

Report

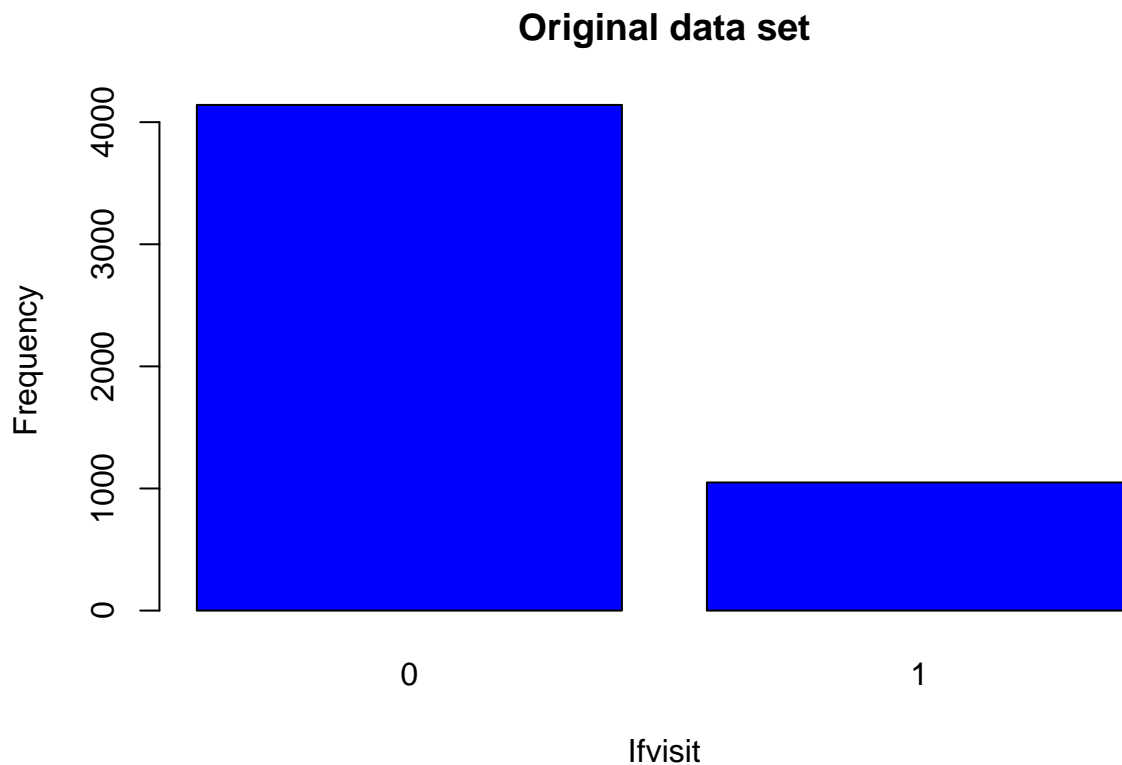
kiki

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 + illness)
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

```

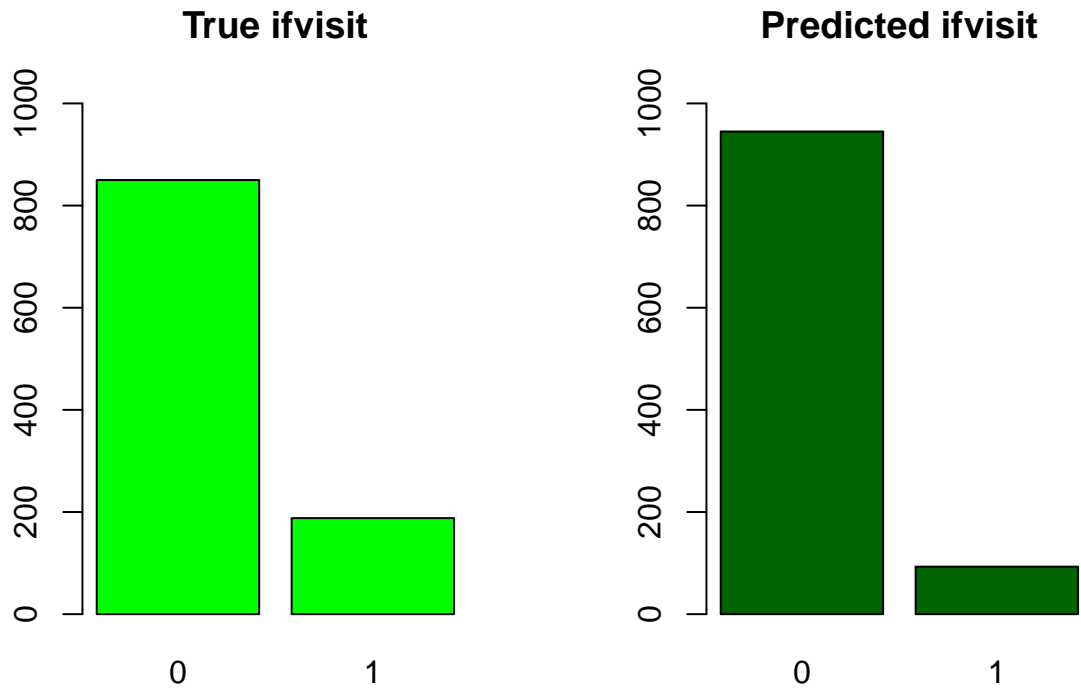
Both the variables “hospdays” and “actdays” were considered as splines after a top to bottom examination of the covariates. The summary shows that we can account for approx. 19% of explained deviance at best, which is not a great result so far.

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

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

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

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

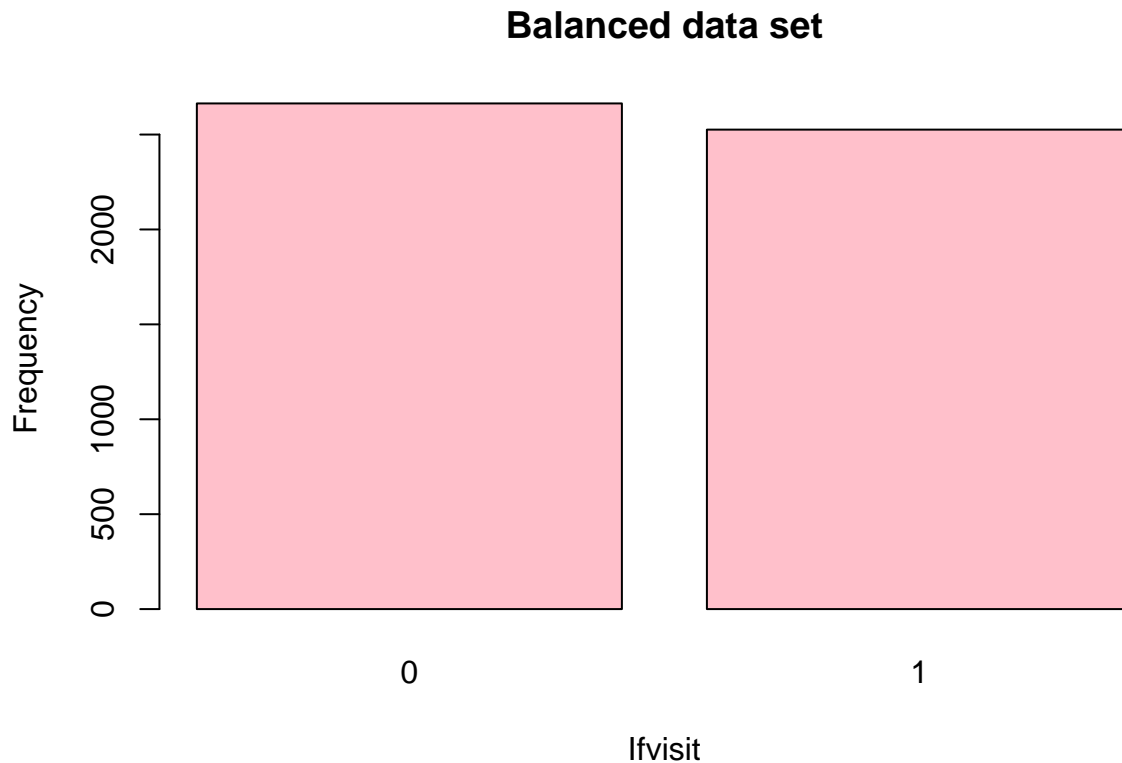


The sum of predicted “ifvisit” yealds a total of 93, against the true value of 188, indicating that GAM isn’t performing very well at predicting the minority class ‘1’. To enhance it’s abilities, we should further manipulate the data set as shown in the following section.

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.

```
data.rose <- ROSE(ifvisit ~ ., data = data, seed = 1, hmult.majo = 0)$data
data.rose <- abs(data.rose)
barplot(table(data.rose$ifvisit), main = "Balanced data set", xlab="Ifvisit", ylab="Frequency", col="pink")
```



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.

GAM with ROSE

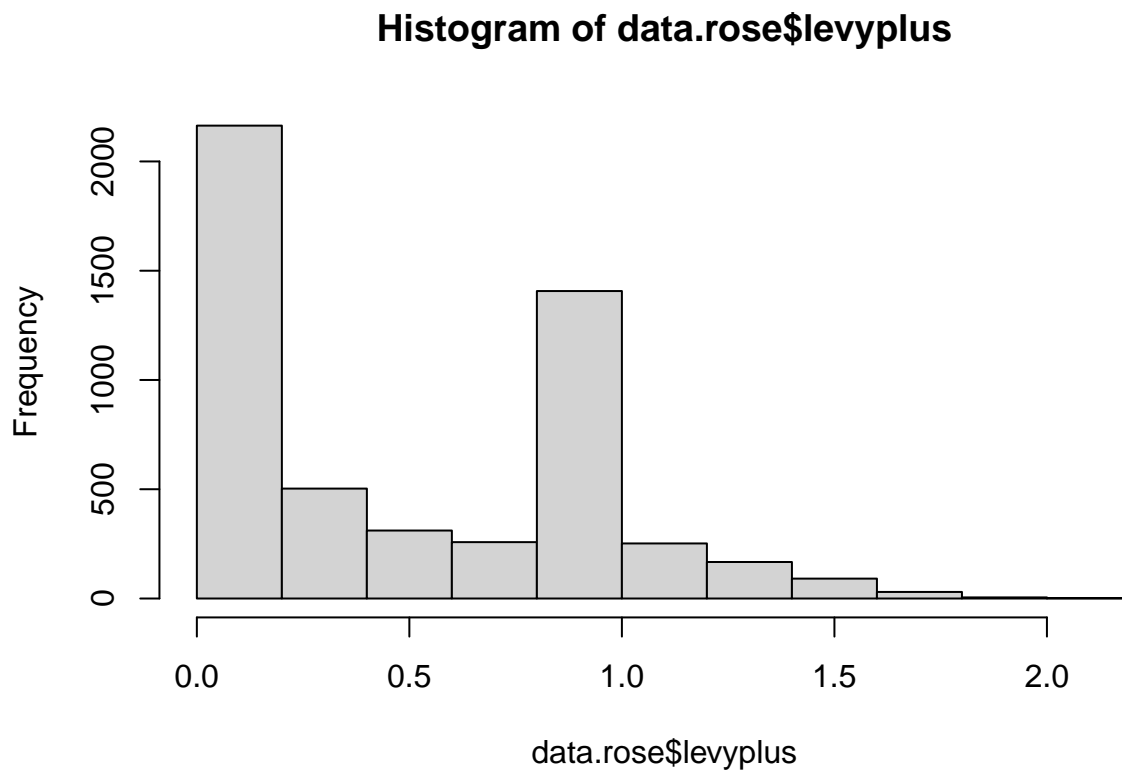
We can try again fitting a GAM on this new balanced data set and see if the performance improved:

```
model_gam <- gam(ifvisit ~ s(actdays) + s(hospadmi) + s(nondocco), data = train_data, family = binomial)
summary(model_gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## ifvisit ~ s(actdays) + s(hospadmi) + s(nondocco)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   214.69      43.68   4.915 8.88e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
```

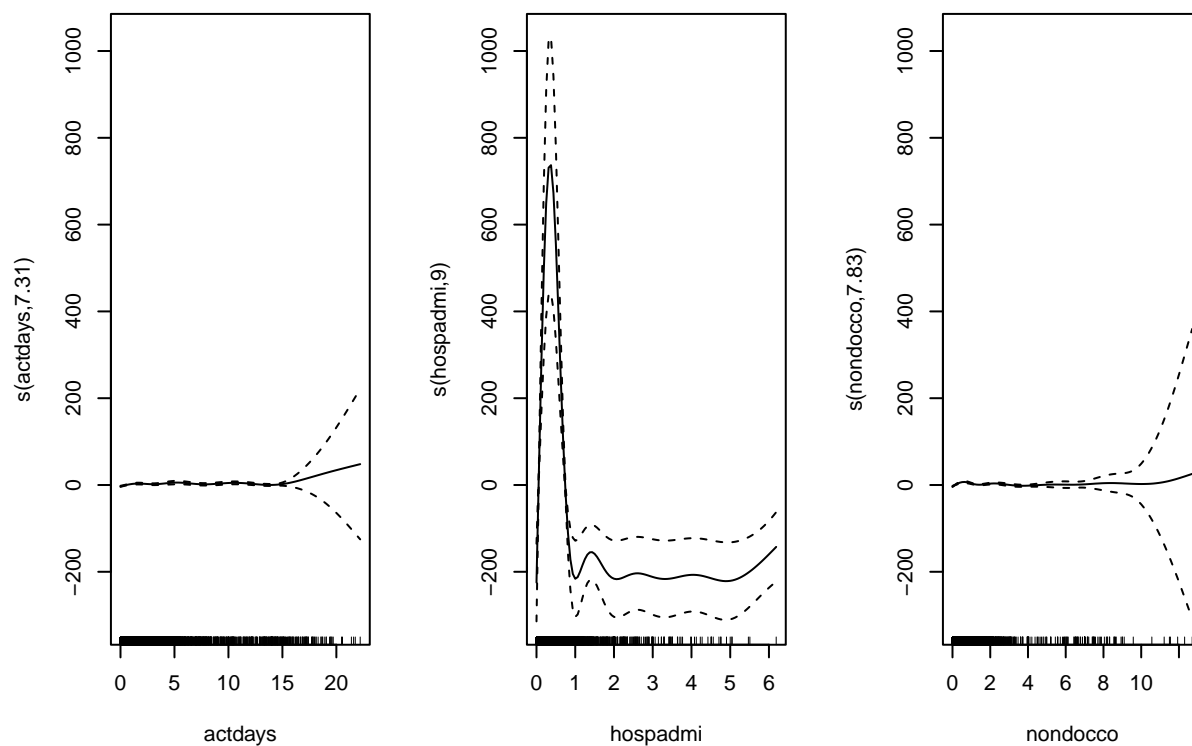
```
##               edf Ref.df Chi.sq  p-value
## s(actdays)  7.314  7.819  29.33 0.000248 ***
## s(hospadmi)  8.999  9.000  41.46 8.80e-07 ***
## s(nondocco)  7.833  8.173  33.83 4.99e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.989   Deviance explained = 98.5%
## UBRE = -0.96751   Scale est. = 1           n = 4152
```

```
hist(data.rose$levyplus)
```



For this model only 3 covariates were selected in order to predict the response variable, those being “actdays”, “hospadmi” and “nondocco”, all three 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 see from the following plot the splines considered:



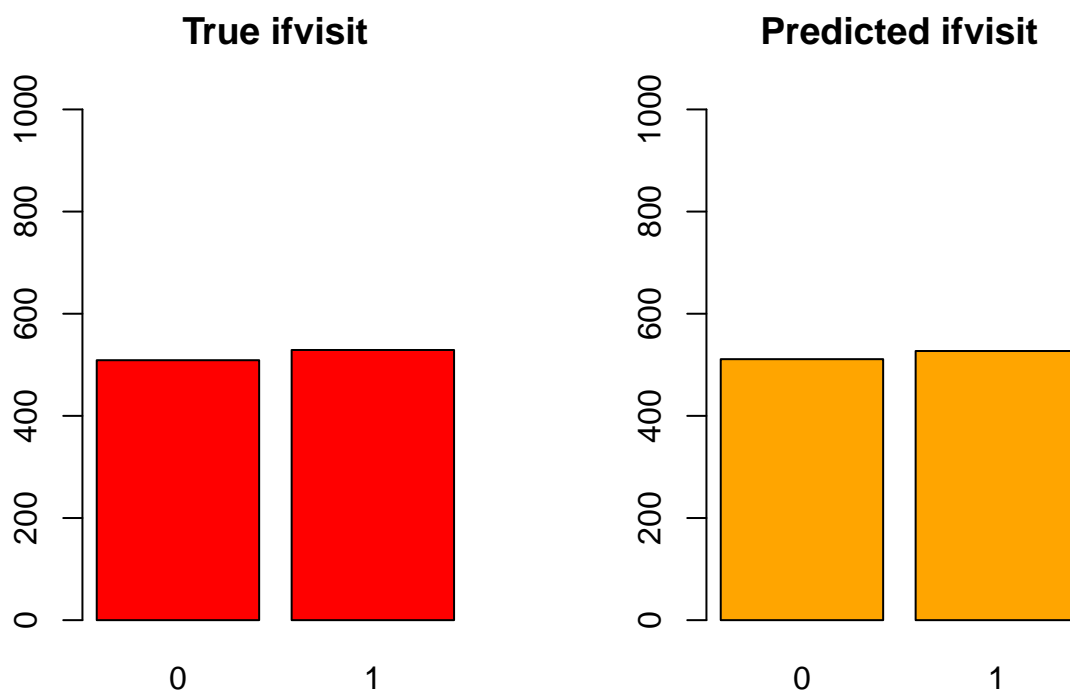
We can appreciate a huge improvement over the previous model, getting to a very high value of approx. 98% of explained deviance.

```
## [1] "AIC (GAM): 134.89"
```

```
## [1] "MAE (GAM): 0.00963"
```

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

```
## [1] "Number of ifvisit predicted: 527"
```



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

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:

```
model_glm <- glm(ifvisit ~ hospadmi + hscore + illness * actdays + prescrib + age_factor + pre_act0, data = train_data, family = binomial)
summary(model_glm)
```

```
##
## Call:
## glm(formula = ifvisit ~ hospadmi + hscore + illness * actdays +
##       prescrib + age_factor + pre_act0, family = binomial, data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.51206    0.07128  -21.213   < 2e-16 ***
## hospadmi       0.27592    0.05776   4.777 1.78e-06 ***
## hscore         0.06489    0.01429   4.542 5.56e-06 ***
## illness        0.13794    0.02557   5.396 6.82e-08 ***
## actdays       0.08999    0.01554   5.790 7.06e-09 ***
## prescrib       0.12983    0.02635   4.928 8.32e-07 ***
## age_factoradult 0.11445    0.08043   1.423  0.1547
## age_factorold   0.68465    0.10499   6.521 6.98e-11 ***
```

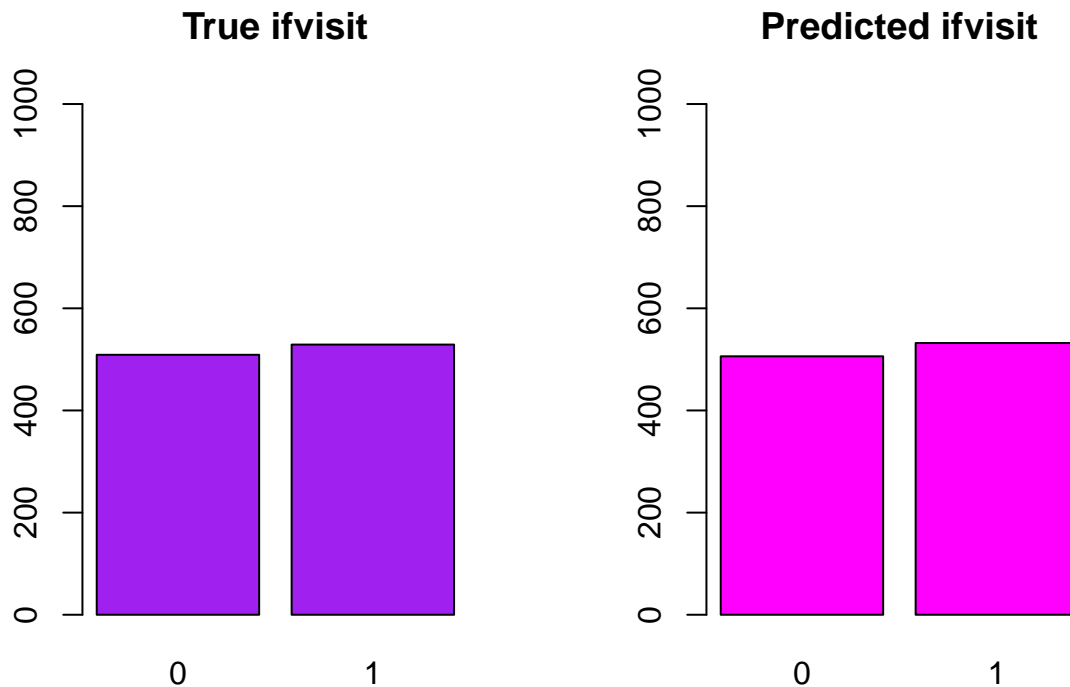
```
## pre_act0          1.14776    0.08010  14.329 < 2e-16 ***
## illness:actdays -0.01178    0.00563  -2.092  0.0364 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 5749.9 on 4151 degrees of freedom
## Residual deviance: 4690.0 on 4142 degrees of freedom
## AIC: 4710
##
## Number of Fisher Scoring iterations: 4
```

Firstly, we can see that the AIC is almost 10 times higher than the AIC from the GAM model. Furthermore, the explained deviance is again really low, not even passing 20%.

```
## [1] "MAE (GLM): 0.2861"
```

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

```
## [1] "Number of ifvisit predicted: 532"
```



From these results obtained we can see that GLM is good at predicting the total number of visits (sum of “ifvisit”) but it scores a high value on the MAE meaning that the predictions are often wrong but sum up to a good approximation.

On the other hand 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.

From this analysis we can say that augmenting a skewed data set such as the one we are analyzing can improve drastically 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.