Report

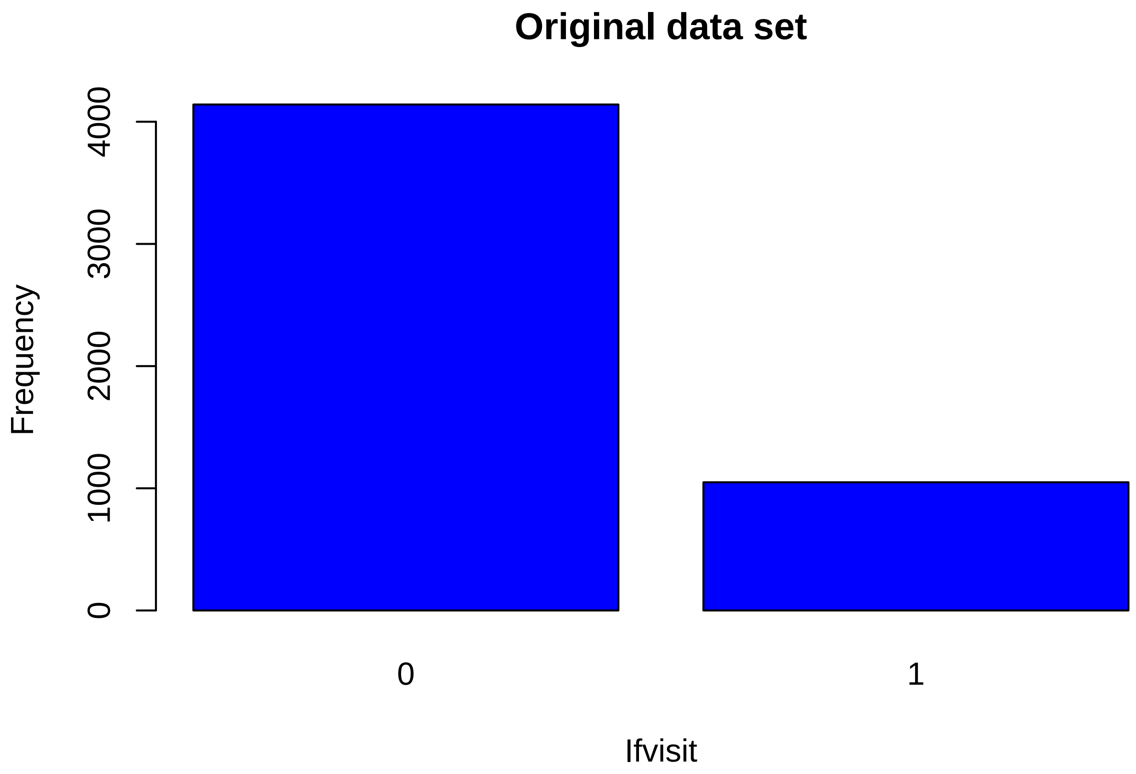
kiki

2024-02-29

Binary classification problem

We start by analyzing some models where the response variable “doctorco” is transformed into the binary response variable “ifvisit”. This binary variable is equal to 0 if “doctorco” is 0, and is equal to 1 otherwise. By doing so, we are modelling the number of people that went to a doctor’s visit at least once in the last two weeks.

To start we import the data set and create the “ifvisit” variable. From the following graph we can see the skewness of the data set regarding this variable:



We can try fitting a GAM using the obtained data set:

```
model_gam <- gam(ifvisit ~ s(hospdays) + s(actdays) + age*prescrib + freepoor + hscore + nonpresc + illn
summary(model_gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## ifvisit ~ s(hospdays) + s(actdays) + age * prescrib + freepoor +
##     hscore + nonpresc + illness
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.72414    0.13763 -19.794  < 2e-16 ***
## age           1.52680    0.27834   5.485 4.13e-08 ***
## prescrib      0.91275    0.10486   8.704  < 2e-16 ***
## freepoor     -0.91922    0.30227  -3.041  0.00236 **
## hscore        0.06140    0.02004   3.064  0.00219 **
## nonpresc     -0.18891    0.06426  -2.940  0.00328 **
## illness       0.16493    0.03493   4.722 2.34e-06 ***
## age:prescrib -1.01709    0.17178  -5.921 3.20e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##               edf Ref.df Chi.sq p-value
## s(hospdays) 4.286  4.955  17.17 0.00514 **
## s(actdays)  3.259  3.931 172.31 < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.205   Deviance explained = 18.8%
## UBRE = -0.16331  Scale est. = 1         n = 4152
```

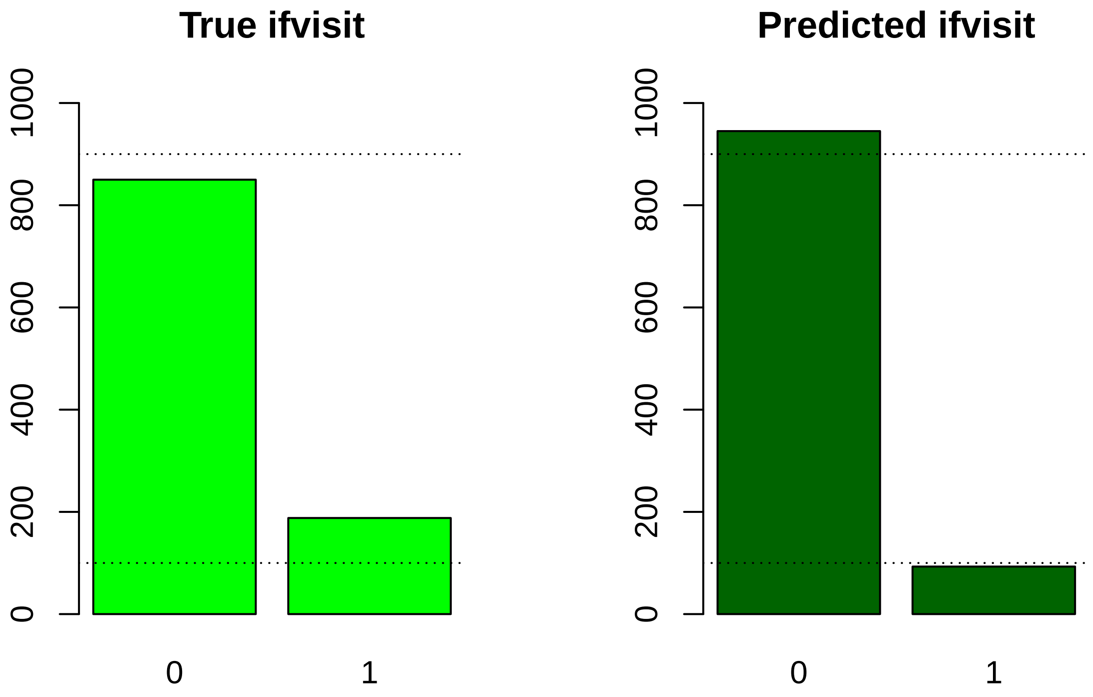
Both the variables “hospdays” and “actdays” were considered as splines after a top to bottom examination of the covariates. The summary shows that we can account for approx. 19% of explained deviance at best, which is not a great result so far.

```
## [1] "AIC (GAM): 3473.93"
```

```
## [1] "MAE (GAM): 0.1686"
```

```
## [1] "Number of true ifvisit: 188"
```

```
## [1] "Number of ifvisit predicted: 93"
```

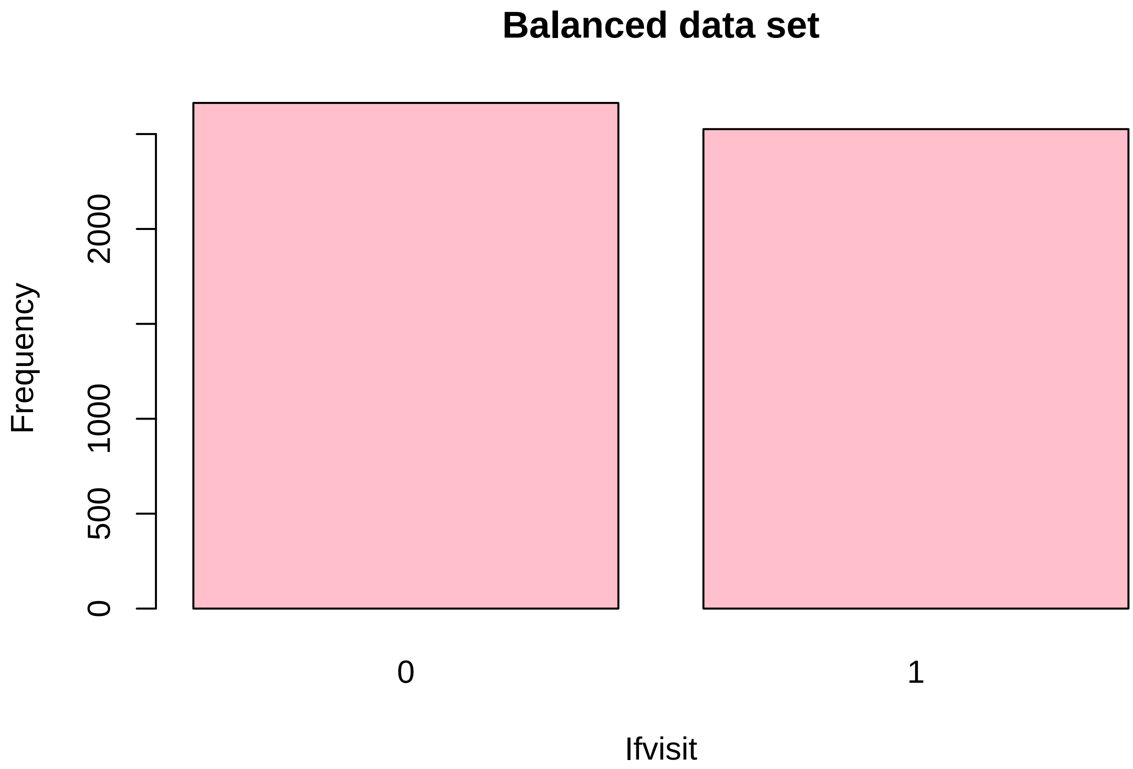


The sum of predicted “ifvisit” yealds a total of 93, against the true value of 188, indicating that GAM isn’t performing very well at predicting the minority class ‘1’. To enhance it’s abilities, we should further manipulate the data set as shown in the following section.

Balancing the data set with ROSE

Since we transformed the problem into a binary classification problem we can use techniques to balance the data set. One option is to use ROSE (Random Over-Sampling Examples), a method used for oversampling the minority class in binary classification problems to balance the data set. It involves generating synthetic examples from the existing minority class instances. This can be achieved by randomly selecting a minority class instance and introducing variations to create new synthetic instances.

```r
data.rose <- ROSE(ifvisit ~ ., data = data, seed = 1, hmult.majo = 0)$data
```



The ROSE method generated some data that were not suitable for the data set, for ex. it generated negative data for variables that can only obtain positive integer values. We solved this problem by taking the absolute value of the data set after the data augmentation was done.

Here ROSE was first used to generate more data, then was considered the absolute value of the dataset since the variables can't obtain negative values, and then rounded accordingly so that every observation has integers as values when the variable is a count.

GAM with ROSE

We can try again fitting a GAM on this new balanced data set and see if the performance improved:

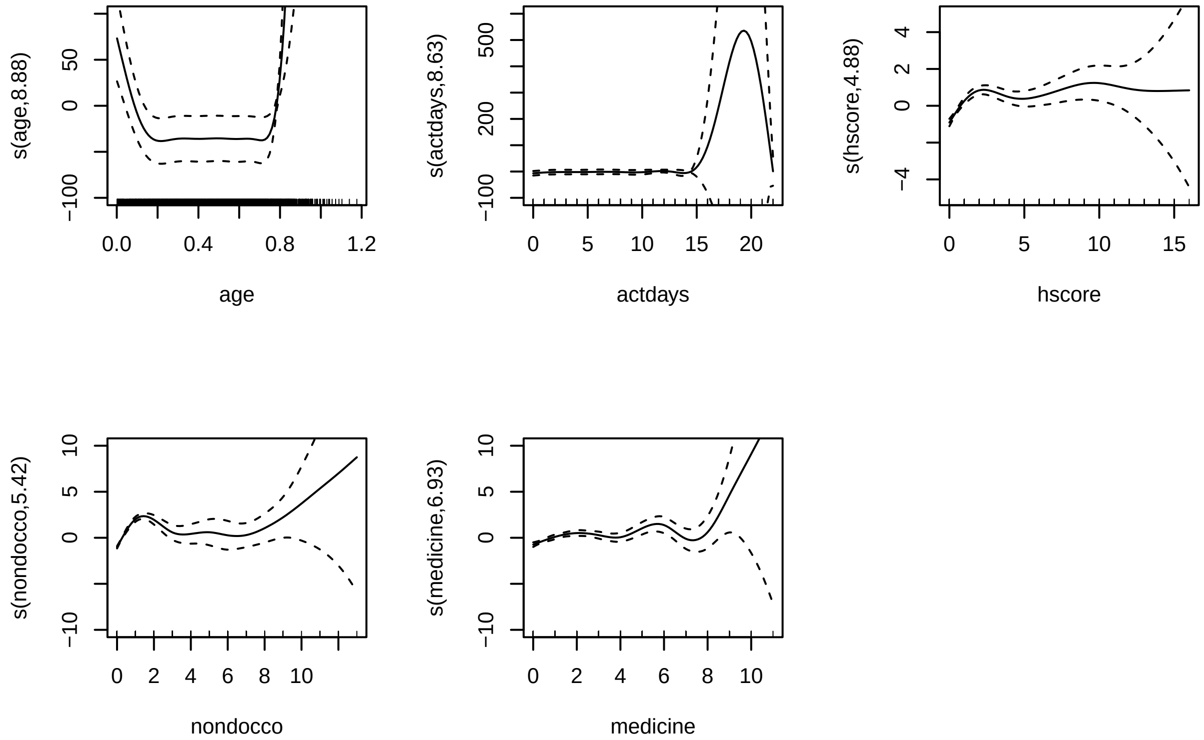
```
model_gam <- gam(ifvisit ~ s(age) + s(actdays) + s(hscore) + s(nondocco) + s(medicine), data=train_data
summary(model_gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## ifvisit ~ s(age) + s(actdays) + s(hscore) + s(nondocco) + s(medicine)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    40.91      13.05   3.135  0.00172 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq p-value
## s(age)       8.882  8.986 140.36  <2e-16 ***
## s(actdays)   8.626  8.860 596.54  <2e-16 ***
## s(hscore)    4.882  5.811  85.30  <2e-16 ***
## s(nondocco)  5.421  6.249 287.02  <2e-16 ***
## s(medicine)  6.930  7.430  49.68  <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.843   Deviance explained = 79.4%
## UBRE = -0.69709  Scale est. = 1         n = 4152
```

For this model five covariates were selected in order to predict the response variable, those being “actdays”, “hscore”, “age”, “medicine” and “nondocco”, all five of them being considered as splines. The selection was done by including all the variables and sequentially cutting those not fit for the model.

We can appreciate a huge improvement over the previous model, getting to a high value of approx. 80% explained deviance.

We can see from the following plot the splines considered:



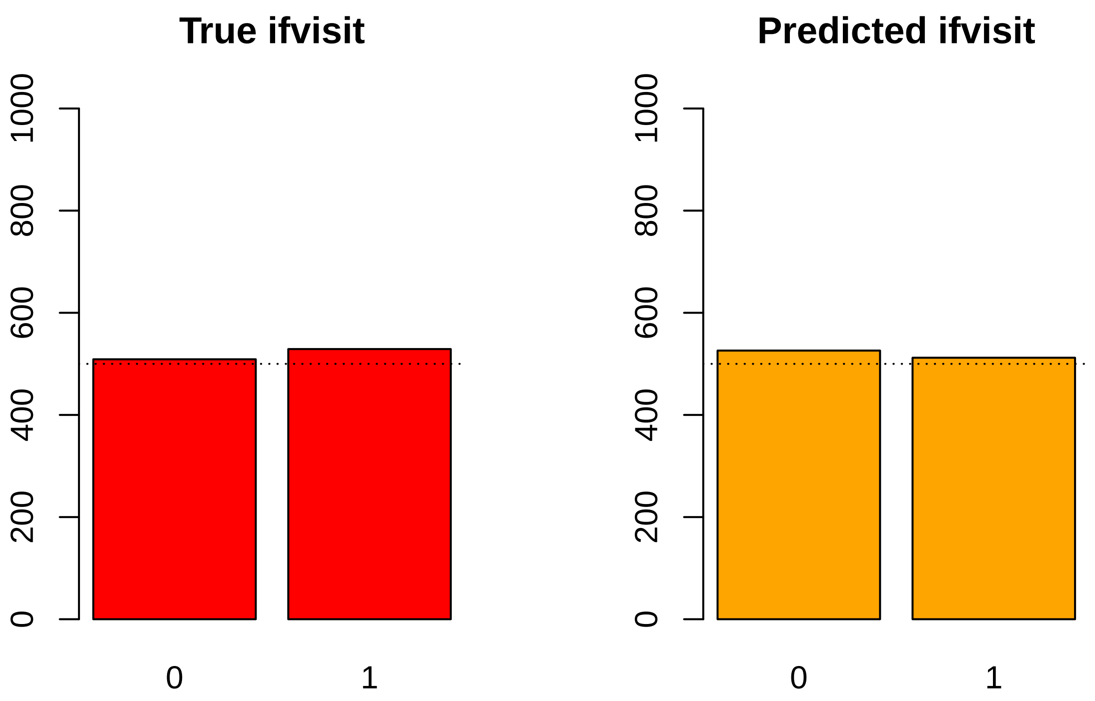
From the “age” spline we can infer that being “young” leads generally to a moderate amount of visits, being “adult” leads to less people going to the doctor, while being “old” implies a big chance of going to a doctor’s visit. This follows the common logic and experience, so it’s a good sign of the model working. Generally, high values of all the covariates implies a high chance of going to the doctor, as can be see in the previous plots.

```
## [1] "AIC (GAM): 1257.7"
```

```
## [1] "MAE (GAM): 0.05491"
```

```
## [1] "Number of true ifvisit: 529"
```

```
## [1] "Number of ifvisit predicted: 512"
```



As we can see from these results, the sum of predicted "ifvisit" gets close to the true value while also maintaining a low MAE value, meaning that the model is predicting correct values pretty consistently.

GLM with ROSE

Now let's try fitting a GLM to the same balanced data set and see if it can compete with the GAM one:

```r
model_glm <- glm(ifvisit ~  hospadmi + nondocco + illness + actdays + prescrib + nonpresc, data = train
summary(model_glm)
```

```
##
## Call:
```

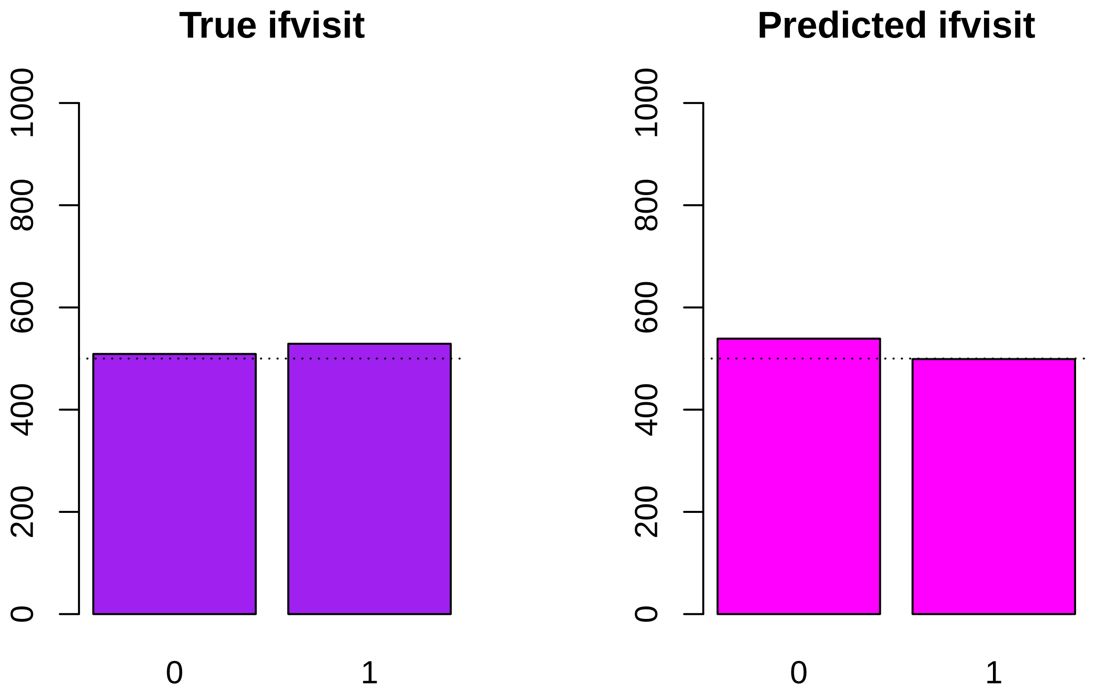
```
## glm(formula = ifvisit ~ hospadmi + nondocco + illness + actdays +
##     prescrib + nonpresc, family = binomial, data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.58171    0.09089 -28.404  < 2e-16 ***
## hospadmi     0.95475    0.09502  10.048  < 2e-16 ***
## nondocco     1.22241    0.08169  14.965  < 2e-16 ***
## illness      0.11996    0.03324   3.609 0.000307 ***
## actdays      0.53808    0.02988  18.005  < 2e-16 ***
## prescrib     0.47077    0.03747  12.563  < 2e-16 ***
## nonpresc     0.25345    0.05829   4.348 1.37e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5749.9  on 4151  degrees of freedom
## Residual deviance: 2947.5  on 4145  degrees of freedom
## AIC: 2961.5
##
## Number of Fisher Scoring iterations: 6
```

Firstly, we can see that the AIC is two times the AIC from the GAM model. Furthermore, the explained deviance is at best around 50%, which is a better result but still not as good as GAM.

```
## [1] "MAE (GLM): 0.1021"
```

```
## [1] "Number of true ifvisit: 529"
```

```
## [1] "Number of ifvisit predicted: 499"
```



From the obtained results we can see that GAM manages to find a good approximation of the total number of visits while also keeping a low value of MAE, meaning that the predictions are correct most of the times.

On the other hand, GLM is a little worse at predicting the total number of visits (sum of “ifvisit”) and it scores double the MAE from GAM meaning that the predictions are overall worse but still useful. GAM manages to understand better the variable interactions but GLM is faster and simpler to interpret.

From this analysis we can say that augmenting a skewed data set such as the one we are analyzing can improve and ease the binary classification problem, and also that the GAM model is much better, in this particular data set, at accurately predicting if a person has gone to the doctor in the past two weeks or not.

This concludes the binary classification digression, from now on all the models will try to predict the whole “doctorco” variable.