Prediction of Insurance Costs by Using Training Regression Model Steps

Sahranur İnce

3/20/23

Calling The Dataset

```
library(readr)
insurance <- read_csv("insurance.csv")
```

1. Detail your task with the problem, features, and target.

Problem: We are dealing with the problem of "prediction of insurance costs". We will look for an answer and prediction to our problem with the features in the dataset.

Features: There are 6 features in this dataset. These are:

age: age of primary beneficiary

sex: insurance contractor gender, female, male

bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9

children: Number of children covered by health insurance / Number of dependents

smoker: Smoking

region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

Target:

charges: Individual medical costs billed by health insurance

2. Describe the dataset in your task in terms of the dimension, variable type, and some other that you want to add.

```r
install.packages("DALEX")
install.packages("ggplot2")
library(DALEX)
library(ggplot2)
```

Here, the `str()` function is used to determine dimensions and what the types of features are.

```r
str(insurance)
```

```
spc_tbl_ [1,338 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ age     : num [1:1338] 19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : chr [1:1338] "female" "male" "male" "male" ...
 $ bmi     : num [1:1338] 27.9 33.8 33 22.7 28.9 ...
 $ children: num [1:1338] 0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : chr [1:1338] "yes" "no" "no" "no" ...
 $ region  : chr [1:1338] "southwest" "southeast" "southeast" "northwest" ...
 $ charges : num [1:1338] 16885 1726 4449 21984 3867 ...
 - attr(*, "spec")=
  .. cols(
  ..   age = col_double(),
  ..   sex = col_character(),
  ..   bmi = col_double(),
  ..   children = col_double(),
  ..   smoker = col_character(),
  ..   region = col_character(),
  ..   charges = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
```

According to the results obtained from the output;

The dimension of this dataset is $[1.338 \times 7]$ matrix. The features `age`, `bmi`, `children` and the target `charges` are numeric. The features `sex`, `smoker` and `region` are categorical.

Here, the sum(is.na()) function is used to check if there are missing instances in the dataset.

```
sum(is.na(insurance))
```

```
[1] 0
```

According to the results obtained, no missing instances were found in the dataset.

3. Train a linear regression model.

Splitting the data set

Here, the `sample()` function used to split the data set as `test` and `train` set. The `set.seed()` function is used for reproducibility.

```
set.seed(1234)
index <- sample(1 : nrow(insurance), round(nrow(insurance) * 0.80))
train <- insurance[index, ]
test  <- insurance[-index, ]
```

As a result, there are 1070 instance and 7 features in the train set. There are 268 instance and 7 features in the test set.

Here, the `dim()` function is used to show the row and column counts of the test and train sets.

```
dim(train)
```

```
[1] 1070    7
```

```
dim(test)
```

```
[1] 268   7
```

Train a linear regression model

Here, the `lm()` function is used to train a linear regression model. We used the `train` data and model formula that we split in the previous step. Categorical features in the dataset are specified as `factors` in the model.

```
lrm_model <- lm(charges ~ age + factor(sex) + bmi + children +
                  factor(smoker) + factor(region), data = train)
```

```
summary(lrm_model)
```

```
Call:
lm(formula = charges ~ age + factor(sex) + bmi + children + factor(smoker) + 
    factor(region), data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-11825.1  -2852.9   -899.8   1411.2  29998.3 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)             -11428.11    1104.04 -10.351  < 2e-16 ***
age                        256.04      13.48  18.992  < 2e-16 ***
factor(sex)male           -249.11     379.62  -0.656  0.51183    
bmi                        334.24      32.19  10.383  < 2e-16 ***
children                   428.93     157.91   2.716  0.00671 ** 
factor(smoker)yes        24161.99     468.14  51.613  < 2e-16 ***
factor(region)northwest   -327.04     543.64  -0.602  0.54758    
factor(region)southeast  -1559.52     539.85  -2.889  0.00395 ** 
factor(region)southwest  -1019.35     540.86  -1.885  0.05975 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6177 on 1061 degrees of freedom
Multiple R-squared:  0.7491,    Adjusted R-squared:  0.7472 
F-statistic: 395.9 on 8 and 1061 DF,  p-value: < 2.2e-16
```

In the output above; was obtained the minimum, maximum, median, 1st quartile and 3rd quartile values of the features. We see some statistics about the residuals, the model parameters, and also some performance metric values such as the multiple R squared and the adjusted R squared. These values show the model performance on train data. We see that the R squared value is 0,74. It means that the features explain the target by 74%. In other words, we can say that the model is meaningful at the rate of 74%. When we look at the p value found, we see that it is less than 0.05. This indicates that the model is statistically significant. Significance of the model means that at least one feature in the model is significantly to the target.

4. Report the performance of the trained model with only one metric that you learned in the lecture and share the reason why you chose the metric.

Here, `predicted_y()` function is used to predict values of the target variable on test set. We should exclude the true values of the target variable from the test set. Therefore, the target variable in the 7th column of the dataset has been exclude.

```r
predicted_y <- predict(lrm_model, test[,-7])
head(predicted_y)
```

```
        1         2         3         4         5         6 
 5841.858 35836.026 31919.852   819.403  2181.643 15968.602 
```

Here, we calculated the value of error terms for each predictions.

```r
error <- test$charges - predicted_y
head(error)
```

```
         1          2          3          4          5          6 
-1975.0026 -8027.3005  7691.9053  1017.8340   213.5288 -2739.7553 
```

Here, we calculated the main performance metrics mean squared error (MSE), root mean squared error (RMSE) and median absolute error (MAE) for the model used in the regression task.

```r
mse_model  <- mean(error ^ 2)
rmse_model <- sqrt(mean(error ^ 2))
mae_model  <- mean(abs(error))
```

Here; we can compare the values of the performance metrics to decide which one to use. Whichever have less level of error value is used.

```
mse_model
```

```
[1] 31627994
```

```
rmse_model
```

```
[1] 5623.877
```

```
mae_model
```

```
[1] 3986.693
```

5. Check any problem related to over and underfitting.

Here, the test and train sets are compared to determine whether the underfitting problem in the model exists.

```
rmse_train <- sqrt(mean((lrm_model$residuals) ^ 2))
rmse_test  <- rmse_model
```

```
rmse_train - rmse_test
```

```
[1] 527.4157
```

The difference is positive means that the performance of the model is better on train set than test set. Since the performance of train set is better than the performance of test set we can say that there is no overfitting problem.

6. Create a new observation (it is up to you, just create an observation with the feature values you want), and predict the value of its target feature.

A new model was created by subtracting `bmi` and `region` features from the dataset. In this new model, our target is again `charges`, our features are `age`, `sex`, `children` and `smoker`.

```
lrm_model2 <- lm(charges ~ age + factor(sex) + children
                 + factor(smoker) , data = train)
```

```
summary(lrm_model2)
```

```
Call:
lm(formula = charges ~ age + factor(sex) + children + factor(smoker),
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-15972.1  -1967.4  -1302.3   -491.7  28890.8

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2720.339    645.452  -4.215 2.71e-05 ***
age                 272.755     14.029  19.442  < 2e-16 ***
factor(sex)male       7.863    397.023   0.020   0.9842
children            415.351    165.344   2.512   0.0122 *
factor(smoker)yes 24140.541    489.440  49.323  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6475 on 1065 degrees of freedom
Multiple R-squared:  0.7232,    Adjusted R-squared:  0.7222
F-statistic: 695.7 on 4 and 1065 DF,  p-value: < 2.2e-16
```

```
predicted_y2 <- predict(lrm_model2, test[,-7])
head(predicted_y2)
```

```
        1         2         3         4         5         6
 6015.684 38331.012 28792.450  2885.220  3560.889 13644.962
```