Model Selection

Yingying Ji, 15220162202134

2019/04/18

Introduction

In this report, I will implement the methods used in the "Supermarket Entry" case by another simulated data. The methods include logistic regression, lasso regression, cross-validation approach and KNN method.

Data Description

This dataset contains daily weather observations from numerous Australian weather stations. [1] And before I use it, I have already processed the original data with Excel by cleaning the unuseful data.

Series Name in Data Set	Interpretion
MinTemp	The minimum temperature in degrees celsius
MaxTemp	The maximum temperature in degrees celsius
Rainfall	The amount of rainfall recorded for the day in mm
Evaporation	The so-called Class A pan evaporation (mm) in the 24 hours to 9am
Sunshine	The number of hours of bright sunshine in the day
WindGustSpeed	The speed (km/h) of the strongest wind gust in the 24 hours to midnight
WindSpeed9am	Wind speed (km/hr) averaged over 10 minutes prior to 9am
WindSpeed3pm	Wind speed (km/hr) averaged over 10 minutes prior to 3pm
Humidity9am	Humidity (percent) at 9am
Humidity3pm	Humidity (percent) at 3pm
Pressure9am	Atmospheric pressure (hpa) reduced to mean sea level at 9am
Pressure3pm	Atmospheric pressure (hpa) reduced to mean sea level at 3pm
Cloud9am	Fraction of sky obscured by cloud at 9am. This is measured in "oktas"
Cloud3pm	Fraction of sky obscured by cloud at 3pm
Temp9am	Temperature (degrees C) at 9am
Temp3pm	Temperature (degrees C) at 3pm
RainTomorrow	1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0

Implementation in R language[2]

- Prepare the data

```
library(ISLR)
library(glmnet)
library(caret)
library(methods)
rm(list = ls())
set.seed(123)
```

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

[2] Since the code would have a problem running here, I put the whole codes and its results in another file.

```
weather <- read.csv("weather.csv")
weather$RainTomorrow <- factor(weather$RainTomorrow,levels=c(0,1),labels=c("FALSE","TRUE"))
summary(weather)
```

- Split data into training and test data sets

```
n <- nrow(weather)
train <- sample(n,n/2)
weather.tr <- weather[train,]
weather.te <- weather[-train,]
```

- Logistic regression on the full model

```
# Fit on training data
logitfit <- glm(RainTomorrow ~.,weather.tr,family='binomial')
summary(logitfit)
```

```
# Predict on test data
p = predict(logitfit,weather.te,type="response")
logitpred = as.factor(p > 0.5)
table(logitpred,weather.te$RainTomorrow,dnn=c("predicted","true"))
logiterr <- 1-mean(logitpred==weather.te$RainTomorrow) #misclassification error rate
logiterr
```

- Lasso regression (variable selection)

```
# Fit on training data
x <- model.matrix(RainTomorrow ~.,weather.tr)[,-1] #no intercept
y <- weather.tr$RainTomorrow
lassofit.all <- glmnet(x,y,alpha=1,family="binomial")
plot(lassofit.all,xvar="lambda")
```

- Cross validation (select a good value of λ)

```
cv.lasso <- cv.glmnet(x,y,alpha=1,family="binomial")
plot(cv.lasso)
```

```
# Refit the model using optimal lambda
lambda.star <- cv.lasso$lambda.min
lassofit.star <- glmnet(x,y,alpha=1,lambda=lambda.star,family="binomial")
coef(lassofit.star)
```

```
# Predict on test data
newx <- model.matrix(RainTomorrow ~.,weather.te)[,-1]
lassopred <- predict(lassofit.star,newx,type="class")
table(lassopred,weather.te$RainTomorrow,dnn=c("predicted","true"))
lassoerr <- 1-mean(lassopred==weather.te$RainTomorrow)
lassoerr
```

- KNN method (fit the model)

```
# Fit on training data
knnfit <- train(RainTomorrow ~.,data=weather.tr,method="knn", trControl=trainControl(method="repeatedcv
preProcess=c("center","scale"),tuneLength = 50) #center and scale predictors before KNN
plot(knnfit) #this step may take some time
```

```
# Predict on test data
knnpred <- predict(knnfit,weather.te)
table(knnpred,weather.te$RainTomorrow,dnn=c("predicted","true"))
```

```
knnerr <- 1-mean(knnpred==weather.te$RainTomorrow)
knnerr
```