Statistical Learning Project

Filippo Santin, Gurjeet Singh, Francesca Zen

18/5/2021

1 Introduction

In the following report we present an analysis computed on stroke disease, and we try to explain from statistical analysis some correlation factors and statistics of the given features/predictors by constructing predictive models in order to assess possible linear and non-linear relationships of features (predictors) to predict a stroke disease in a person (predicted variable).

"Stroke" is the medical term for damage to brain tissue or the death of a portion of it, due to insufficient blood supply to an area of the brain.

Our aim is to see if and how the variables we are dealing with are related, in order to predict which individual is more probable to have a stroke.

The symptoms of stroke vary from patient to patient, depending on the severity of the condition, the affected brain area, causes, type of stroke, etc.

Stroke is characterized by sudden onset and for this reason it involves the need for immediate therapeutic intervention and adapted to the needs of the patient. In this sense, looking for relation between features may help to prevent or assess it.

In order to have a guide for the interpretation of the data we underline the following information:

- The normal values of glucose level are between 60 and 110 mg/dl and with a value greater than 126 mg/dl a person is considered diabetic;
- a body mass index (BMI) between 18.5-24.9 indicates a normal/healthy weight, below 18.5 indicates underweight, 25.0-29.9 indicates overweight and above 30.0 indicates obese person.

2 Exploring the Dataset

The dataset we used is provided by Kaggle [1] and it is composed of 5,110 entries with a total of 12 columns: `id`, `gender`, `age`, `hypertension`, `heart_disease`, `ever_married`, `work_type`, `Residence_type`, `avg_glucose_level`, `bmi`, `smoking_status`, `stroke`.

```
library(knitr)
stroke_data <- read.csv('healthcare-dataset-stroke-data.csv')
kable(stroke_data[1:5,], format = 'simple', align='ccccccccc',
      col.names = c('id','gender','age', 'hypert.', 'hd' ,'ev_marr',
                    'work_type','res_type','glucose', 'bmi','smoking','stroke'))
```

id	gender	age	hypert.	hd	ev_marr	work_type	res_type	glucose	bmi	smoking	stroke
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1

[1] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

id	gender	age	hypert.	hd	ev_marr	work_type	res_type	glucose	bmi	smoking	stroke
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1

2.1 Preprocessing

The preliminary part of the analysis focuses on the study of the dataset and its pre-processing: we looked at the `id` column and verified that all the data collected was referring to different people, thus no recidivist status were involved. After this check we removed the column from the dataset as it does not hold useful information for our study.

```
stroke_data<-stroke_data[,-1]
```

In order to use the variables through the analysis we transformed the categorical variables into factors:

```
stroke_data$gender<- as.factor(stroke_data$gender)
stroke_data$ever_married<-as.factor(stroke_data$ever_married)
stroke_data$work_type<-as.factor(stroke_data$work_type)
stroke_data$Residence_type<-as.factor(stroke_data$Residence_type)
stroke_data$smoking_status<-as.factor(stroke_data$smoking_status)
```

In addition, the variable `bmi` was not numeric because of the presence of "N/A" string values which identify missing information, hence we transformed it into numeric values and then removed the NA values generated.

```
stroke_data$bmi <- as.numeric(stroke_data$bmi)
```

```
## Warning: NA introdotti per coercizione
```

```
stroke_data<- na.omit(stroke_data)
```

We ended up having 4,909 entries and 11 total columns. Here we give a quick overview of the main information about the dataset:

```
summary(stroke_data)
```

```
##     gender          age         hypertension     heart_disease    ever_married
##  Female:2897   Min.   : 0.08   Min.   :0.00000   Min.   :0.0000   No :1705
##  Male  :2011   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.0000   Yes:3204
##  Other :   1   Median :44.00   Median :0.00000   Median :0.0000
##                Mean   :42.87   Mean   :0.09187   Mean   :0.0495
##                3rd Qu.:60.00   3rd Qu.:0.00000   3rd Qu.:0.0000
##                Max.   :82.00   Max.   :1.00000   Max.   :1.0000
##          work_type    Residence_type avg_glucose_level      bmi
##  children     : 671   Rural:2419     Min.   : 55.12    Min.   :10.30
##  Govt_job     : 630   Urban:2490     1st Qu.: 77.07    1st Qu.:23.50
##  Never_worked :  22                  Median : 91.68    Median :28.10
##  Private      :2811                  Mean   :105.31    Mean   :28.89
##  Self-employed: 775                  3rd Qu.:113.57    3rd Qu.:33.10
##                                      Max.   :271.74    Max.   :97.60
##          smoking_status     stroke
##  formerly smoked: 837   Min.   :0.00000
##  never smoked   :1852   1st Qu.:0.00000
##  smokes         : 737   Median :0.00000
##  Unknown        :1483   Mean   :0.04257
##                         3rd Qu.:0.00000
##                         Max.   :1.00000
```

2.2 Descriptive Statistic

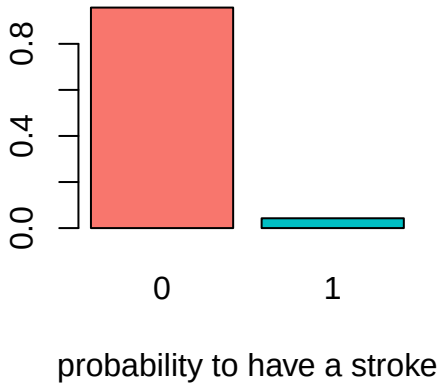
```
attach(stroke_data)
```

In order to highlight and study better the data, we used some plots to study their statistics and distribution. A relevant and important information is provided by the following barplot, in which we see an unbalance dataset issue: 209 people on a total of 4909 get a stroke, i.e. the 4.25 % of the people.

```
table(stroke)/dim(stroke_data)[1]
```

```
## stroke
##          0          1
## 0.95742514 0.04257486
```

```
barplot(table(stroke)/dim(stroke_data)[1],
        xlab='probability to have a stroke',col = c('#F8766D','#00BFC4'))
```

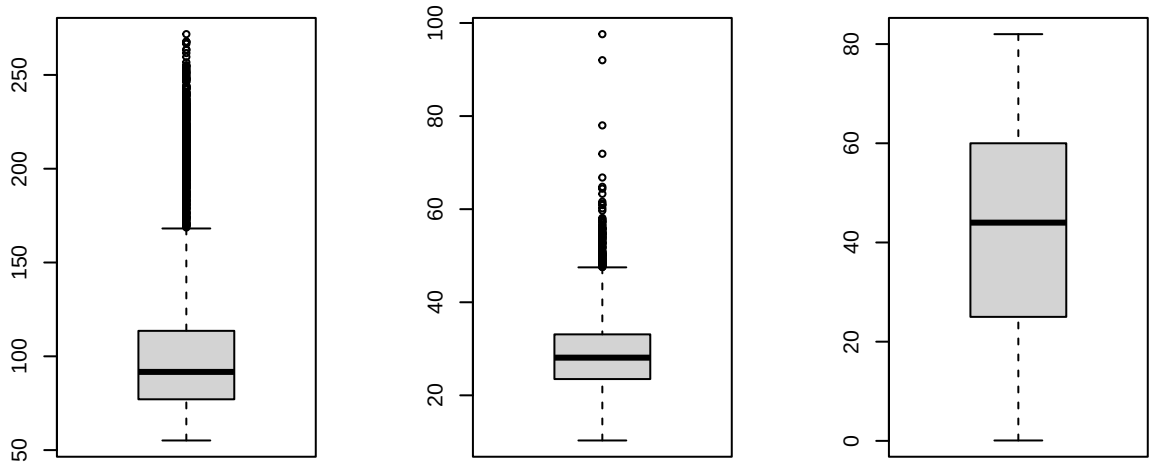


This value is representative of the real situation in which there are not many stroke cases compared with the whole population. The incidence of stroke in Europe at the beginning of the 21st century varies from 95 to 290 cases/100,000[2]. Furthermore in many clinical diseases analysis this issue is commonly present.

A visual transformation of the values seen in the `summary` function is provided in the following boxplots:

```
par(mfrow=c(1,3))
boxplot(avg_glucose_level, xlab= 'average glucose level' )
boxplot(bmi, xlab = 'body mass index')
boxplot(age, xlab = 'age',pch=20)
```

[2]QUADERNI dell'Italian Journal of Medicine, A Journal of Hospital and Internal Medicine, Michele Meschi,volume 8, issue 2, March-April 2020




```
par(mfrow=c(1,1))
```

From above we can see that in the first two boxplots (starting from the left) there are lots of outliers, that can also be see from the summary looking at the difference between the third quantile and the maximum value in the `avg_glucose_level` and `bmi` variables. Actually, they represent real-scenario (few people are affected by high glucose levels) and possible interesting cases of pathologies bounded with diabetes. Hence these data points have to be considered during modeling, they could be helpful to predict stroke cases since as mentioned in the medical literature stroke could be also due to complications of diabetes.

In order to compare entities in pairs and judge which of each entity is preferred, or has a greater amount of some quantitative property we provide a pair-wise plot. In addition, to involve also the categorical variables we wrote some useful function:

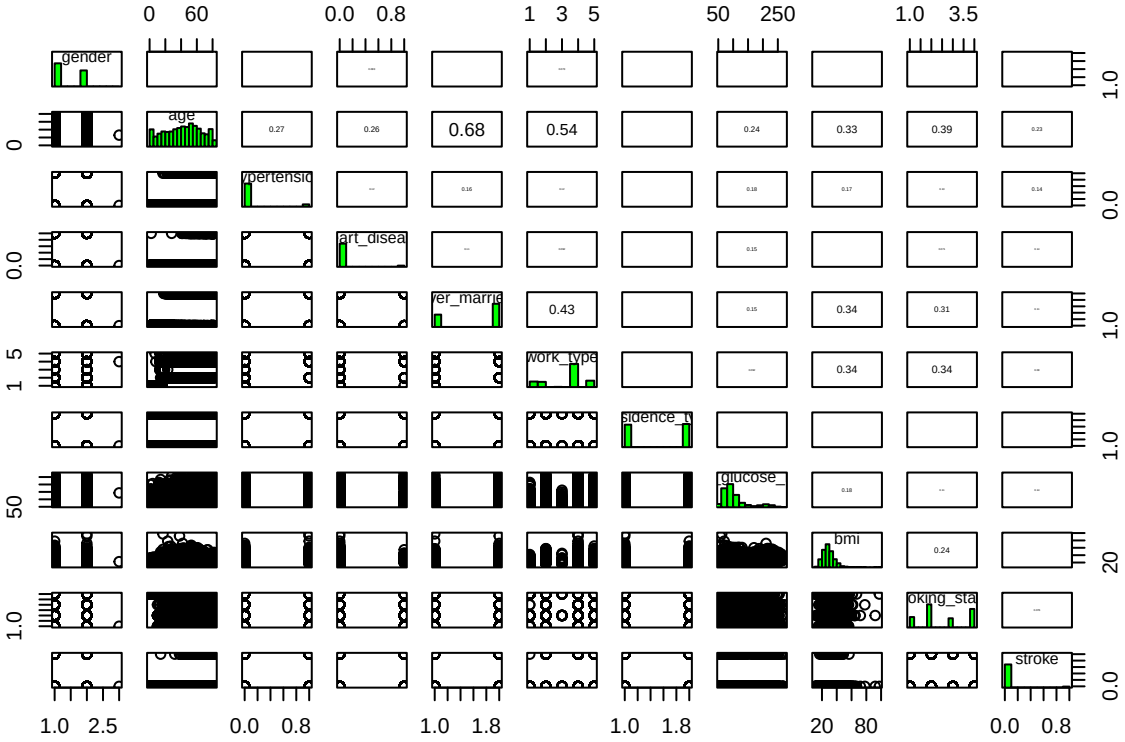
```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "green", ...)
}
```

```
box_plot_categories <- function(data, y){
  n_features = length(data)
  grid = round(sqrt(n_features))
  print(grid)
  par(mfrow=c(grid, grid))
  names = colnames(data)
  for (idx in c(1:n_features)) {
    plot(y ~ data[, idx], xlab=names[idx], main=c('Boxplot y ~ ', names[idx]))
  }
  par(mfrow=c(1, 1))
}
```

And here we show the results from the pairs plot:

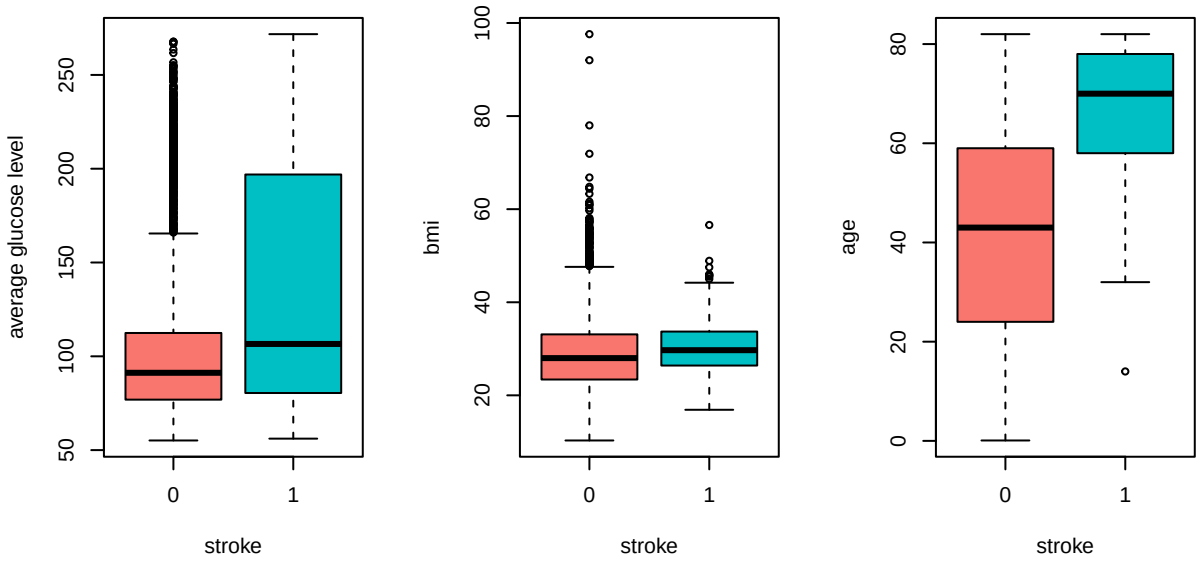
```
pairs(stroke_data, diag.panel=panel.hist, upper.panel=panel.cor)
```



The plot shows that the stronger relationships involve quite often the variable `age`. There are also other relevant relation between `work_type` and `ever_married` plus `bmi` with `working_status`. In addition, we can see strong collinearity among th dummy variables `age`, `work_type`, and `ever_married`, so they do not contribute in the fitting.

We go on looking at some intuitive relation of `stroke` with `age`,`bmi` and `avg_glucose_level`:

```
par(mfrow=c(1,3))
boxplot(avg_glucose_level~stroke, xlab= 'stroke',
        ylab = 'average glucose level', col = c('#F8766D','#00BFC4'))
boxplot(bmi~stroke, xlab = 'stroke', ylab = 'bmi',col = c('#F8766D','#00BFC4'))
boxplot(age~stroke, xlab='stroke' ,ylab = 'age',col = c('#F8766D','#00BFC4'))
```

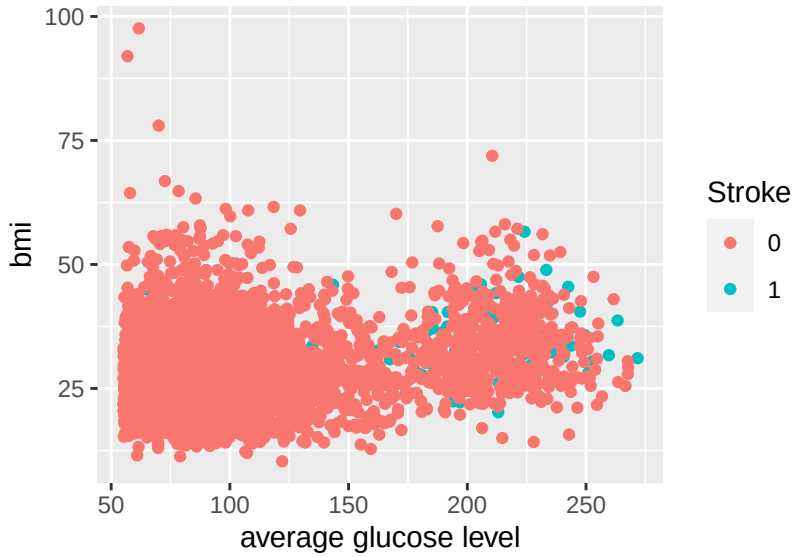



```
#boxplot(heart_disease~stroke, xlab='stroke' ,ylab = 'heart_disease')
par(mfrow=c(1,1))
```

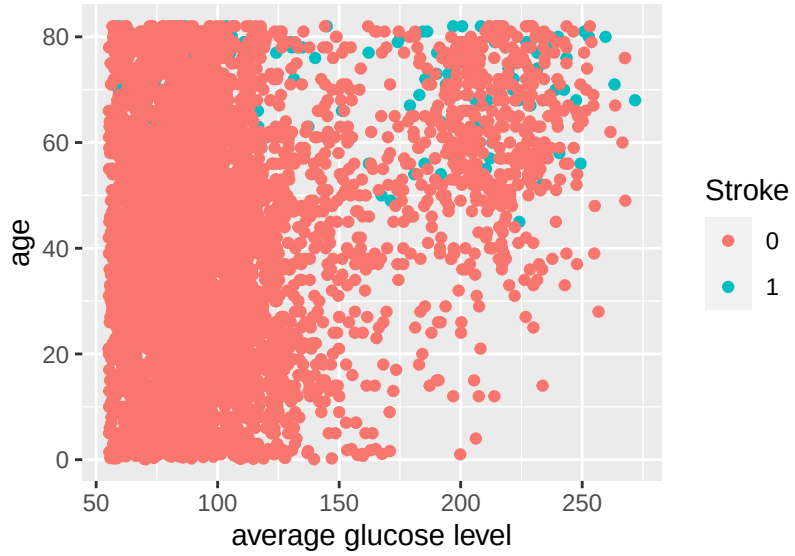
Looking at these plots we can see that the incidence of the disease increases progressively with age and that if we sum also the information about `avg_glucose_level` we may wonder if diabetic people are more probable to get a stroke or not. In addition there is no apparent relation of `stroke` with `bmi`.

We now highlight other visual relationship between the variables used before, thanks to the scatter plots:

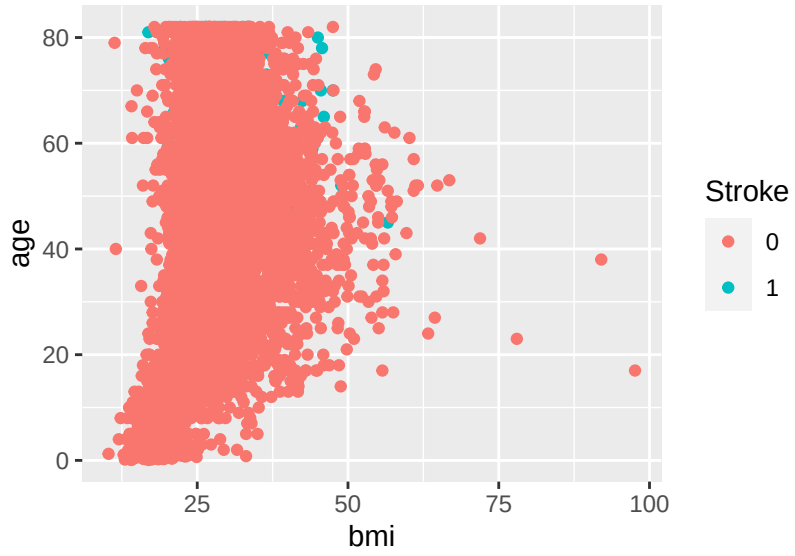
```
library(ggplot2)
ggplot(stroke_data, aes(x = avg_glucose_level, y = bmi,col = as.factor(stroke))) +
  labs(x = "average glucose level", y = "bmi", color = "Stroke") + geom_point()
```




```
ggplot(stroke_data, aes(x = avg_glucose_level, y = age,col = as.factor(stroke))) +
  labs(x = "average glucose level",y = "age", color = "Stroke") + geom_point()
```




```
ggplot(stroke_data, aes(x = bmi, y = age, col = as.factor(stroke))) +
  labs(x = "bmi", y = "age", color = "Stroke") +geom_point()
```



Difficult to read data and give a clear classification/explanation.

Even with high `avg_glucose_level` and `bmi` it's not so straightforward to detect the stroke, since they could not be so strictly related to disease but maybe correlated to other illnesses linked (or not) to it. In the end it seem to be not so simple to identify the direct relationship with the stroke while dealing with the features that we have.

be related to other disease (which are not correlated with stroke????) or there are still not enough complications to develop a stroke. Insomma l'associazione non e' cosi diretta e il problema sembra essere complicato perche non descrive una chiara classificazione guardando i punti.

At this point we can ask some questions:

- Is it possible to prevent ictus?
- Which factors are the most related to it?
- How strong are the relations between the features?

- Are the given variables enough to predict a good accuracy of some possible person affected by ictus?

We will explore the data trying to answer them.

3 Modeling

We will now present some different approaches for the classification of the data.

3.1 Logistic Regression

In this part of the predictive analysis we will present three different type of models, compared to discover the best one that can better interpret the data.

While going on with the classification using logistic regression we could meet the following problems: non-linearity of the data, correlation of error terms, heteroschedasticity, outliers, leverage point and collinearity.

3.1.1 Full and Reduced Models

We start with the full model to see if all the features of the dataset contribute positively/negatively on the prediction of a stroke.

```
mod.full <- glm(stroke~., data=stroke_data, family = binomial)
summary(mod.full)
```

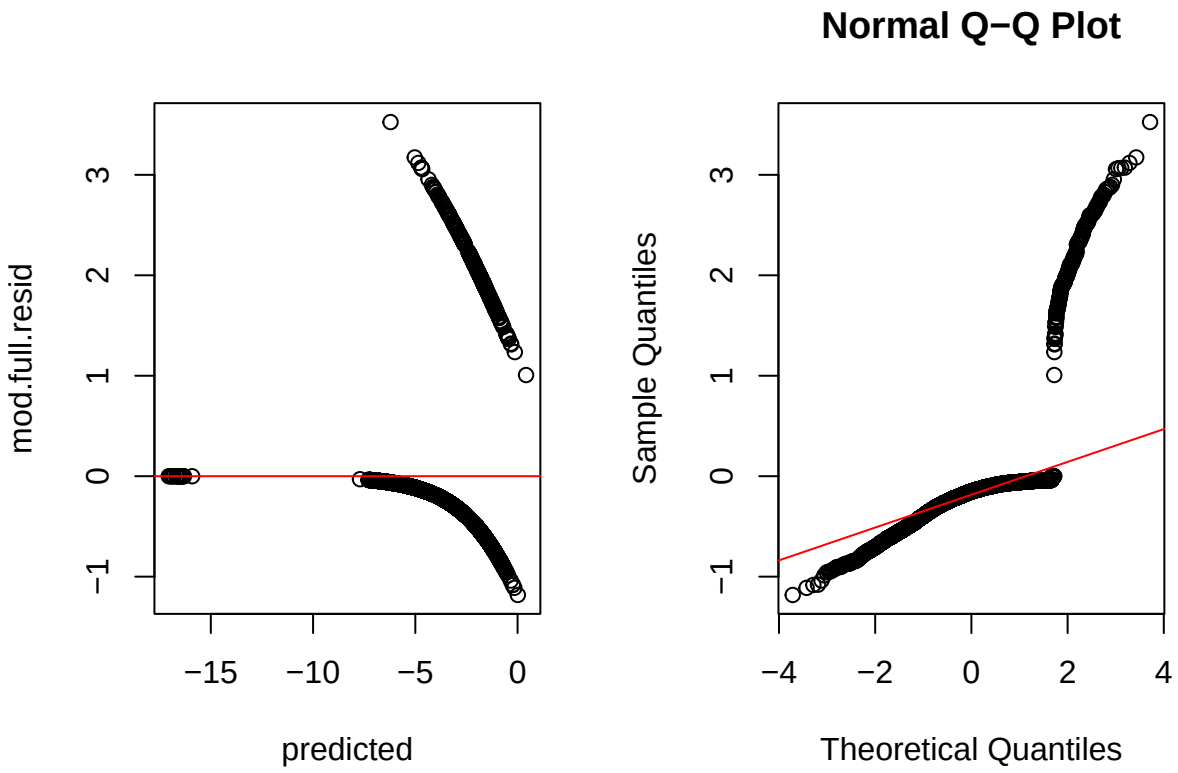
```
##
## Call:
## glm(formula = stroke ~ ., family = binomial, data = stroke_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1823  -0.2947  -0.1524  -0.0744   3.5251
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -7.360e+00  1.067e+00  -6.895 5.37e-12 ***
## genderMale                  -1.463e-02  1.544e-01  -0.095 0.924525
## genderOther                 -1.135e+01  2.400e+03  -0.005 0.996225
## age                          7.348e-02  6.347e-03  11.578  < 2e-16 ***
## hypertension                 5.249e-01  1.750e-01   2.999 0.002711 **
## heart_disease                3.488e-01  2.072e-01   1.683 0.092381 .
## ever_marriedYes             -1.152e-01  2.473e-01  -0.466 0.641394
## work_typeGovt_job           -6.817e-01  1.114e+00  -0.612 0.540660
## work_typeNever_worked       -1.082e+01  5.090e+02  -0.021 0.983036
## work_typePrivate            -5.208e-01  1.100e+00  -0.473 0.635943
## work_typeSelf-employed      -9.459e-01  1.119e+00  -0.845 0.397906
## Residence_typeUrban          4.514e-03  1.500e-01   0.030 0.975990
## avg_glucose_level            4.652e-03  1.294e-03   3.595 0.000324 ***
## bmi                          4.062e-03  1.188e-02   0.342 0.732387
## smoking_statusnever smoked -6.722e-02  1.886e-01  -0.356 0.721556
## smoking_statussmokes         3.139e-01  2.295e-01   1.368 0.171310
## smoking_statusUnknown       -2.753e-01  2.471e-01  -1.114 0.265193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1363.2  on 4892  degrees of freedom
## AIC: 1397.2
##
## Number of Fisher Scoring iterations: 15
```

Here we see that `age`, `avg_glucose_level` and `hypertension` are the variables most related to `stroke`.

Let's use the residual plots to get more information about this model (we use `type="deviance"` because we have a binary response):

```
mod.full.resid <- residuals(mod.full, type="deviance")
predicted <- predict(mod.full, type = "link")
par(mfrow=c(1,2))
plot(mod.full.resid~predicted)
abline(h=0, col='red')
qqnorm(mod.full.resid)
qqline(mod.full.resid, col='red')
```




```
par(mfrow=c(1,1))
```

The residual plots are not satisfactory because it is not easy to interpret them. From the right plot we can see that the data are not normal.

We now make some test in order to find the best reduced model: we start from the full model and then remove all the features that have collinearity between each other,i.e. `work_type`, `Residence_type` and `ever_married`.

```
mod.red1 <- glm(stroke ~ age + bmi + avg_glucose_level + hypertension +
                  smoking_status + gender + heart_disease, family=binomial)
```

Also in this case the variables more important are the same of the ones found in the full moel but also `heart_disease` seems to contribute to the prediction. In this step we try to remove the `gender` variable:

```
mod.red2 <- glm(stroke ~ age + bmi + avg_glucose_level + hypertension +
                  smoking_status  + heart_disease, family=binomial)
summary(mod.red2)
```

The variables left to remove are `bmi` and `smoking_status`, and we ended up with the final reduced model:

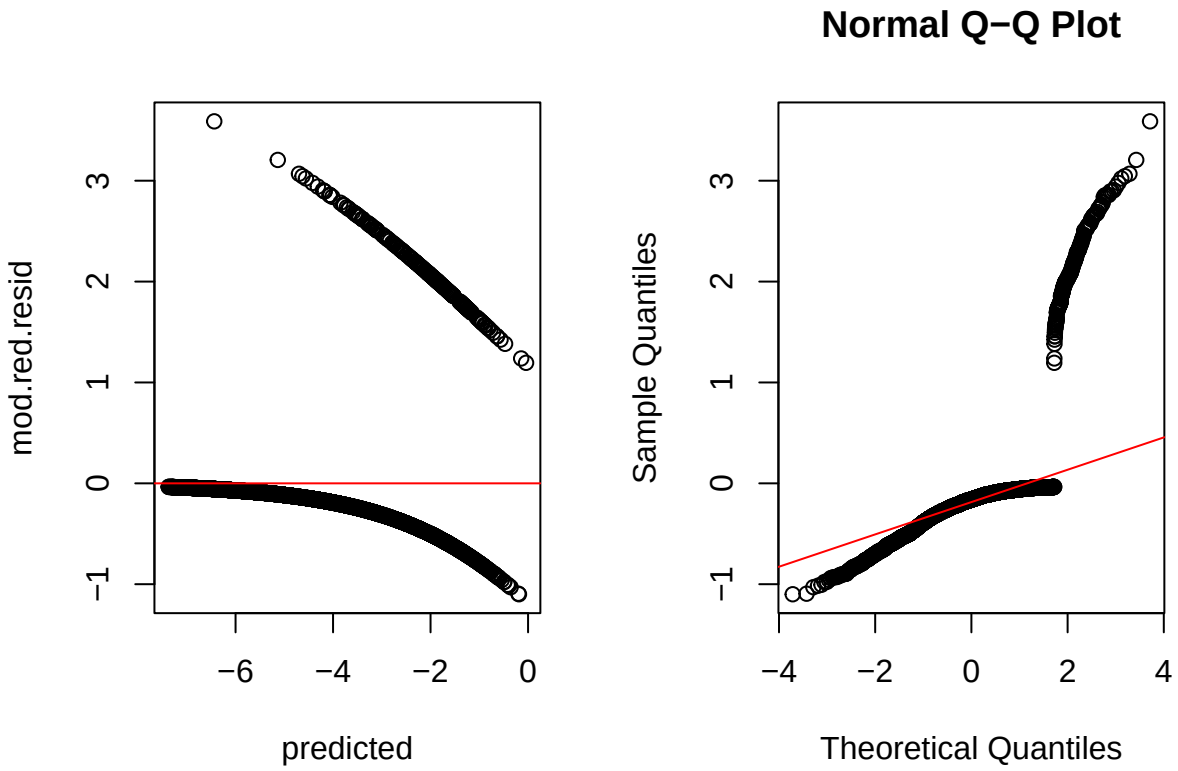
$$\texttt{stroke} = \beta_0 + \beta_1\,\texttt{age} + \beta_2\,\texttt{heart\_disease} + \beta_3\,\texttt{avg\_glucose\_level} + \beta_4\,\texttt{hypertension}$$

```
mod.red <- glm(stroke~age + heart_disease + avg_glucose_level+ hypertension,
               data=stroke_data, family = binomial)
summary(mod.red)
```

```
##
## Call:
## glm(formula = stroke ~ age + heart_disease + avg_glucose_level +
##     hypertension, family = binomial, data = stroke_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0995  -0.2940  -0.1599  -0.0778   3.5885
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.660740   0.387152 -19.787  < 2e-16 ***
## age                0.067547   0.005571  12.124  < 2e-16 ***
## heart_disease      0.404298   0.203447   1.987 0.046895 *
## avg_glucose_level  0.004802   0.001255   3.828 0.000129 ***
## hypertension       0.539613   0.173055   3.118 0.001820 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1374.6  on 4904  degrees of freedom
## AIC: 1384.6
##
## Number of Fisher Scoring iterations: 7
```

For this model we also give a descriptive statistic through the residual plots:

```
mod.red.resid <- residuals(mod.red, type="deviance")
predicted <- predict(mod.red, type = "link")
par(mfrow=c(1,2))
plot(mod.red.resid~predicted)
abline(h=0, col='red')
qqnorm(mod.red.resid)
qqline(mod.red.resid, col='red')
```




```
par(mfrow=c(1,1))
```

We can see from the residual vs predicted values the presence of high non-linearity in the dataset. In the qqplot instead we see that residuals do no follow a normal distribution.

Positive coefficient estimates —> positive association. So, the larger the value, the higher is the estimated probability of stroke.

Instead in the standard deviance vs predicted we can see that homoscedasticity does not hold since the line of the standard residual is not flat, hence even by standardizing the residual we end up having high variance among residuals.

In the end by looking at the leverage plot, we see the presence of some sample with high leverage values (bottom right), which could influence the prediction of the model.

Buuut I don't know in which range of levarage value is considered to change a lot the prediction of the model. Furthermore R does not show the index of the sample with high leverage, *I guess because a lot of values could change the prediction.*

Some outliers with high variance are: idx: 119, 183, 246.

In the end, if we compare the results obtained from the full and reduced models we can say that the reduced seems to make a more accurate prediction and fits better the data, also the AIC is lower than the one of the full model. To have a confirmation of this, we use the anova function, which compare the two models and returns the better between the two.

```
anova(mod.full, mod.red, test="Chisq")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: stroke ~ gender + age + hypertension + heart_disease + ever_married +
##     work_type + Residence_type + avg_glucose_level + bmi + smoking_status
## Model 2: stroke ~ age + heart_disease + avg_glucose_level + hypertension
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      4892     1363.2
## 2      4904     1374.7 -12   -11.42   0.4933
```

As expected from the anova test rejects that the complex model is more significant than the reduced one, since the p-value is not less than 5%. Hence the full model does not help with our prediction.

Outliers of the reduced model:

	gender	age	hypert.	hd	ev_marr	work_type	res_type	glucose	bmi	smoking	stroke
119	Female	38	0	0	No	Self-employed	Urban	82.28	24.0	formerly smoked	1
183	Female	32	0	0	Yes	Private	Rural	76.13	29.9	smokes	1
246	Female	14	0	0	No	children	Rural	57.93	30.9	Unknown	1

3.1.2 Interaction

In order to see which variables were relevant on our research we tested various models using the mixed approach: we started by the reduced model `mod.red` and then tested some interaction between the explanatory variables for improving the performance of the model.

We started with `mod.red`, which had `stroke` as response and `age`, `avg_glucose_level`,`hypertension` and `heart_disease` as predictors. We recall that its AIC was 1384.6.

We then consider the interaction of `age` with the other numerical features, i.e. `avg_glucose_level`,`hypertension` and `heart_disease`: we find out that only `age*heart_disease` was relevant between the ones tested, with an AIC = 1384, which was lower than the one of the reduced model `mod.red`.

```
mod1 <- glm(stroke~age + avg_glucose_level+ heart_disease+ hypertension +
              age*heart_disease, family=binomial)
summary(mod1)
```

```
##
## Call:
## glm(formula = stroke ~ age + avg_glucose_level + heart_disease +
##     hypertension + age * heart_disease, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9897  -0.2980  -0.1557  -0.0737   3.6232
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.816578   0.407175 -19.197  < 2e-16 ***
## age                0.070133   0.005889  11.908  < 2e-16 ***
## avg_glucose_level  0.004702   0.001253   3.752 0.000176 ***
## heart_disease      2.765299   1.396557   1.980 0.047694 *
## hypertension       0.536550   0.172602   3.109 0.001880 **
## age:heart_disease -0.032872   0.019486  -1.687 0.091604 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1372.0  on 4903  degrees of freedom
## AIC: 1384
##
## Number of Fisher Scoring iterations: 7
```

We went on considering all the interaction of `avg_glucose_level` with the remaining predictors and find out that `avg_glucose_level*hypertension` was the best of the possible interaction but cannot improve the previous model, it had an AIC of 1385.9.

```
mod2 <- glm(stroke~age + avg_glucose_level+ heart_disease+hypertension +
              avg_glucose_level*hypertension, family=binomial)
summary(mod2)
```

In the end it was left the interaction `heart_disease*hypertension` which return an AIC 1384.5 for the model.

```
##
## Call:
## glm(formula = stroke ~ age + avg_glucose_level + heart_disease +
##     hypertension + heart_disease * hypertension, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0079  -0.2951  -0.1584  -0.0774   3.5900
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -7.657815   0.387787 -19.747  < 2e-16 ***
## age                          0.067108   0.005592  12.001  < 2e-16 ***
## avg_glucose_level            0.004760   0.001256   3.791 0.000150 ***
## heart_disease                0.592399   0.236015   2.510 0.012073 *
## hypertension                 0.660347   0.189663   3.482 0.000498 ***
## heart_disease:hypertension -0.635251   0.444325  -1.430 0.152803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1372.5  on 4903  degrees of freedom
## AIC: 1384.5
##
## Number of Fisher Scoring iterations: 7
```

We then tried to see the results of the `mod.red` without `heart_disease` which was the explanatory variables with higher p-value, and then we added some interaction terms, starting with the one with `age`:

```
mod4 <- glm(stroke~age + avg_glucose_level + hypertension +
              age*hypertension, family=binomial)
summary(mod4)
```

The AIC of this model was 1386.2, higher that the ones seen on the previous tested models. We go further testing also the interaction add `avg_glucose_level*hypertension`

```
mod5 <- glm(stroke~age + avg_glucose_level + hypertension +
              avg_glucose_level*hypertension, family=binomial)
```

```
summary(mod5)
```

It had an AIC = 1387.6. In the end we return on our base model, `mod.red` and add the teo best iteractions, i.e. the two terms that reduced the AIC term:

```
mod6 <- glm(stroke~age + avg_glucose_level+ heart_disease+ hypertension +
              age*heart_disease + heart_disease*hypertension, family=binomial)
summary(mod6)
```

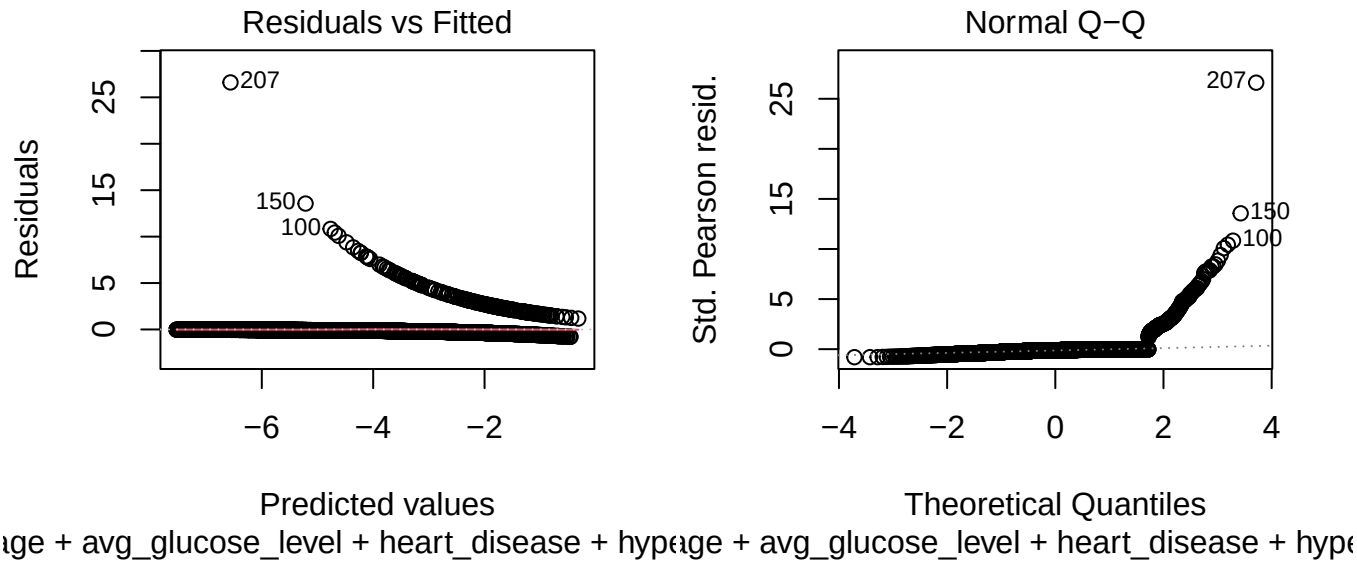
It has AIC = 1384.2 but the interactions had a p-value greater than 0.1 and it didn't seem to represent our data properly. At the end of all we promote `mod1` as the model which better fits our data.

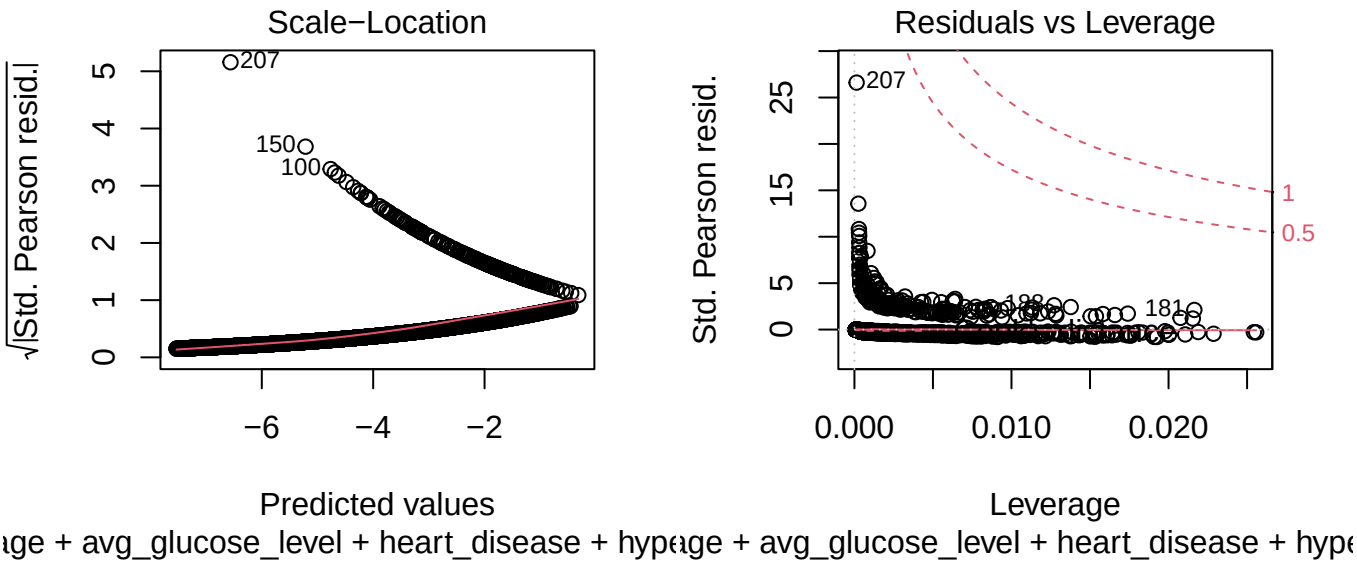
Let's now see some relevant information, such as outliers on the `mod1`:

	gender	age	hypert.	hd	ev_marr	work_type	res_type	glucose	bmi	smoking	stroke
207	Female	81	0	0	Yes	Private	Rural	80.13	23.4	never smoked	1
150	Female	70	0	1	Yes	Private	Rural	239.07	26.1	never smoked	1
100	Female	69	0	0	Yes	Govt_job	Urban	82.81	28.0	never smoked	1

but also leverage point and collinearity:

```
plot(mod1)
```





3.1.3 Polynomial models

We tried to use a polynomial model starting from the reduced model `mod.red` with also `bmi` as predictors. Then we contribute with the square of `bmi`, `avg_glucose_level` and then both for the `mod.poly1`,`mod.poly2` and `mod.poly3` respectively.

```
mod.poly1 <- glm(stroke~age + heart_disease + avg_glucose_level+ hypertension+bmi+
                      I(bmi^2), family = binomial)
summary(mod.poly1)
mod.poly2 <- glm(stroke~age + heart_disease + avg_glucose_level+ hypertension+bmi+
                    + I(avg_glucose_level^2), family = binomial)
summary(mod.poly2)
mod.poly3 <- glm(stroke~age + heart_disease + avg_glucose_level+ hypertension+bmi+
                      I(bmi^2) + I(avg_glucose_level^2), family = binomial)
summary(mod.poly3)
```

At the end of the tests nothing interesting appeared with polynomial models. There were no improvement in the results.

3.2 LDA

Assumption: samples are normally distributed and have same variance in every class => strong assumption.

```
library(MASS)
lda.fit <- lda(stroke~age+bmi+avg_glucose_level+hypertension+work_type+gender
               +smoking_status+ever_married+Residence_type + heart_disease)
lda.pred <- predict(lda.fit)
table(lda.pred$class, stroke)
```

```
##    stroke
##        0    1
##   0 4646  186
##   1   54   23
```

```
lda.pred.stroke <- lda.pred$posterior[, 2]
```

3.3 QDA

Assumption: sample are normally distributed BUT NOT SAME variance among classes.

```
qda.fit <- qda(stroke~age+bmi+avg_glucose_level+hypertension+heart_disease+smoking_status, data = stroke
# ERROR rank deficiency, i.e. some variables
# are collinear and one or more covariance matrices cannot be inverted to obtain the estimates in group
qda.pred <- predict(qda.fit, stroke_data)
qda.pred.stroke <- qda.pred$posterior[, 2]
table(qda.pred$class, stroke)
```

```
##    stroke
##        0    1
##   0 4260  123
##   1  440   86
```

4 Predictions

In order to split the dataset into validation, training and test sets we recall that the amount of data that we have is of 4909, so we decided to keep approximately the 75% of the data for the training phase: 3682 people of which 3562 are the ones which hadn't the stroke, while 120 had it. We didn't use the cross-validation because it was difficult to split tha data and bla bla.

CODICE CORRETTO

4.1 ROC and PRECISION-RECALL Curves

We introduced an hand-written function to make usefull plot because Allora ieri guardando i plot della ROC curve io e francesca c'eravamo posti due domande sui risultati. Siccome siamo in un problema medico di predizione di ictus di un paziente oppure no, alla fine la ROC curve non emolto d'aiuto perche massimizza i true positive con i true predicted. Questo significa che possibilmenete gli errori di false negativo possono incrementare. In ambito del nostro problema e in generale in ambito medico se il nostro modello predice una persona senza ictus quando invece lo presenta, eh questa e un errore piu grave rispetto a un false positive. Noi vorremmo invece che il false negative sia basso e quindi considerato. Insomma significa che dobbiamo usare la Precision Recall curve.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Caricamento pacchetto: 'pROC'
```

```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     cov, smooth, var
```

```
library(ROCR)
get.roc.recall.values <- function(pred_models, true_value) {
  result <- data.frame(Threshold1=double(), Specificity=double(), Sensitivity=double(),
                       Threshold2=double(), Recall=double(), Precision=double())
  n_models = length(list(mod.red.probs,lda.pred.stroke, qda.pred.stroke))
  par(mfrow=c(n_models, 2))
```

```
  for (pred in pred_models) {
    roc.res <- roc(true_value, pred, levels=c("0", "1"))
    plot(roc.res, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate",
         ylab="True positive rate")

    tmp.res  <- coords(roc.res, "best")
    pred.rec = prediction(mod.red.probs, true_value)
    perf = performance(pred.rec, "prec", "rec")
    plot(perf)
    pr_cutoffs <- data.frame(cutrecall=perf@alpha.values[[1]], recall=perf@x.values[[1]],
                             precision=perf@y.values[[1]])
    best_recall <- pr_cutoffs[which.min(pr_cutoffs$recall + pr_cutoffs$precision), ]

    result[nrow(result) + 1,] = c(tmp.res[1, 1], tmp.res[1, 2], tmp.res[1, 3],
                                  best_recall[1, 1], best_recall[1, 2], best_recall[1, 3])
  }
  par(mfrow=c(1, 1))
  return(result)
}
mod.red <- glm(stroke~age + avg_glucose_level + hypertension + bmi, data=stroke_data,
               family = binomial)
summary(mod.red)
```

```
##
## Call:
## glm(formula = stroke ~ age + avg_glucose_level + hypertension +
##     bmi, family = binomial, data = stroke_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0265  -0.2986  -0.1600  -0.0755   3.6075
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.852291   0.541579 -14.499  < 2e-16 ***
## age                0.069793   0.005593  12.479  < 2e-16 ***
## avg_glucose_level  0.004984   0.001276   3.905 9.41e-05 ***
## hypertension       0.543399   0.173304   3.136  0.00172 **
## bmi                0.002621   0.011598   0.226  0.82121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1378.4  on 4904  degrees of freedom
## AIC: 1388.4
##
## Number of Fisher Scoring iterations: 7
```

```
mod.red.probs <- predict(mod.red,type="response")
```

4.2 Training Set

Insertion of the no-stroke people into the training set:

```
no.strokes.data <- stroke_data[stroke == 0, ]
rnd.idx.no.strokes <- sample(c(1:dim(no.strokes.data)[1]))
```

Insertion of the stroke people into the training set:

```
yes.strokes.data <- stroke_data[stroke == 1, ]
rnd.idx.yes.strokes <- sample(c(1:dim(yes.strokes.data)[1]))
```

We mix together the two parts and we use the `shuffle` function because the strokes are added in the last positions

```
training.set <- no.strokes.data[rnd.idx.no.strokes[1:3562], ]
training.set <- rbind(training.set, yes.strokes.data[rnd.idx.yes.strokes[1:120], ])
shuffle <- sample(nrow(training.set))
training.set <- training.set[shuffle, ]
```

4.3 Validation Set

We mix shuffle together the remaining samples into forming the validation set.

```
val.set <- no.strokes.data[rnd.idx.no.strokes[3563:4700], ]
val.set <- rbind(val.set, yes.strokes.data[rnd.idx.yes.strokes[121:209], ])
shuffle <- sample(nrow(val.set))
val.set <- val.set[shuffle, ]
```

5 Conclusions

Si stima che la percentuale di persone che possono avere un ictus andrà via via crescendo dal momento che l'età media della popolazione è in costante crescita.