

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 designing predictive models to predict the presence of stroke disease. In addition, we highlight possible linear and non-linear relationships among the given features (predictors) and the stroke disease variable (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
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	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
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 since it did 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)
```

The variable `bmi` was not numeric because of the presence of “N/A” string values which identify missing information, hence we transformed its elements into numerical 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.

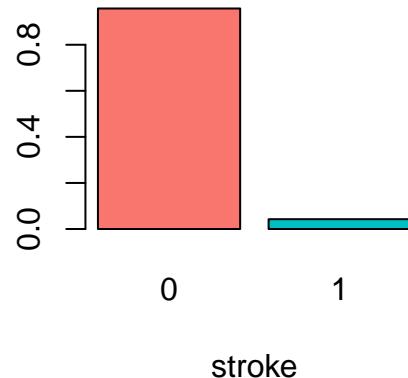
```
table(stroke)
```

```
## stroke
##   0   1
## 4700 209
```

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

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

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



These values are representative of the real situation in which there are not many stroke cases compared with the whole population, the people who have had a stroke are much less than the ones who did not have it. The incidence of stroke in Europe at the beginning of the 21st century varies from 95 to 290 cases/100,000². Furthermore in many clinical disease analyses 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' , col='#00BA38')
boxplot(bmi, xlab = 'body mass index', col='#00BA38')
boxplot(age, xlab = 'age', pch=20, col='#00BA38')
```

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

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

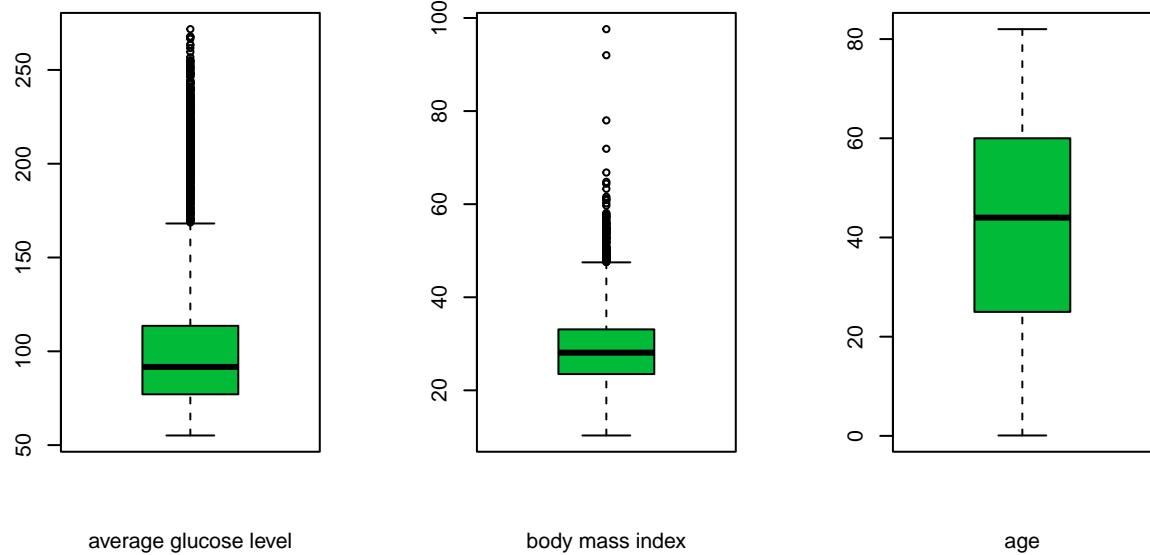


Figure 1: Visual description of some features of the dataset

From Figure 1 we can see that in the first two boxplots (starting from the left) there are lots of outliers, that can also be seen 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-case scenarios (people affected by high glucose levels or with a bmi out of parameters are only a few) 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 because it could be due to complications of diabetes, as mentioned in the medical literature.

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 functions:

```
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)
```

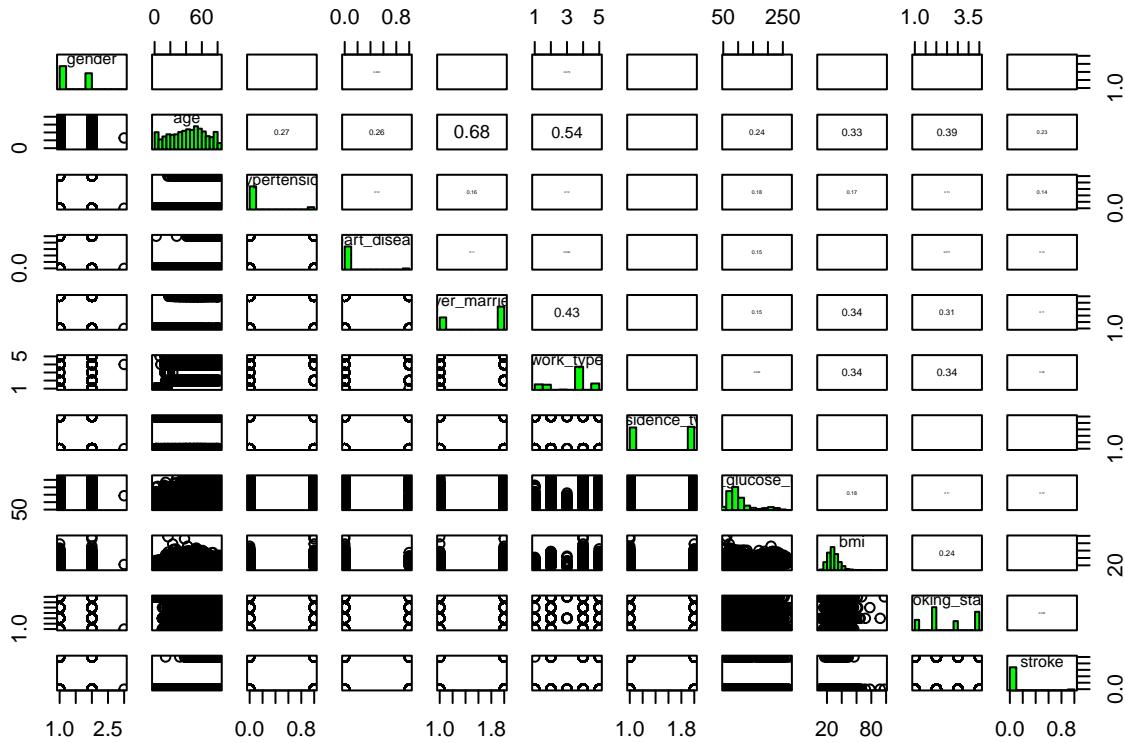


Figure 2: Pair wise plot between all the features that compose the dataset

Figure 2 shows that the stronger relationships involve quite often the variable `age`. There are also other relevant relation, for example between `work_type` and `ever_married` or `bmi` with `working_status`. In addition, we can see strong collinearity among th dummy variables `age`, `work_type`, and `ever_married`, this mean that they are closely related to one another. For this reason we get uncertainty in the coefficient estimates and so we do not consider them in the fitting of the models.

We go on looking at some intuitive relationship 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'))

```

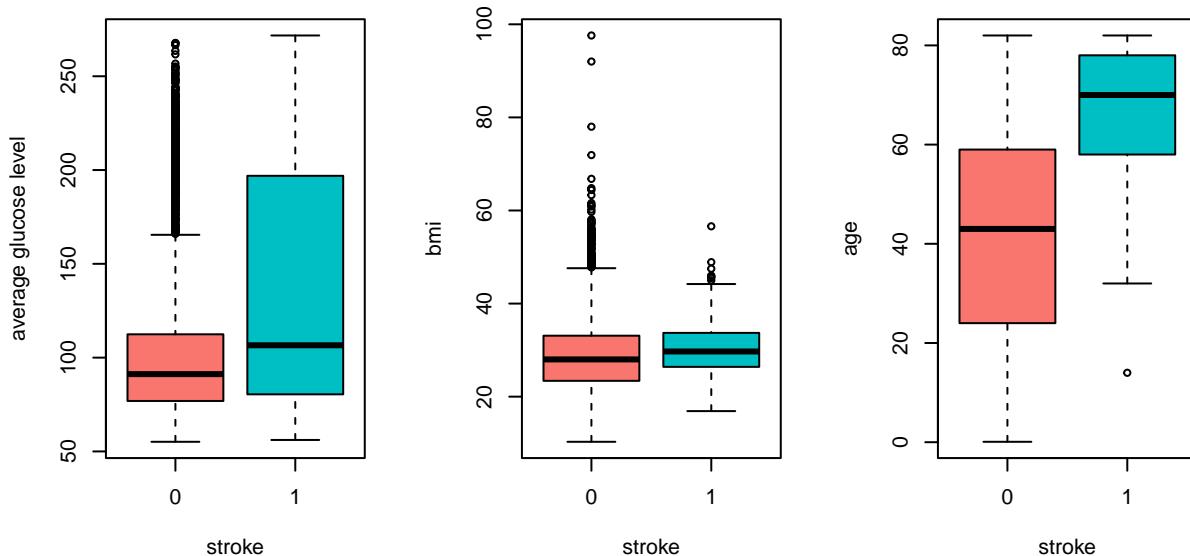


Figure 3: Visual rappresentation of the comparison between `stroke` and not-`stroke` of the features: `avg_glucose_level`, `bmi` and `age`

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

Looking at Figure 3 we can see that the incidence of the disease increases progressively with age, in particular the summary of `age` shows that it has been computed a sampling between a baby of 8 days and a senior of 82 years old and this can also be seen in the boxplot. We get also the information that the youngest person affected of stroke disease is 14 years old (an outlier), while the oldest is 82 years old.

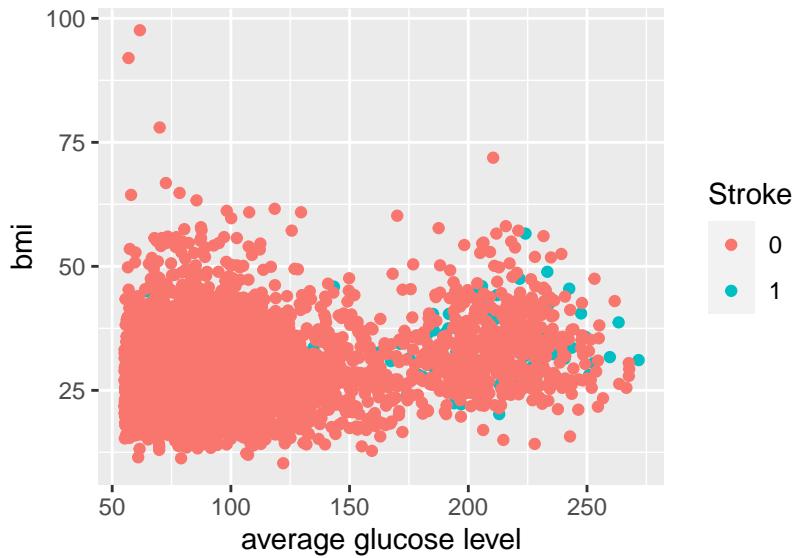
If we sum to `age` the information about `avg_glucose_level` we may wonder if diabetic people are more probable to get a stroke or not. On the other hand, there is no apparent relation of `stroke` with `bmi`.

We now highlight other visual relationship between the variables used before:

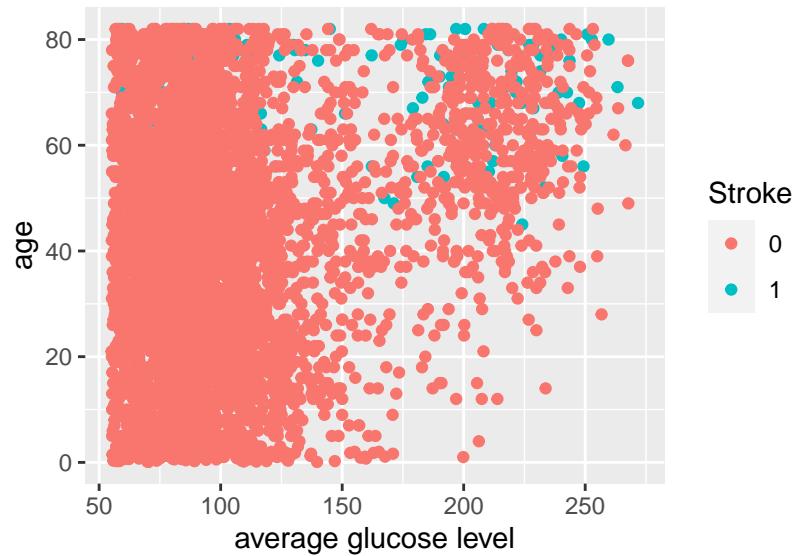
```

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()

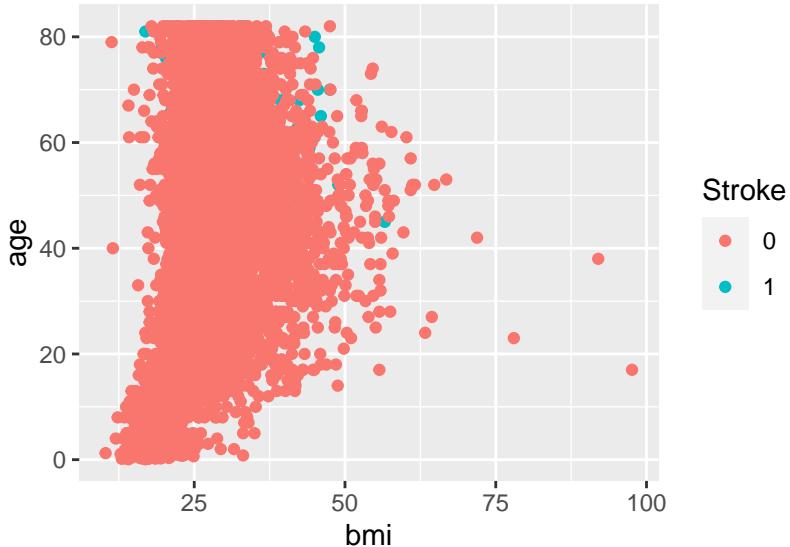
```



```
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()
```



What we can see from these scatter plots is that if we have low `avg_glucose_level` both in relation with `age` or `bmi`, the effect is that no-stroke is present. On the other hand, if we look at the `bmi` predictor we can notice that there are samples of stroke in all of its range of values, so maybe there is no clear correlation of `bmi` with stroke, even when we put this features in relation with the others.

`avg_glucose_level` and `bmi` could not be so strictly related to disease but maybe correlated to other illnesses linked (or not) to it.

In the end it seems to be not so simple to identify the direct relationship with the stroke while dealing with the features that we have, and it is difficult a further interpretation of the data.

2.3 Data Science Questions

At this point we can ask some questions:

- Which factors are the most related to the stroke disease?
- 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?
- Is it possible to prevent stroke?

We will explore the data trying to answer them.

3 Modeling

In order to explore different approaches for predicting the qualitative response `stroke`, in the following paragraph we will cover three of the most widely-used classifiers: *logistic regression*, *linear discriminant analysis (LDA)* and *quadratic discriminant analysis (QDA)*. The prediction process is known as *classification*.

##mancano questi: `exp(beta)` and odds for the prediction success

3.1 Logistic Regression

Logistic regression can be defined as a type of generalized linear model (GLM). In particular, it is a statistical model used to examine the association of categorical or continuous independent variables with one dichotomous dependent variable.

Rather than computing directly the response Y , logistic regression computes the probability that Y belongs to a particular category, in our case 1 if the person has a stroke disease, 0 otherwise.

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. We will apply the Akaike Information Criterion (AIC) as selection of the candidate models (value given directly from the function “summary”). We decide to exclude the Mallow’s Cp and the R² adjusted techniques because they work efficiently in the case of linear regression.

3.1.1 Full and Reduced Models

We start with the full model to see if all the features of the dataset contribute on the prediction of a stroke. Up to now we have already mentioned if some predictor seems to be more or less correlated to the response variable, and a way for analyze such a relation is by introducing the null hypothesis. In the case of the full model, we are dealing with 10 explanatory variables and so we have

$$H_0 : \beta_1 = \cdots = \beta_{10} = 0$$

and this means that we are making the assumption that no predictors is somehow related with `stroke`. If we reject the null hypothesis we are assuming that there is in fact a correlation and we have to find which feature is involve.

To decide whether or not reject H_0 we look at the p-values: in our case we decided to put $\alpha = 0.1$, for p-values smaller than α we keep the feature, otherwise we remove it from the model. This means that we will ended up with an accuracy of the 90 in the classification.

```
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`, with a p-value smaller than 0.005.

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

```

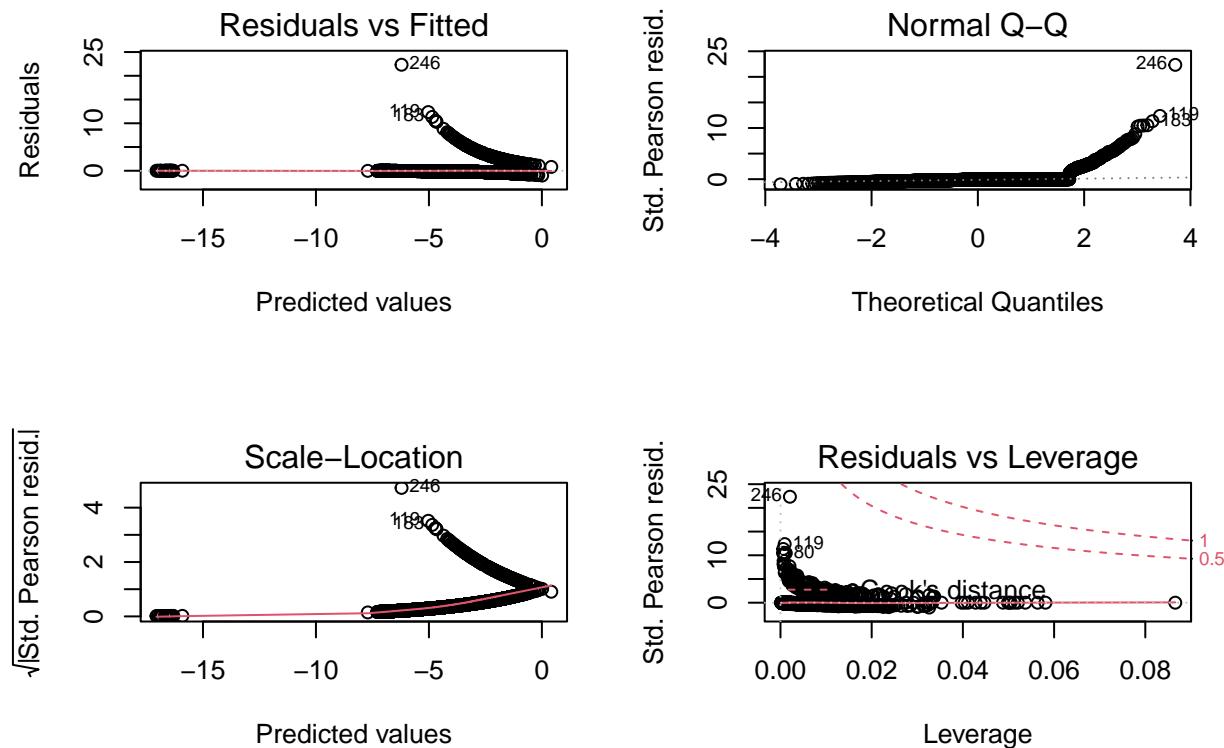
par(mfrow=c(2,2))
plot(mod.full)

```

```

## Warning: not plotting observations with leverage one:
##      2971

```



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

The residual plots are not satisfactory because it is not easy to interpret them. It is simply evident that data do not follow a linear trend.

We now make some tests in order to find the best reduced model: we take the full model and we remove all the predictors that show up collinearity between each other, i.e. `work_type`, `Residence_type` and `ever_married`, following a sort of greedy backward selection

```

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

```

```

## 
## Call:
## glm(formula = stroke ~ age + bmi + avg_glucose_level + hypertension +
##      smoking_status + gender + heart_disease, family = binomial)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q      Max
## -1.0751 -0.2957 -0.1572 -0.0734  3.6706
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -7.823946  0.588445 -13.296 < 2e-16 ***
## age                      0.069041  0.005847  11.808 < 2e-16 ***
## bmi                      0.003458  0.011745   0.294  0.768441
## avg_glucose_level        0.004697  0.001289   3.644  0.000269 ***
## hypertension              0.517649  0.174438   2.968  0.003002 **
## smoking_statusnever smoked -0.057792  0.187972  -0.307  0.758502
## smoking_statussmokes       0.321264  0.228512   1.406  0.159754
## smoking_statusUnknown     -0.256978  0.245259  -1.048  0.294740
## genderMale                -0.011195  0.154011  -0.073  0.942055
## genderOther                 -7.287812 324.743860  -0.022  0.982096
## heart_disease              0.372836  0.206072   1.809  0.070411 .
## ---
## 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: 1369.4 on 4898 degrees of freedom
## AIC: 1391.4
##
## Number of Fisher Scoring iterations: 11

```

Also in this case the variables more important are the same of the ones found in the full model, with the difference of `heart_disease` that seems to help the improvement of the prediction.

In this step we remove the `gender` variable:

```

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

```

```

## 
## Call:
## glm(formula = stroke ~ age + bmi + avg_glucose_level + hypertension +
##      smoking_status + heart_disease, family = binomial)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q      Max
## -1.0766 -0.2964 -0.1572 -0.0733  3.6720
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -7.830942  0.582317 -13.448 < 2e-16 ***
## age                      0.069065  0.005842  11.822 < 2e-16 ***
## bmi                      0.003487  0.011744   0.297  0.766535

```

```

## avg_glucose_level      0.004689   0.001285   3.648 0.000264 ***
## hypertension           0.517599   0.174427   2.967 0.003003 **
## smoking_statusnever smoked -0.055884   0.186343   -0.300 0.764254
## smoking_statussmokes    0.321811   0.228437   1.409 0.158907
## smoking_statusUnknown   -0.255822   0.244827   -1.045 0.296066
## heart_disease          0.371145   0.204739   1.813 0.069867 .
## ---
## 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: 1369.4  on 4900  degrees of freedom
## AIC: 1387.4
##
## Number of Fisher Scoring iterations: 7

```

The variables left to eliminate are `bmi` and `smoking_status` and `working_status`. Now we can define the final reduced model:

$$\text{stroke} = \beta_0 + \beta_1 \times \text{age} + \beta_2 \times \text{heart_disease} + \beta_3 \times \text{avg_glucose_level} + \beta_4 \times \text{hypertension}$$

with four explanatory variables.

```

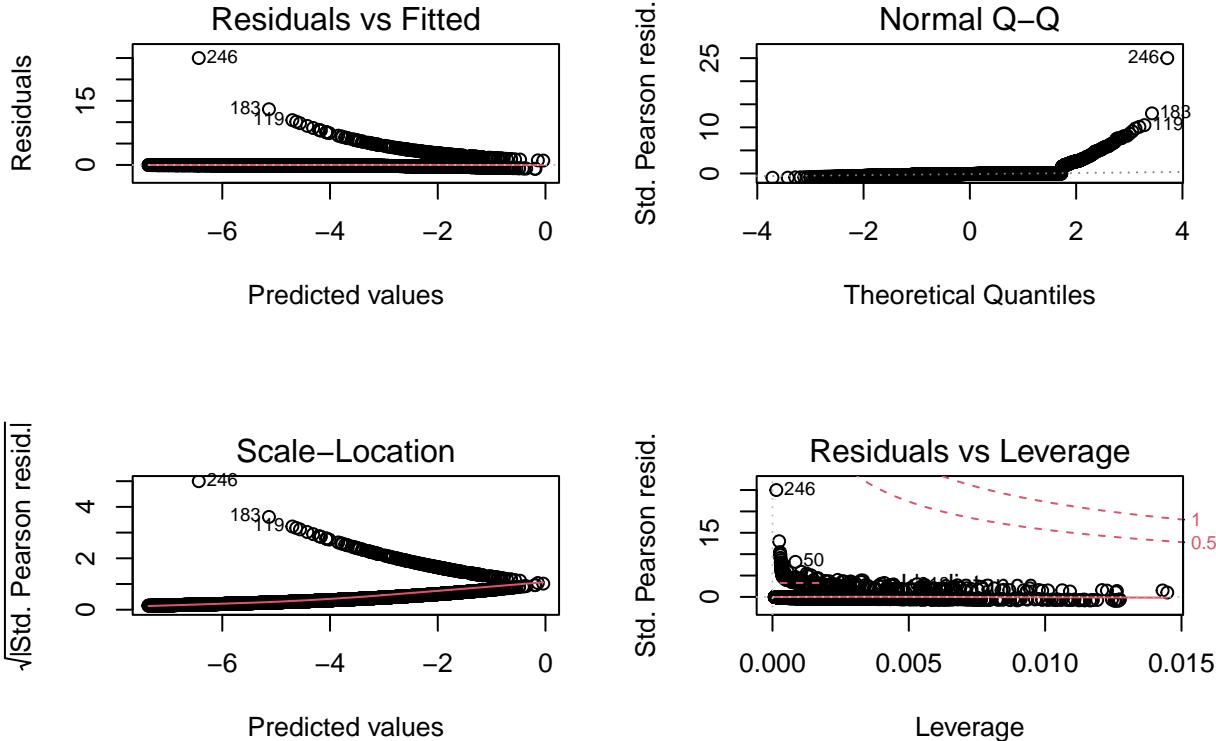
mod.red <- glm(stroke~age + heart_disease + avg_glucose_level+ hypertension,
                data=stroke_data, family = binomial)
summary(mod.red)

##
## 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:

```
par(mfrow=c(2, 2))
plot(mod.red)
```



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

Reading and interpreting these graphs, we can state some important outcomes:

Residuals vs Fitted

We can observe the presence of high non-linearity in the dataset. Indeed we can find not spread equal residuals around a horizontal line without distinct patterns, attesting to the fact we don't have a linear relationship.

Normal Q-Q

We discover that residuals do no follow a normal distribution.

We know that obtaining positive coefficient estimates, we have a positive association. So, the larger the value the higher is the estimated probability of stroke.

Scale-Location

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. But in our case, as you can notice from this plot, the red line is slightly curved and the residuals seem to increase as the fitted Y values increase. So, the inference here is, heteroscedasticity exists.

Residual vs Leverage

Looking at the leverage plot, we see the presence of a sample with high leverage values (bottom right), which could influence the prediction of the model (even if the R-plot does not show us its index). Moreover, there are obvious outliers (they have no connection with our regression) with high variance.

	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

If we compare the results, we can affirm that the reduced model seems to make a more accurate prediction and fits better data than the full one. Indeed, its AIC is the lower of the two.

To have a confirmation of this concept, we use ‘anova’, a function which compares them and returns the best:

```
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, the anova test rejects that the complex model is more significant than the reduced one, since the p-value is not less than 5%. Therefore the full model does not enhance the prediction.

3.1.2 Interaction Models

Let's try to check other various models using a mixed approach. We use the reduced model `mod.red` as landmark and we proceed with interactions between the explanatory variables with the aim to understand which variables are relevant on our research and to improve the performance of the model. We remember that we move on with our selection choosing significance levels equal to 0.1 and searching for an AIC value less and less.

The model `mod.red` has `stroke` as response and `age`, `avg_glucose_level`, `hypertension` and `heart_disease` as predictors. We recall that its AIC is 1384.6.

Initially we consider the interaction of `age` with the other numerical features, i.e. `avg_glucose_level`, `hypertension` and `heart_disease`: we realize that only `age*heart_disease` is relevant.

AIC = 1384.

```
stroke =  $\beta_0 + \beta_1 \times \text{age} + \beta_2 \times \text{heart\_disease} + \beta_3 \times \text{avg\_glucose\_level} + \beta_4 \times \text{hypertension} + \beta_5 \times \text{age} \times \text{heart\_disease}$ 

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 go 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 even if it can not improve the previous model.

It has an AIC of 1385.9.

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

Ultimately, we have the interaction `heart_disease*hypertension` which seems a good version and 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

```

After that, we pick up `mod.red` without `heart_disease` which represents the explanatory feature with higher

p-value and we continue to work in the same way have just seen. We repeat the scheme with `age`, without earning progresses.

`AIC = 1386.2`

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

We go further testing also the interaction add `avg_glucose_level*hypertension`.

`AIC = 1387.6`

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

In the end, we include the possible best interactions into the reduced model. It could be a good way, in fact it does not represent our winner due to a high p-value.

`AIC = 1384.2`

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

As conclusion, we can promote `mod1` as the model which better fits our data:

$$\text{stroke} = \beta_0 + \beta_1 \times \text{age} + \beta_2 \times \text{heart_disease} + \beta_3 \times \text{avg_glucose_level} + \beta_4 \times \text{hypertension} + \beta_5 \times \text{age} * \text{heart_disease}$$

At the end of all we promote `mod1` as the model which better fits our data. This means that our probability become:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5}}$$

where β_0, \dots, β_5 can be read in the summary of the model `mod1` and also the respectively variables X_1, \dots, X_5 which represent `age`, `avg_glucose_level` and so with the other variables of the model.

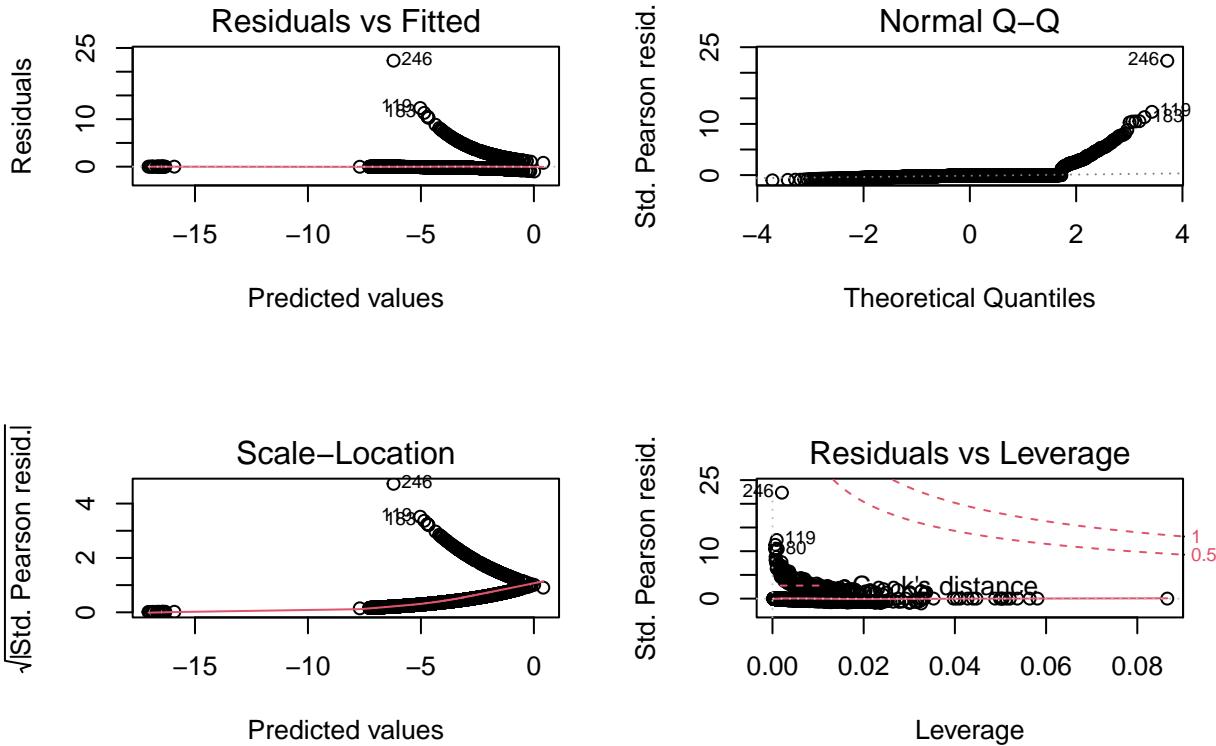
From $p(X)$ we can get the quantity $p(X)/(1 - p(X))$ which is called *odds* and can take any value between 0 and ∞ . This quantity is important because of its interpretation: an increment of one unit in the predictor X_1 cause an increment of e^{β_1} in the probability of success. Let's now see some 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

and other properties and characteristics:

```
par(mfrow=c(2,2))
plot(mod.full)
```

```
## Warning: not plotting observations with leverage one:
##      2971
```



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

3.1.3 Polynomial models

Let's try to do one more test using the polynomial model starting from the reduced model `mod.red`, adding also `bmi` as predictor. We contribute with the square of `bmi`, `avg_glucose_level` and then both of them, obtaining `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)

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

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

However, we do not achieve anything interesting with polynomial model. There are no improvements in the results, seeing the value of the AIC which is extremely high.

3.2 Bayesian models

In this section we now study other type of predictive models, which are based on Bayesian concepts. These methods exploits conditional probability and Bayes theorems to make prediction, they are also widely used in classification problems because of their nice structure and properties . Furthermore we would like to compare their results with the logistic model in order to assess their performances. ## 3.2.1 LDA We first consider the Linear Discriminant Analysis (LDA) for the Multivariate case, since we are dealing with multiple

explanatory variables Assumption: samples are normally distributed and have same variance in every class
 \Rightarrow strong assumption.

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

##      stroke
##          0    1
##  0 4653 194
##  1   47   15

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

3.2.2 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_data)
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 Best Model Selection & Validation Test

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 the 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 possibilmente 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, yes_plot=TRUE) {
  result <- data.frame(Thr.ROC=double(), Specificity=double(), Sensitivity=double(),
                        Thr.Prec.Rec=double(), Recall=double(), Precision=double())
  n_models = length(pred_models)
  par(mfrow=c(n_models, 2))
  for (pred in pred_models) {
    ### ROC
    roc.res <- roc(true_value, pred, levels=c("0", "1"))
    if (yes_plot==TRUE){
      plot(roc.res, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate",
            ylab="True positive rate")
    }
    best.roc <- coords(roc.res, "best")
    ###

    ### PREC-REC
    pred.rec = prediction(pred, true_value)
    perf = performance(pred.rec, "prec", "rec")
    if (yes_plot==TRUE){
      plot(perf)
    }
    pr_cutoffs <- data.frame(cutrecall=perf@alpha.values[[1]], recall=perf@x.values[[1]],
                             precision=perf@y.values[[1]])
    pr_cutoffs <- pr_cutoffs[pr_cutoffs$recall>0 & pr_cutoffs$precision >0, ]
    best_recall <- pr_cutoffs[which.min(pr_cutoffs$recall + pr_cutoffs$precision), ]

    result[nrow(result) + 1,] = c(best.roc[1, 1], best.roc[1, 2], best.roc[1, 3],
                                    best_recall[1, 1], best_recall[1, 2], best_recall[1, 3])
  }
  par(mfrow=c(1, 1))
  return(result)
}

```

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, ]

```

Now we evaluate the training models.

```
attach(training.set)
```

```

## I seguenti oggetti sono mascherati da stroke_data:
##
##      age, avg_glucose_level, bmi, ever_married, gender, heart_disease,
##      hypertension, Residence_type, smoking_status, stroke, work_type

```

Reduced Model:

```

# mod.red.train
mod.red.train <- glm(stroke ~ age + heart_disease + avg_glucose_level + hypertension,
                      data=training.set, family = binomial)
mod.red.train.pred <- predict(mod.red.train, data=training.set, type="response")

# mod1.train
mod1.train <- glm(stroke ~ age + avg_glucose_level+ heart_disease + hypertension +
                    age*heart_disease, data=training.set, family=binomial)
mod1.train.pred <- predict(mod1.train, data=training.set, type="response")

# lda.fit.train
lda.fit.train <- lda(stroke ~ age + bmi + avg_glucose_level + hypertension +
                      smoking_status + Residence_type + heart_disease,
                      data=training.set)
lda.fit.train.pred <- predict(lda.fit.train, data=training.set)
lda.fit.train.pred <- lda.fit.train$posterior[, 2]

# qda.fit.train
qda.fit.train <- qda(stroke ~ age + bmi + avg_glucose_level + hypertension +
                      heart_disease + smoking_status, data = training.set)
qda.fit.train.pred <- predict(qda.fit.train, data=training.set)
qda.fit.train.pred <- qda.fit.train$posterior[, 2]

# qda.inter.fit.train
qda.inter.fit.train <- qda(stroke ~ age + avg_glucose_level+ heart_disease +
                            hypertension + age*heart_disease, data = training.set)
qda.inter.fit.train.pred <- predict(qda.inter.fit.train, data=training.set)
qda.inter.fit.train.pred <- qda.inter.fit.train$posterior[, 2]

```

The interaction qda, which has the same variable of the logistic interaction model, improved a lot the result in the training set, but in the validation test it does perform the same as the previous qda.

We apply this

```

res = get.roc.recall.values(list(mod.red.train.pred, mod1.train.pred,
                                 lda.fit.train.pred, qda.fit.train.pred,
                                 qda.inter.fit.train.pred), training.set$stroke,
                                 yes_plot=FALSE)
print(res)

##      Thr.ROC Specificity Sensitivity Thr.Prec.Rec      Recall Precision
## 1 0.04171433  0.7902864  0.7833333  0.2936870 0.016666667 0.1666667
## 2 0.04204012  0.7838293  0.7833333  0.2227987 0.033333333 0.1052632

```

```

## 3 0.02710058 0.7650197 0.8000000 0.7321485 0.016666667 0.1333333
## 4 0.03103868 0.7768108 0.8166667 0.9993139 0.066666667 0.2051282
## 5 0.01451399 0.7490174 0.8166667 1.0000000 0.008333333 0.1666667

recall_thresholds = res$Thr.Prec.Rec
roc_thresholds = res$Thr.ROC

Nice:

mod.red.train.pred.class <- as.numeric(mod.red.train.pred >= roc_thresholds[1])
table(mod.red.train.pred.class, training.set$stroke)

##
## mod.red.train.pred.class    0     1
##                      0 2815   26
##                      1 747   94

mod.red.train.pred.class <- as.numeric(mod.red.train.pred >= recall_thresholds[1])
table(mod.red.train.pred.class, training.set$stroke, dnn=c('ground thruth','pred'))

##
##          pred
## ground thruth 0     1
##                 0 3552  118
##                 1  10   2

mod1.train.pred.class <- as.numeric(mod1.train.pred >= recall_thresholds[2])
table(mod1.train.pred.class, training.set$stroke)

##
## mod1.train.pred.class    0     1
##                      0 3528   116
##                      1  34    4

lda.fit.train.pred.class <- as.numeric(lda.fit.train.pred>=recall_thresholds[3])
table(lda.fit.train.pred.class, training.set$stroke)

##
## lda.fit.train.pred.class    0     1
##                      0 3549   118
##                      1  13    2

qda.fit.train.pred.class <- as.numeric(qda.fit.train.pred>= recall_thresholds[4])
table(qda.fit.train.pred.class, training.set$stroke)

##
## qda.fit.train.pred.class    0     1
##                      0 3531   112
##                      1  31    8

qda.inter.fit.train.pred.class <- as.numeric(qda.inter.fit.train.pred >= recall_thresholds[5])
table(qda.inter.fit.train.pred.class, training.set$stroke)

##
## qda.inter.fit.train.pred.class    0     1
##                      0 3557   119
##                      1  5     1

```

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, ]
```

Models on validation

```
mod.red.val <- predict(mod.red.train, val.set, type="response")
mod.red.val.class <- as.numeric(mod.red.val >= recall_thresholds[1])
table(mod.red.val.class, val.set$stroke)
```

```
##
## mod.red.val.class      0      1
##                  0 1136    87
##                  1      2      2
mod1.val <- predict(mod1.train, val.set, type="response")
mod1.val.class <- as.numeric(mod1.val >= recall_thresholds[2])
table(mod1.val.class, val.set$stroke)
```

```
##
## mod1.val.class      0      1
##                  0 1131    81
##                  1      7      8
lda.val <- predict(lda.fit.train, val.set)
lda.val <- lda.val$posterior[, 2]
lda.val.class = as.numeric(lda.val >= recall_thresholds[3])
table(lda.val.class, val.set$stroke)
```

```
##
## lda.val.class      0      1
##                  0 1138    86
##                  1      0      3
qda.val <- predict(qda.fit.train, val.set)
qda.val <- qda.val$posterior[, 2]
qda.val.class <- as.numeric(qda.val >= recall_thresholds[4])
table(qda.val.class, val.set$stroke)
```

```
##
## qda.val.class      0      1
##                  0 1129    84
##                  1      9      5
qda.inter.val <- predict(qda.inter.fit.train, val.set)
qda.inter.val <- qda.inter.val$posterior[, 2]
qda.inter.val.class <- as.numeric(qda.inter.val >= recall_thresholds[5])
table(qda.inter.val.class, val.set$stroke)
```

```
##
## qda.inter.val.class      0      1
##                  0 1138    89
```

The interaction qda, which has the same variable of the logistic interaction model, improved a lot the result in the training set, but in the validation test it does perform the same as the previous qda

5 Conclusions and Further Analysis

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.

Risposta alle domande ed eventualmente riferimenti ai plot.

Consigli di aggiungere features