Traning a Regression Model in R

Merve Topdemir

10 03 2021

CONTENTS

PACKAGES IMPORTING....1
TRANING A REGRESSION MODEL IN R....2
Splitting PhDPublications Data....2
Structure of PhDPublications Data....2
Creating a Linear Regression Model....3
Model Performance on Train Set....4
Predicting for train set....4
Calculating of MSE, RMSE and MAE on train set....4
Model Performance on Test Set....5
Predicting for test set....5
Calculating of MSE, RMSE and MAE on test set....5
CONCLUSİON....6

PACKAGES IMPORTING

```
#install.packages("AER") #to install the data package
library(AER)

data("PhDPublications")

#install.packages("dplyr") #For the glimpse function
library(dplyr)
```

TRANING A REGRESSION MODEL IN R

Splitting PhDPublications Data

The following code splits 80% of the data selected randomly into training set and the remaining 20% sample into test set.

Training set is implemented to build up a model, while a test set is to validate the model built.

```
set.seed(123)
index <- sample(nrow(PhDPublications), nrow(PhDPublications) * 0.8)
train <- PhDPublications[index,]
test <- PhDPublications[-index,]

nrow(train)

## [1] 732

nrow(test)

## [1] 183
```

Structure of PhDPublications Data

There are 915 observations and 6 variables : article, gender, married, child, prestige, mentor.

```
glimpse(PhDPublications)

## Rows: 915
## Columns: 6
## $ articles <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0~
## $ gender   <fct> male, female, female, male, female, female, female, mal
e, mal~
## $ married  <fct> yes, no, no, yes, no, yes, no, yes, no, yes, no, no, ye
s, yes~
## $ kids     <int> 0, 0, 0, 1, 0, 2, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0
, 0, 0~
## $ prestige <dbl> 2.520, 2.050, 3.750, 1.180, 3.750, 3.590, 3.190, 2.960,
4.620~
## $ mentor   <int> 7, 6, 6, 3, 26, 2, 3, 4, 6, 0, 14, 13, 3, 4, 0, 1, 7, 1
3, 7, ~
```

Creating a Linear Regression Model

Dependent (target) variable: Articles

Independent (feature) variables: Gender, Married, Kids, Prestige, Mentor

Model:

$$\mathrm{Articles} = \beta_0 + \beta_1\mathrm{Gender} + \beta_2\mathrm{Married} + \beta_3\mathrm{Kids} + \beta_4\mathrm{Prestige} + \beta_5\mathrm{Mentor}$$

Hypothesis test in linear regression:

$$H_0 = \beta_1 = \beta_2 = \cdots = \beta_k = 0$$

$$H_1 = \beta_j \neq 0 \text{ for at least one } j \neq 0$$

(F statistic is used to test H_0)

Hypothesis tests on individual regression coefficients:

$$H_0 = \beta_j = 0$$

$$H_1 = \beta_j \neq 0$$

(t statistic is used to test H_0)

P-value:

If the p-value is 0.05 or lower, the result is significant, but if it is higher than 0.05, the result is not significant.

P value is $1.752e^{-15}$ for significant of model. The model is significant because the p value is less than 0.05. The hypothesis is rejected because the model is significant.

According to the p values intercept, gender, married, children and mentor variables are significant, but the prestige variable is not significant.

R^2:

R-squared value is small and closer to zero. ($R^2 = 0.1023$)

A low R-squared value indicates that your independent variable is not explaining much in the variation of your dependent variable.

```
model <- lm(articles ~ factor(gender) + factor(married) + kids + prestige + 
mentor, data = train)
summary(model)

## 
## Call:
## lm(formula = articles ~ factor(gender) + factor(married) + kids + 
##     prestige + mentor, data = train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.8241 -1.2370 -0.3954  0.7405 14.9766 
## 
```

```
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.185994   0.271405   4.370 1.42e-05 ***
## factor(gender)female -0.292774   0.142614  -2.053  0.04044 *  
## factor(married)yes     0.297746   0.161199   1.847  0.06514 .  
## kids                  -0.285540   0.104722  -2.727  0.00655 ** 
## prestige               0.018469   0.072610   0.254  0.79929    
## mentor                 0.059649   0.007592   7.856 1.42e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 1.817 on 726 degrees of freedom
## Multiple R-squared:  0.1023, Adjusted R-squared:  0.09607
## F-statistic: 16.54 on 5 and 726 DF,  p-value: 1.752e-15
```

Model Performance on Train Set

Predicting for train set

```
predicted_train <- predict(model, train)
modelEvaluation <- data.frame(train$articles,predicted_train)
colnames(modelEvaluation) <- c('Actual','Predicted_train')
head(modelEvaluation, 10)
```

```
##       Actual   Predicted_train
## 415      1       0.9397615
## 463      1       1.2934630
## 179      0       1.0616298
## 526      2       1.2812197
## 195      0       1.7508940
## 818      4       1.5358844
## 118      0       0.9525216
## 299      1       1.4441727
## 229      0       1.5857992
## 244      0       1.6666874
```

Calculating of MSE, RMSE and MAE on train set

MSE: Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set.

RMSE: Root Mean Squared Error is the square root of Mean Squared Error.

MAE: The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset.

```
mse_train <- mean((modelEvaluation$Actual - modelEvaluation$Predicted_train
) ^ 2)
mse_train
```

```
## [1] 3.272876
```

```
rmse_train <- sqrt(mse_train)
rmse_train

## [1] 1.809109
```

```
mae_train <- mean(abs(modelEvaluation$Actual - modelEvaluation$Predicted_tr
ain))
mae_train

## [1] 1.305884
```

Model Performance on Test Set

Predicting for test set

```
predicted_test <- predict(model, test)
modelEvaluation2 <- data.frame(test$articles, predicted_test)
colnames(modelEvaluation2) <- c('Actual','Predicted_test')
head(modelEvaluation2, 10)

##      Actual   Predicted_test
## 1       0      1.9478260
## 3       0      1.3203734
## 7       0      1.1310832
## 12      0      1.6826042
## 22      0      0.8607355
## 27      0      1.4479995
## 28      0      1.4563788
## 32      0      2.0401649
## 35      0      1.2262727
## 43      0      1.5566377
```

Calculating of MSE, RMSE and MAE on test set

```
mse_test <- mean((modelEvaluation2$Actual - modelEvaluation2$Predicted_test
) ^ 2)
mse_test

## [1] 3.405654
```

```
rmse_test <- sqrt(mse_test)
rmse_test

## [1] 1.845441



mae_test <- mean(abs(modelEvaluation2$Actual - modelEvaluation2$Predicted_t
est))
mae_test

## [1] 1.316018
```

CONCLUSİON

```
df<- data.frame("MSE"=c(mse_train,mse_test), "RMSE"=c(rmse_train,rmse_test)
, "MAE"=c(mae_train,mae_test),row.names=c("Train","Test"))
df<-round(df,2)
df

##          MSE   RMSE   MAE

## Train   3.27   1.81   1.31

## Test    3.41   1.85   1.32
```

The results came out very close to each other. Overfitting may have occurred but it is not certain.