error rate equals to 11.59%.

Claasification Tree

```
set.seed(100)
fit0 = rpart(Attrition~.,data_train,control=rpart.control(cp=0))
fit = prune(fit0,cp=fit0$cptable[which.min(fit0$cptable[,"xerror"]),"CP"])
rpart.plot(fit,box.palette=list("Grays","Reds"))
```

Test error in classification tree

```
yhat = predict(fit,data_test,type="class")
table(ytrue,yhat)
1-mean(yhat==ytrue)#misclassification error rate
```

- The result is 0.1545455, which means the misclassification error rate equals to 14.45%.
- We can see that the error rate increased by 2.86%. It seems that the prediction results by logistic regression can be preciser than by classification tree.

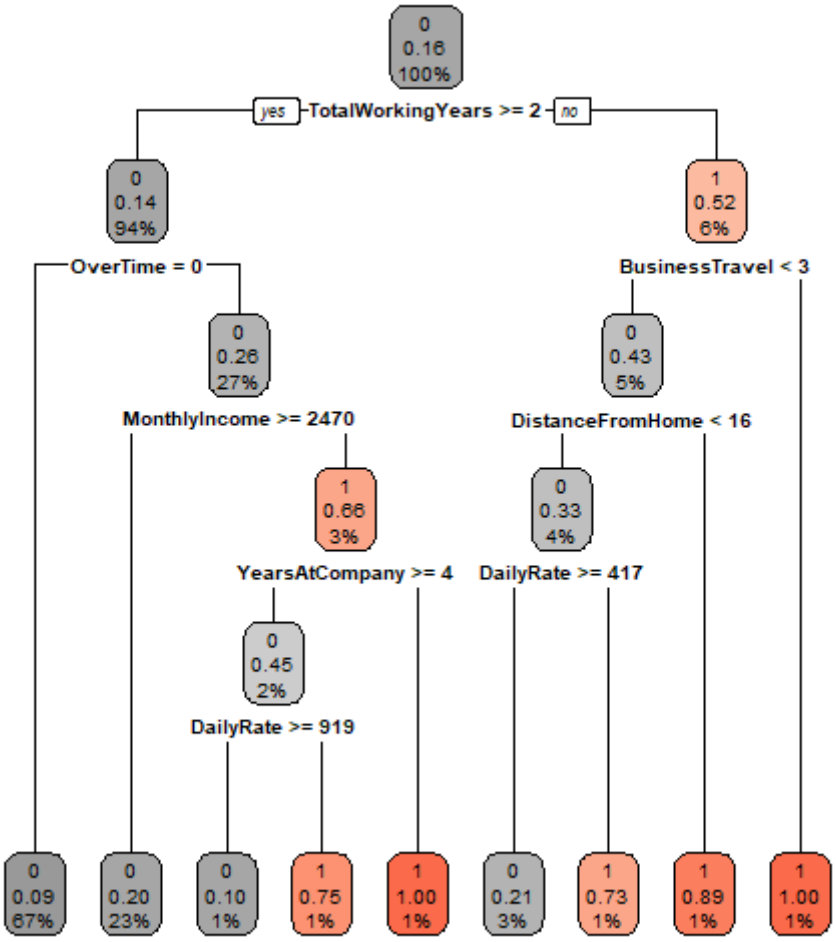

0
0.16
100%
yes
TotalWorkingYears >= 2
no
0
0.14
94%
1
0.52
6%
OverTime = 0
BusinessTravel < 3
0
0.26
27%
0
0.43
5%
MonthlyIncome >= 2470
DistanceFromHome < 16
1
0.66
3%
0
0.33
4%
YearsAtCompany >= 4
DailyRate >= 417
0
0.45
2%
DailyRate >= 919
0
0.09
67%
0
0.20
23%
0
0.10
1%
1
0.75
1%
1
1.00
1%
0
0.21
3%
1
0.73
1%
1
0.89
1%
1
1.00
1%

Figure 1:

Random Forest

```
set.seed(100)
fit=train(Attrition~.,data=data_train,method="rf",
          trControl=trainControl(method = "cv"),
          tuneLength=10) #tuneLength: number of mtry to try
fit$bestTune
fit = randomForest(Attrition~.,data=data_train,mtry=19)
```

- Here I use cross validation to select the best mtry. And the result is 19.

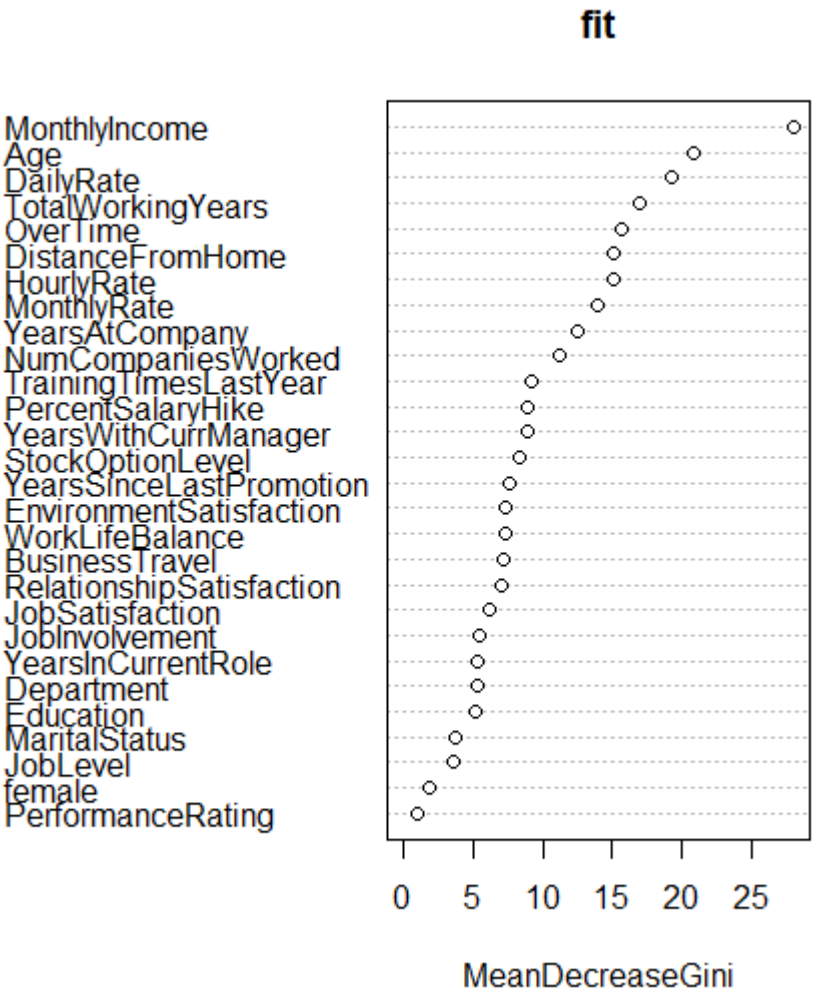
Test error in random forest

```
yhat = predict(fit,data_test)
table(ytrue,yhat)
1-mean(yhat==ytrue)
```

- The result is 0.1363636, which means the misclassification error rate equals to 13.64%.
- We can see that the error rate decreased by 0.81%.

The importance of variables

```
varImpPlot(fit)
```

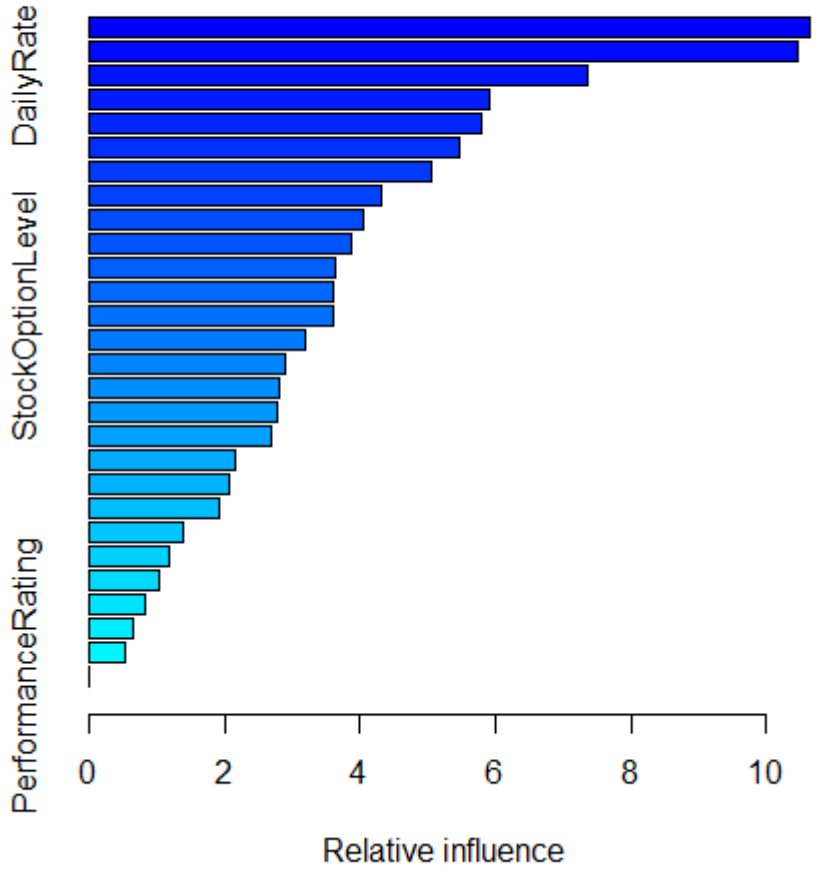


- From the graph, since the greater the "MeanDecreaseGini", the greater the importance of the variable, we can see that monthly income of workers is the most important factor in preventing workers from leaving the company. Age and daily rate of workers are the next.In the meanwhile, the performance rating of workers has no effects basically.

Boosting

```
set.seed(100)
data_boost = transform(data_train,Attrition=as.numeric(Attrition)-1)
ntree=5000
fit = gbm(Attrition~.,data_boost, distribution="adaboost",
          n.trees=ntree,interaction.depth=10,shrinkage=0.1)
summary(fit)
```

- The boosting method shows almost the same resluts as the ransom forest: monthly income and age of workers are the most important factors in preventing workers from leaving the company. Daily rate of workers are the next.In the meanwhile, the performance rating of workers has no effects basically.



Test error in boosting

```
phat=predict(fit,data_test,n.trees=5000,type = "response")
yhat=as.numeric(phat>0.5)
table(ytrue,yhat)
1-mean(yhat==ytrue)
```

- The result is 0.1363636, which means the misclassification error rate equals to 13.64%.