HOMEWORK_1

ANASTASIA_SANTO

2023-03-28

The data 'breastfeed' come from a study conducted at a UK hospital, investigating the possible factors affecting the decision of pregnant women to breastfeed their babies.For the study, 135 expectant mothers were asked what kind of feeding method they would use for their coming baby. The responses were classified into two categories (variable breast in the dataset): the first 2 category (coded 1) includes the cases "breast-feeding", "try to breastfeed" and "mixed breast- and bottlefeeding", while the second category (coded 0) corresponds to "exclusive bottle-feeding". The possible factors, that are available in the data, are the advancement of the pregnancy (pregnancy), how the mothers were fed as babies (howfed), how the mother's friend fed their babies (howfedfr), if they have a partner (partner), their age (age), the age at which they left full-time education (educat), their ethnic group (ethnic) and if they have ever smoked (smokebf) or if they have stopped smoking (smokenow).

The first important thing I do is to look at the data, and to check the summary. I can see that I have I have 2 quantitative variables(educat and age) and 8 categhrical variables;in the quantitative variables I have some NA's. Since I don't have many of them I remove the rows in which I have the NA's.

```
library(tidyverse)
library(ISLR2)
load("./breastfeed.rdata")
# head(breastfeed) as_tibble(breastfeed)
df <- (breastfeed)
summary(df)
```

```
##     breast        pregnancy     howfed     howfedfr     partner    smokenow
##  Bottle: 39   End      :84   Bottle:59   Bottle:54   Single : 21   No :107
##  Breast:100   Beginning:55   Breast:80   Breast:85   Partner:118   Yes: 32
##
##
##
##
##
##  smokebf       age            educat           ethnic
##  No :88   Min.   :17.00   Min.   :14.00   White    :80
##  Yes:51   1st Qu.:25.00   1st Qu.:16.00   Non-white:59
##           Median :28.00   Median :17.00
##           Mean   :28.26   Mean   :18.15
##           3rd Qu.:32.00   3rd Qu.:19.00
##           Max.   :40.00   Max.   :38.00
##           NA's   :2       NA's   :2
```

```
dim(df)
```

```
## [1] 139  10
```

```
## [1] 135  10
```

Only four of the 139 women involved had been removed. Now I explore the relationship between quantitative variables(in our case age and educat) and I quantify the correlation between them by computing pairwise correlations:

```
# pairs(df)
cor(df[, c(8, 9)])
```

```
##              age    educat
## age    1.0000000 0.2001478
## educat 0.2001478 1.0000000
```

The Pearson correlation coefficient is a measure of any linear trend between two variables. As we can see from the matrix, there is a not a strongly positive linear trend.

At the beginning I Fit the following GLM model: $\text{logit}(E(\text{breast})) = 0 + 1\text{pregnancy} + 2\text{howfed} + 3\text{howfedfr} + 4\text{partner} + 5\text{age} + 6\text{educat} + 7\text{ethnic} + 8\text{smokenow} + 9\text{smokebf}$. The first thing to do is to divide the dataset into a train set(70%) and a test set(30%),paying attention to mantain a balance in the output variables.

```
library(caret)
attach(df)
set.seed(112)
trainIndex <- createDataPartition(df$breast, p = 0.7, list = FALSE, times = 1)
# head(trainIndex)
train_set <- df[trainIndex, ]
test_set <- df[-trainIndex, ]
Breast_train <- breast[trainIndex]
Breast_test <- breast[-trainIndex]
train_set$breast <- as.factor(train_set$breast)
test_set$breast <- as.factor(test_set$breast)
```

Now I fit the model with the 'train' function and check the summary.

```
glm.fits <- train(breast ~ howfed + pregnancy + howfedfr + partner + age + educat +
    ethnic + smokenow + smokebf, method = "glm", data = train_set, family = binomial)
summary(glm.fits)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4882  -0.2369   0.2146   0.4099   1.8954
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.53319    2.97590  -2.195  0.02814 *
## howfedBreast        0.63855    0.75105   0.850  0.39521
## pregnancyBeginning -1.94368    0.81137  -2.396  0.01659 *
```

```
## howfedfrBreast      1.53936    0.78045   1.972  0.04856 *
## partnerPartner      0.98045    0.83677   1.172  0.24131
## age                 0.09707    0.06796   1.428  0.15319
## educat              0.17512    0.15235   1.149  0.25037
## `ethnicNon-white`   2.28576    0.89899   2.543  0.01100 *
## smokenowYes        -3.75287    1.29178  -2.905  0.00367 **
## smokebfYes          2.22679    1.25123   1.780  0.07513 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 112.144  on 95  degrees of freedom
## Residual deviance:  64.612  on 86  degrees of freedom
## AIC: 84.612
##
## Number of Fisher Scoring iterations: 6
```

With an alpha-value of 0.5 to determine which predictors are significant in this model, I can say that 'howfedfr','pregnancy', 'smokenow' and 'ethnic' are statistically significant predictors. The coefficients estimate in the output indicate the average change in the log odds of the response variable associated with a one unit increase in each predictor variable,so a 'Partner' response in the partner factor is positively correlated to a 'breast' response;a 'Breast' response in the 'howfedfr' factor is positively correlated to a 'breast' response; a 'Beginning' response in the pregnancy factor is negatively correlated with a 'breast' response; a 'Yes' response in the 'smokenow' factor is negatively correlated to a 'breast' response; a 'non-white' response in the ethnic factor is positively correlated to a 'breast' response. Now I test the model on the test_set to check the accuracy.

```
glm.pred <- predict(glm.fits, newdata = test_set)
glm.pred[1:10]
```

```
##  [1] Breast Breast Breast Breast Breast Breast Breast Breast Bottle Breast
## Levels: Bottle Breast
```

```
table(glm.pred, Breast_test)
```

```
##         Breast_test
## glm.pred Bottle Breast
##   Bottle      8      4
##   Breast      2     25
```

```
# accuracy
accglm <- mean(glm.pred == Breast_test)
# error
log_err_rate <- mean(glm.pred != Breast_test)
```

The accuracy (the proportion of total correctly classified cases out of all the classification) is 85%, while the MSE is 15%.

Resampling methods are an indispensable tool that allow us to estimate the variability of a model,that would not be available from fitting the model only once using the original training sample. K-fold Cross-validation method can be used to estimate the test error associated with a given statistical learning method in order

to evaluate its performance. Therefore,to evaluate the performance of the difference methods that I'm going to create I will calculate the average error of this method using the cross validation. I use 8 as number of folds(k),since I don't have a big number of data.

```
ctrlspecs <- trainControl(method = "cv", number = 8, savePredictions = "all", classProbs = TRUE)
set.seed(15)
glm.fitk <- train(breast ~ howfed + pregnancy + howfedfr + partner + age + educat +
    ethnic + smokenow + smokebf, data = train_set, method = "glm", family = binomial,
    trControl = ctrlspecs)
print(glm.fitk)
```

```
## Generalized Linear Model
##
## 96 samples
##  9 predictor
##  2 classes: 'Bottle', 'Breast'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 83, 84, 84, 83, 85, 84, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7585956  0.3878696
```

The mean accuracy is 76%. The Kappa is the baseline probability of our output variable, and it's 58%.

```
summary(glm.fitk)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4882  -0.2369   0.2146   0.4099   1.8954
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.53319    2.97590  -2.195  0.02814 *
## howfedBreast        0.63855    0.75105   0.850  0.39521
## pregnancyBeginning -1.94368    0.81137  -2.396  0.01659 *
## howfedfrBreast      1.53936    0.78045   1.972  0.04856 *
## partnerPartner      0.98045    0.83677   1.172  0.24131
## age                 0.09707    0.06796   1.428  0.15319
## educat              0.17512    0.15235   1.149  0.25037
## `ethnicNon-white`   2.28576    0.89899   2.543  0.01100 *
## smokenowYes        -3.75287    1.29178  -2.905  0.00367 **
## smokebfYes          2.22679    1.25123   1.780  0.07513 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 112.144  on 95  degrees of freedom
## Residual deviance:  64.612  on 86  degrees of freedom
## AIC: 84.612
##
## Number of Fisher Scoring iterations: 6
```

Looking at the summary I can assess that the significant variables are the same as the first logistic model glm.fits. Now I estimate this model on my test_set.

```
pred_gl <- predict(glm.fitk, newdata = test_set)
table(pred_gl, Breast_test)
```

```
##         Breast_test
## pred_gl  Bottle Breast
##   Bottle      8      4
##   Breast      2     25
```

```
# accuracy
accglk <- mean(pred_gl == Breast_test)
# error
errglk <- mean(pred_gl != Breast_test)
```

In the test_set the accuracy is 85% while the MSE is 15%. Nothing changed from the first model.

I try to fit the model using only the significant variables I found in the models above.

In this model the accuracy is 85%, same as the model with all the variables.

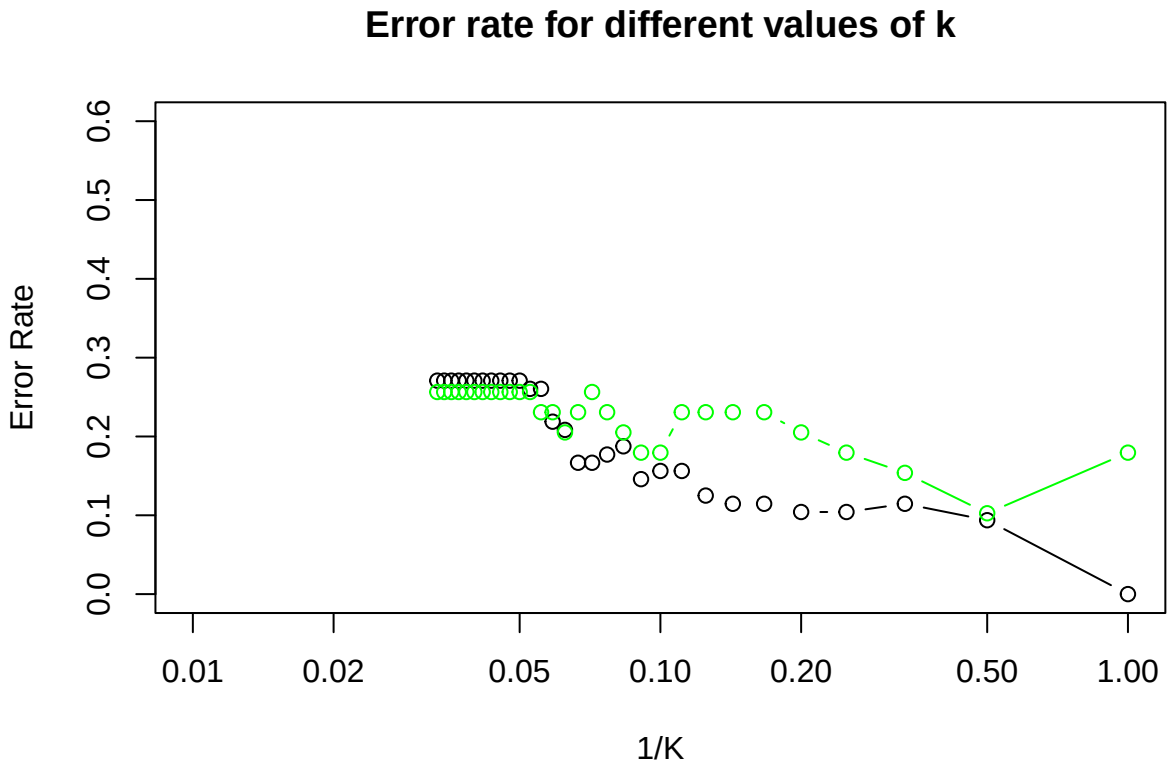
Now I Fit a K-Nearest Neighbors classifier, by performing a selection of the tuning parameter.As KNN is based on distances to identify observations near to each other, variables on a large scale will impact the distance more than variables on a smaller scale.In this case most of the variables are categorical, so I don't need to standardize the data, but I need to convert the categorical values into numbers. With knn() function I consider 'Euclidean' distance.

```
library(class)
df2 <- df
df2 <- lapply(df2, as.numeric)
df2 <- data.frame(df2)

set.seed(112)
trainIndexk2 <- createDataPartition(df$breast, p = 0.7, list = FALSE, times = 1)
ktrain2 <- df2[trainIndexk2, ]
ktest2 <- df2[-trainIndexk2, ]
breast_traink2 <- breast[trainIndexk2]
breast_testk2 <- breast[-trainIndexk2]
calc_error_rate <- function(predicted.value, true.value) {
    mean(true.value != predicted.value)
}
errors_tr2 <- errors_ts2 <- c()

kvec <- c(seq(1:30))
for (k in kvec) {
    pred_tr2 <- knn(ktrain2, ktrain2, breast_traink2, k = k)
```

```
    pred_ts2 <- knn(ktrain2, ktest2, breast_traink2, k = k)
    err_tr2 <- calc_error_rate(pred_tr2, breast_traink2)
    err_ts2 <- calc_error_rate(pred_ts2, breast_testk2)
    errors_tr2 <- append(errors_tr2, err_tr2)
    errors_ts2 <- append(errors_ts2, err_ts2)
}
plot(1, type = "n", xlim = c(0.01, 1), ylim = c(0, 0.6), log = "x", xlab = "1/K",
    ylab = "Error Rate", main = "Error rate for different values of k", )
lines(1/kvec, errors_tr2, type = "b")
lines(1/kvec, errors_ts2, type = "b", col = "green")
```




```
## [1] 2
```

Iterating through values of k that goes from 1 to 30,the value of K that minimizes the MSE is K=2. Now I do the prediction on my test_set using k=2.

The accuracy for k=2 is 92%, while the MSE is 8%.

Naive Bayes algorithm is a classification technique based on Bayes' Theorem with an independence assumption among predictors. Now I Fit a Naïve Bayes classifier.

```
library("e1071")
library("dplyr")
library("gmodels")
set.seed(115)
```

```
trainIndex <- createDataPartition(df$breast, p = 0.7, list = FALSE, times = 1)

nfit <- naiveBayes(breast ~ howfed + pregnancy + howfedfr + partner + age + educat +
    ethnic + smokenow + smokebf, data = train_set)
npred <- predict(nfit, test_set, type = "class")
# posterior probability
post <- predict(nfit, test_set, type = "raw")
table(npred, Breast_test, dnn = c("Prediction", "Actual"))
```

```
##           Actual
## Prediction Bottle Breast
##     Bottle      8      6
##     Breast      2     23
```

```
accn <- mean(npred == Breast_test)
log_err_raten <- mean(npred != Breast_test)
# summary(nfit)
```

The accuracy of this model is 80%.

Given the result above, we can say that the model that better predict the output variable is the knn, with k=2, because of the accuracy level. Obviously we have to remember that the accuracy of the knn model is influenced by the random partition of the data_set, so we can see some fluctuations on the accuracy and to the MSE values among different choices on the partition of the dataset.

‘‘‘