Statistical Learning, Homework #1

Annalisa Xamin

Introduction

Understanding the factors that mostly influence pregnant women's decisions to breastfeed their children is the main goal of this analysis. To do that, different prediction models will be compared.

The dataset

The data come from a study conducted at a UK hospital. For the study, 139 expectant mothers were asked what kind of feeding method they would use for their coming baby.

```
load("breastfeed.Rdata")
dataf <- breastfeed
summary(dataf)
```

```
##     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
```

The response variable `breast` is categorical. The responses are classified into two categories: the first category (coded 1) includes the cases "breastfeeding", "try to breastfeed" and "mixed breast- and bottle-feeding", while the second category (coded 0) corresponds to "exclusive bottle-feeding".

We can visualize the two categories with a barplot.

```
ggplot(data = dataf, aes(x = breast, fill = breast)) + geom_bar(alpha = 0.7) + theme(axis.title.x = ele
    axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

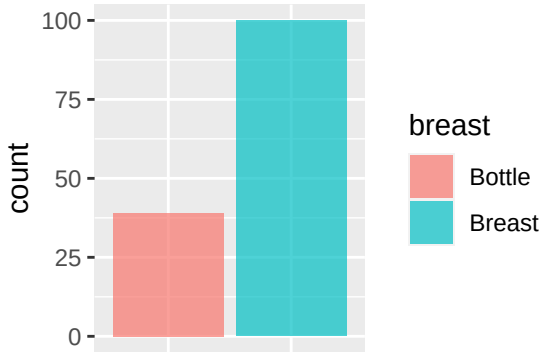

Figure 1: Categories from the breast variable.

```
# length(which(breastfeed$breast == 'Breast')) #100
# length(which(breastfeed$breast == 'Bottle')) #39
```

A clear imbalance can be observed in the dataset: there are many more women that prefer breastfeeding (100) over bottlefeeding (39).

The factors that could influence the decision of breastfeeding 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**).

Pre-processing

A potential issues could be the presence of NAs. As we can see from the summary of the data, there are 2 NAs in `age` and 2 NAs in `educat`.

We check to which category of `breast` these NAs belongs to.

```
dataf %>%
    select(breast, age, educat) %>%
    filter(is.na(age) | is.na(educat))
```

```
##     breast age educat
## 6   Bottle  NA     28
## 22  Bottle  31     NA
## 46  Bottle  38     NA
## 125 Breast  NA     16
```

We can notice that 3 of them belong to the category `Bottle`. Since we have just few observations in the `Bottle` category, we decided to not remove the NAs, but we could substitute them with the mean or the median of the column instead. In order to take a decision, we check the distribution of our variable of interest.

```
grid.arrange(ggplot(data = dataf, aes(x = educat)) + geom_histogram(binwidth = 3,
    colour = 1, size = 0.1, na.rm = TRUE), ggplot(data = dataf, aes(x = age)) + geom_histogram(binwidth
    colour = 1, size = 0.1, na.rm = TRUE), ncol = 2)
```

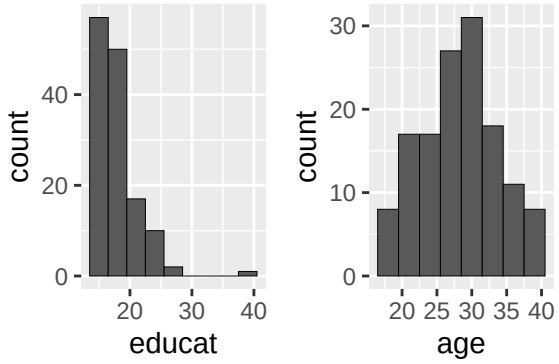

Figure 2: Distribution of the variables educat and age.

By looking at the plots, we can see that the distribution of the variable `age` doesn't seem skewed, so we could replace the NAs with the mean. While, the distribution of the variable `educat` is more skewed and we will use the median to replace the NAs, since the mean is sensible to outliers.

```
dataf <- dataf %>%
    mutate(educat = ifelse(is.na(educat), median(educat, na.rm = TRUE), educat),
        age = ifelse(is.na(age), mean(age, na.rm = TRUE), age))
any(is.na(dataf))
```

```
## [1] FALSE
```

As we can see the substitution of NAs has been successful.

Splitting into training and test sets

For reproducibility, we set the seed to 1.

```
set.seed(1)
```

Now, I split the data into training and testing sets. Given the class imbalance we saw before, I used the function `caret::createDataPartition` to have sets that have the same imbalance with respect to the outcome variable. This function takes as parameter the percentage of the training data to generate automatically a random list of indexes of the data that will be used in the training set. In this case, I decided that the training set will contain 80% of the samples.

```
part <- caret::createDataPartition(dataf$breast, p = 0.8)
train_df <- dataf[part$Resample1, ]
test_df <- dataf[-part$Resample1, ]
control <- trainControl(method = "cv")
```

A cross validation control was set for future computations.

Prediction models

Generalized Linear Model

I fit the following GLM model:

$$\begin{aligned}\text{logit}(\text{E}(\text{breast})) = \beta_0 &+ \beta_1 \text{pregnancy} + \beta_2 \text{howfed} + \beta_3 \text{howfedfr} \\ &+ \beta_4 \text{partner} + \beta_5 \text{age} + \beta_6 \text{educat} + \beta_7 \text{ethnic} + \beta_8 \text{smokenow} + \beta_9 \text{smokebf}\end{aligned}$$

```
glm.fit <- train(breast ~ ., data = train_df, method = "glm", trControl = control)
summary(glm.fit)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0378  -0.5937   0.2877   0.5239   2.6194
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.09899    2.23642  -1.386   0.1658
## pregnancyBeginning -1.00614    0.59142  -1.701   0.0889 .
## howfedBreast       -0.45483    0.64588  -0.704   0.4813
## howfedfrBreast      1.91894    0.61805   3.105   0.0019 **
## partnerPartner      1.01147    0.75844   1.334   0.1823
## smokenowYes        -2.57514    1.02505  -2.512   0.0120 *
## smokebfYes          1.28735    1.02446   1.257   0.2089
## age                 0.01930    0.05352   0.361   0.7183
## educat              0.10177    0.10002   1.017   0.3089
## `ethnicNon-white`   1.86669    0.73463   2.541   0.0111 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134.012  on 111  degrees of freedom
## Residual deviance:  87.655  on 102  degrees of freedom
## AIC: 107.65
##
## Number of Fisher Scoring iterations: 5
```

By looking at the coefficient, we can observe that `howfedfr` for the value `Breast` appears to be the most important one. This means that how the mother's friend fed their babies seems to have influence on the mother's own method of feeding.

We notice that other two coefficients, `smokenow` for the value `Yes` and `ethnic` for the value `Non-white`, also appear to be significant, even if less. In other terms, if the mother hasn't quit smoking, the log odds of breastfeeding decreases by -2.57 (i.e. the odds of breastfeeding are multiplied by $e^{-2.57}$) if every other predictors are kept constant.

Then, we compute the predictions for the test data and visualize the confusion matrix.

```r
glm.probs <- predict(glm.fit, test_df, type = "prob")

glm.pred <- rep("Bottle", nrow(test_df))
glm.pred[glm.probs[, 2] > 0.5] <- "Breast"

table(glm.pred, test_df$breast)
```

```
##
## glm.pred Bottle Breast
##   Bottle      5      4
##   Breast      2     16
```

```r
# Accuracy
glm.acc <- mean(glm.pred == test_df$breast)  # ~77.8%
```

The accuracy of the generalized linear model is ~77.8%.

K-nn claissifier

Now, we try to fit a k-nn classifier. Firstly, we divide the training data in multiple partitions to test.

```r
train_part <- train_df
folds <- list()
n_folds <- 5
len_sample <- round((nrow(train_part)/n_folds))  # round to the nearest lower integer
for (n in c(1:n_folds)) {
    set.seed(1)
    sample <- sample.int(nrow(train_part), len_sample, replace = FALSE)
    folds[paste0("fold", n)] <- list(train_part[sample, ])
    train_part <- train_part[-sample, ]
}
```

Then, a model is trained for each fold with k that ranges between 1 and 20.

```r
foldFits <- list()
i <- 1
k_max <- 20  # set the maximum k
# train a model on each fold for all the possible values of k
for (fold in folds) {
    tmp <- train(breast ~ ., data = fold, method = "knn", trControl = trainControl(method = "cv",
        number = n_folds), tuneGrid = expand.grid(k = 1:k_max))
    foldFits[paste0("fold", i)] <- list(tmp)
    i <- i + 1
}
```

Find out the mean accuracy of each model to later choose the best k.

```r
accuracies <- list()
i <- 1

# save the accuracies
```

```
for (fold in foldFits) {
    accuracies[[i]] <- fold$results$Accuracy
    i <- i + 1
}

means <- c()
tmp <- 0

# calculate the mean for the accuracies
for (i in 1:k_max) {
    for (acc in accuracies) {
        tmp <- tmp + acc[i]
    }
    tmp <- tmp/n_folds
    means <- c(means, tmp)
    tmp <- 0
}

ks <- tibble(k = c(1:k_max), means = means)
ggplot(data = ks, aes(x = k, y = means)) + geom_point()
```

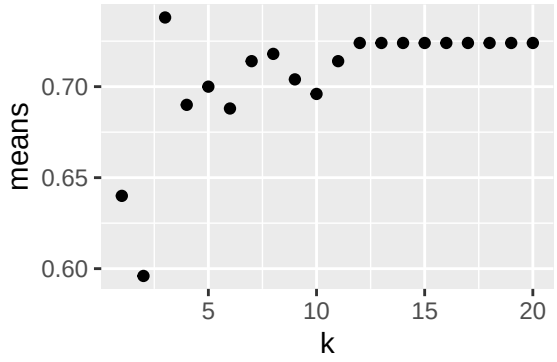

Figure 3: This plot shows how the average accuracy changes with respect to k.

The confusion matrix for the prediction is then visualized.

```
knn.spec <- nearest_neighbor(neighbors = which.max(means)) %>%
    set_mode("classification") %>%
    set_engine("kknn")

knn.fit <- knn.spec %>%
    fit(breast ~ ., data = train_df)

augmented <- augment(knn.fit, new_data = test_df)

augmented %>%
    conf_mat(truth = breast, estimate = .pred_class)
```

```
##           Truth
## Prediction Bottle Breast
```

```
##     Bottle      4      4
##     Breast      3     16
```

```r
# Accuracy
knn.acc <- augmented %>%
    accuracy(truth = breast, estimate = .pred_class)  # ~74.07%
```

Naive Bayes classifier

The last fit of the analysis is done on a Naïve Bayes classifier.

```r
naive.fit <- train(breast ~ ., data = train_df, method = "naive_bayes", trControl = control)

naive.pred <- predict(naive.fit, test_df, type = "prob")
pred.final <- ifelse(naive.pred[, 2] >= 0.5, "Breast", "Bottle")
table(pred.final, test_df$breast)
```

```
##
## pred.final Bottle Breast
##     Bottle      2      0
##     Breast      5     20
```

```r
# Accuracy
naive.acc <- mean(pred.final == test_df$breast)  # 81.48%
```

Comparison between the different methods

Evaluate the performance of the methods and compare the results.

```r
# Compute specificity
glm.spec <- specificity(table(glm.pred, test_df$breast))
knn.spec <- specificity(table(augmented$.pred_class, test_df$breast))
naive.spec <- specificity(table(pred.final, test_df$breast))

# Compute sensitivity
glm.sens <- sensitivity(table(glm.pred, test_df$breast))
knn.sens <- sensitivity(table(augmented$.pred_class, test_df$breast))
naive.sens <- sensitivity(table(pred.final, test_df$breast))

# Compute ROC
glm.roc = roc(test_df$breast ~ glm.probs[, 2])
knn.roc = roc(test_df$breast ~ augmented$.pred_Breast)
bay.roc = roc(test_df$breast ~ naive.pred[, 2])

# Compare all the statistics between the models
df_comparison <- data.frame(GLM = c(glm.acc, glm.spec, glm.sens, glm.roc$auc), KNN = c(knn.acc$.estimat
    knn.spec, knn.sens, knn.roc$auc), NB = c(naive.acc, naive.spec, naive.sens, bay.roc$auc))
rownames(df_comparison) <- c("Accuracy", "Specificity", "Sensitivity", "AUC")
# df_comparison
```

Conclusions

On the basis of the completed analysis, the following statements may be made:

- K-NN doesn't perform the training on the train split, but requires a choice of a parameter k that depends on the error rate on the test split. This means that the model already saw the test data in the model assessment phase. So, to make a fair comparison with the generalized linear model and the Naive Bayes, it was performed a nested cross validation to select the optimal k.
- As we can see from the following table, although the Naive Bayes (NB) has better overall accuracy, it has a very low sensitivity, this means that if the test observation has `breast = Bottle` the model has $\approx 28\%$ probability to predict the observation correctly, and since the response variable is unbalanced, the sensitivity measure is more noteworthy than the others.

```
df_comparison
```

```
##                    GLM       KNN        NB
## Accuracy    0.7777778 0.7407407 0.8148148
## Specificity 0.8000000 0.8000000 1.0000000
## Sensitivity 0.7142857 0.5714286 0.2857143
## AUC         0.9142857 0.8000000 0.8571429
```

- The same inference can be done with the K-NN model. In this case, however, the trade-off between specificity and sensitivity is less pronounced.
- The best model may be the generalized linear model: it has high specificity and sensitivity.

Depending on our goal, we may prefer a model to another: if we want to correctly classify the observation with `breast = Breast`, then the Naive Bayes is preferred. However, if we want to predict correctly observations in general, then the generalized linear model is the best choice as it has both high specificity and sensitivity.

Additionally, given the class imbalance we should pay more attention to the sensitivity performance when choosing the best model.