Machine learning predictive models to classify Pima Indians Diabetes dataset

Jordi Toneu

15/5/2020

This project is the second exercice of the HarvardX: PH125.9x Data Science Capstone project course. Each student has to freely choose a dataset, define the target of the project, apply different machine learning techniques and properly communicate the process and insights gained from the dataset analysis.

Dedicated to my loved children Oriol and Maria, which well deserved more attention from my side while I was passionately inmmersed with this Harvard Professional Certificate Program in Data Science during the 2020 lockdown in The Netherlands.

Index

1.1 Overview

Diabetes is a chronic disease that causes high sugar level in blood. Over the time, it can damage organs like heart and kidneys. Some consequences are blindness, heart attacks, stroke, kidney failure or partial amputation.

We will consider different approaches to develop a supervised machine learning algorithm to classify the risk to be diabetic, given independent features.

1.2 The target

The target is to evaluate four different predictive models performance to predict the risk to be diabetic. The techniques we will use to build the four models are: logistic regression, KNN, CART and Random Forest.

We will use the metric **overall accuracy** to assess models performance.

1.3 The dataset

The **Pima Indians Diabetes dataset** was provided by the National Institute of Diabetes and Digestive and Kidney Diseases. The dataset can be found on the Keggel website. It is a csv file. It can also be downloaded from **https://github.com/jorditoneu/Diabetes_Project/blob/master/diabetes.csv** or automatically download it by using the following code in **R**.

```r
###########################
# Downloading the dataset #
###########################

# Installing required packages
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(ggpubr)) install.packages("ggpubr", repos = "http://cran.us.r-project.org")

# Diabetes dataset source
urlfile="https://raw.githubusercontent.com/jorditoneu/Diabetes_Project/master/diabetes.csv"

# Reading the Pima Indians Diabetes csv file
diabetes<-read_csv(url(urlfile))
```

2.1 Data exploration and cleaning

The dataset contains 9 variables, 8 predictors and 1 categorical outcome with 2 classes, and 768 observations.

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, ...
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, ...
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92,...
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, ...
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, ...
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, ...
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, ...
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30...
## $ Outcome                  <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, ...
```

We can have a look at the 10 first rows of the dataset.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31.0	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0.0	0.232	54	1

This is the meaning of each variable.

Variable	Description
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction	Diabetes pedigree function
Age	Age (years)
Outcome	class variable. 0 indicates non-diabetes, 1 indicates diabetes

The variable **Pregnancies** denotes that the dataset contains women information only.

The R **summary()** function provides some extra information of all the variables.

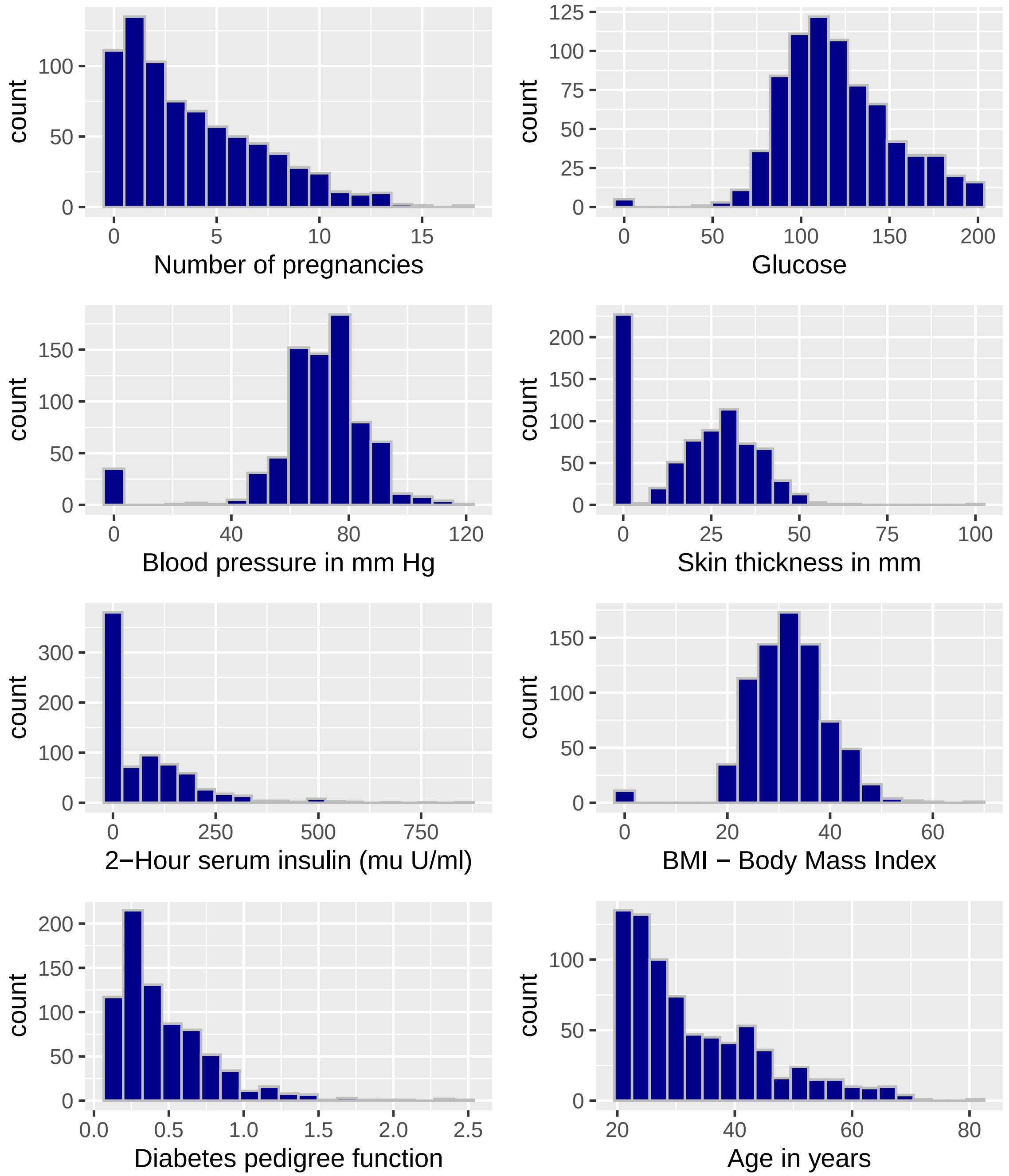
```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

We check if there is any **NA** (Not Available, No Answer) value in the dataset.

```
## [1] 0
```

No NA's found in the dataset.

Predictors histograms



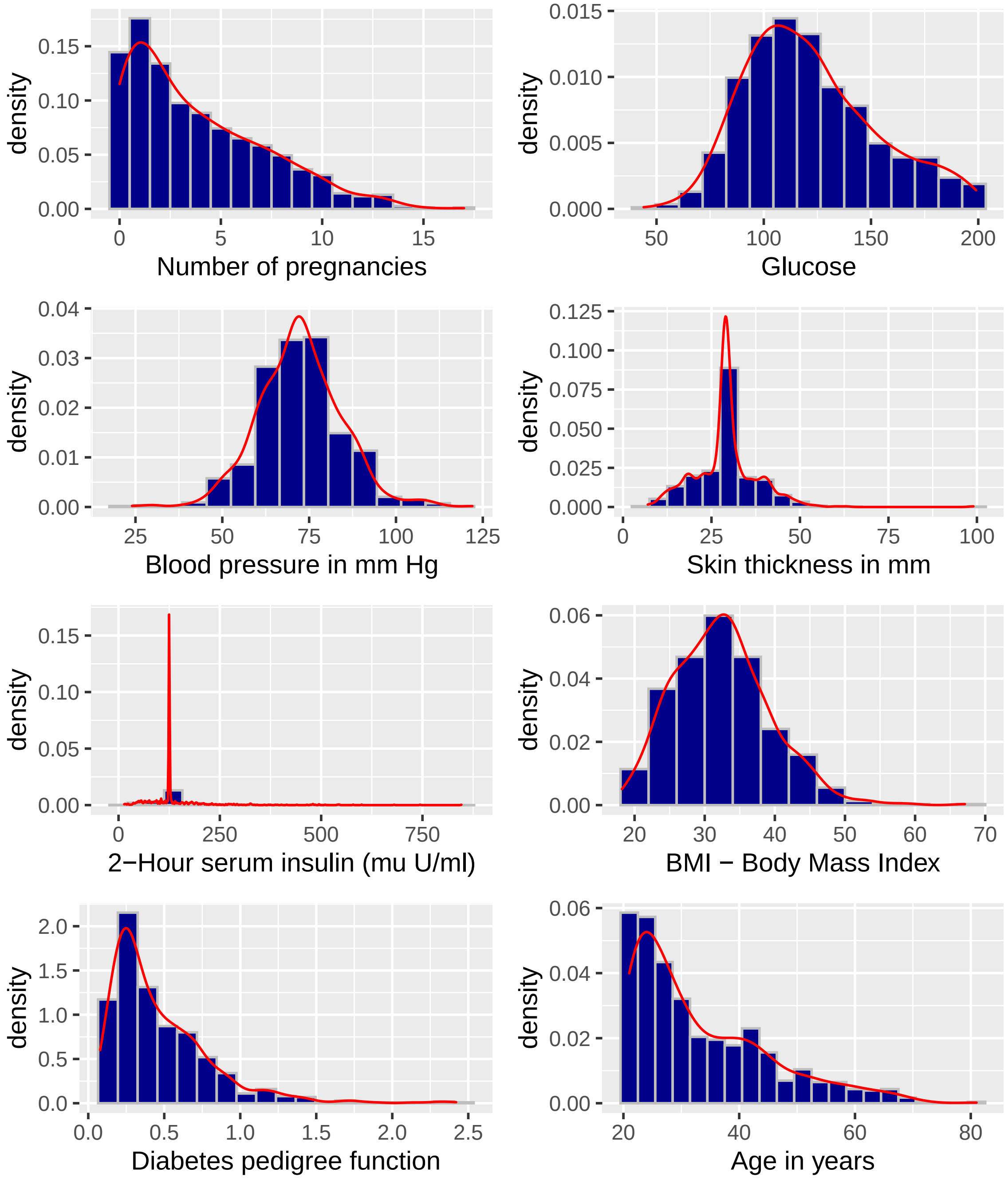
We found no NA in the dataset, however, we observe in the previous histograms that some predictors contain a big number of 0. These values are not outliers but incorrect observations. As an example, it is not possible to have 0 mm of skin thickness. The predictors with such problems are **Glucose**, **BloodPressure**, **SkinThickness**, **Insulin** and **BMI**.

The proportion of rows in the dataset with 0's is

```
## [1] 0.5625
```

In the event we delete the rows with 0's, the number of observations will be reduced by 44% and that will make more difficult to predict an accurate outcome. In order to avoid this problem, we will replace the 0's for the median value of each feature.

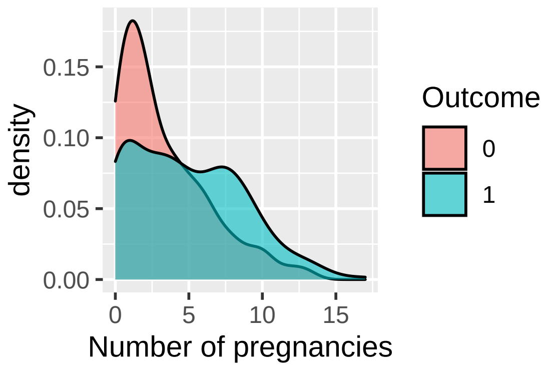
Predictors histograms and density plot without 0

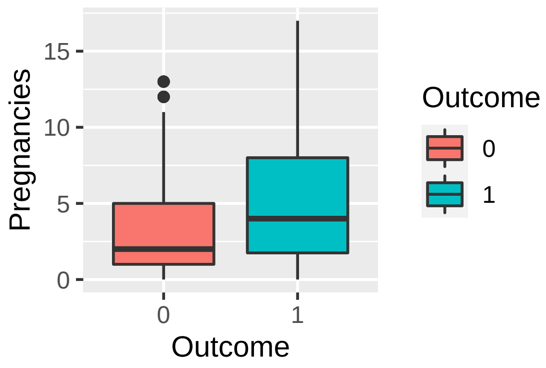


We can see some predictors have outliers. The most significant cases are **Insulin**, **Skin thickness**, **Diabetes pedigree function** and **BMI**.

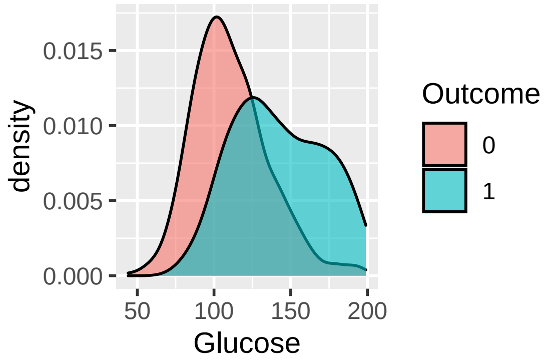
None of the predictors follow a normal distribution, except **Blood pressure**, but this case needs to be confirmed with further analysis.

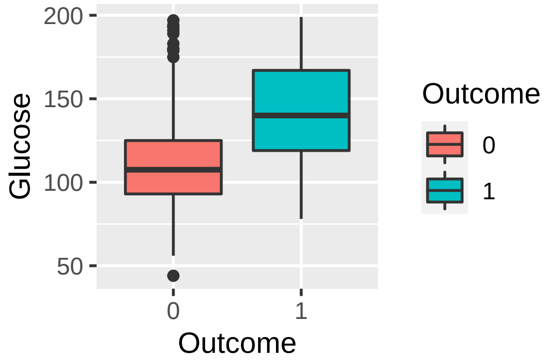
Predictors stratified by outcome



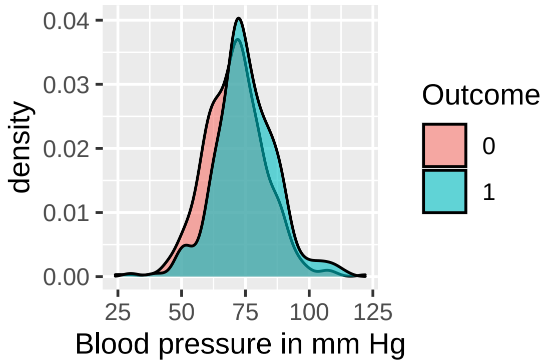


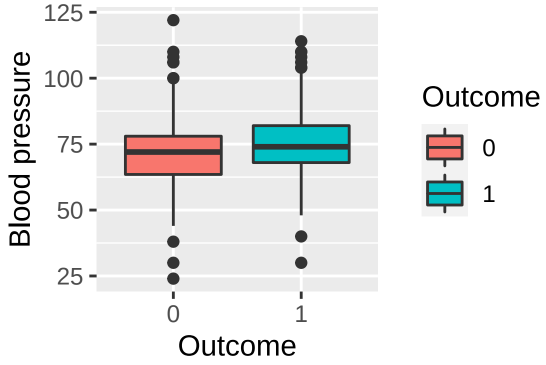
We can observe that diabetic women have a higher number of pregnancies.



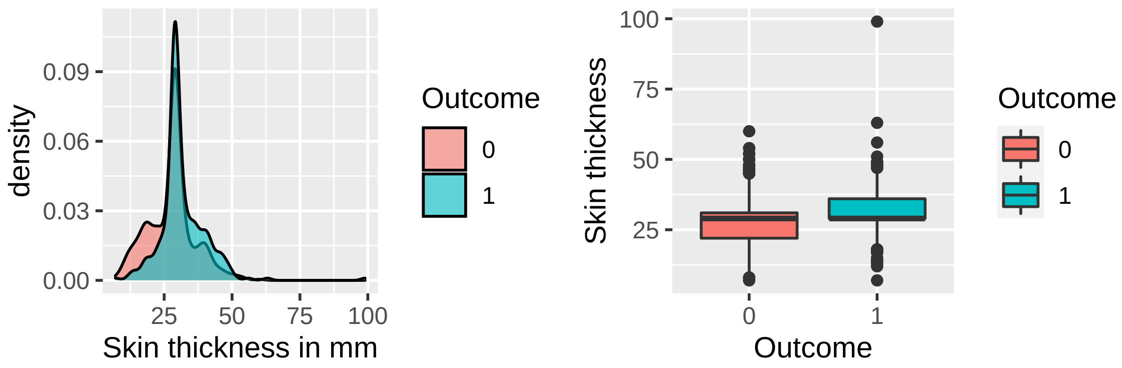


In this case we can pretty well distinguish non diabetic and diabetic women by the level of glucose.

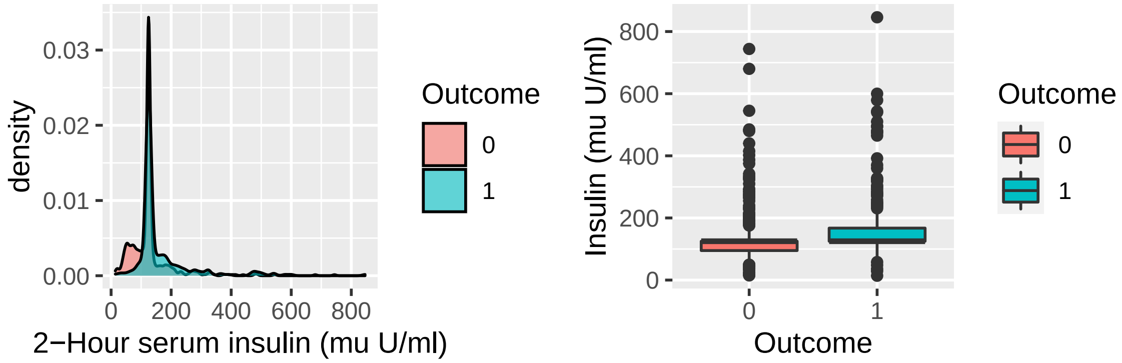




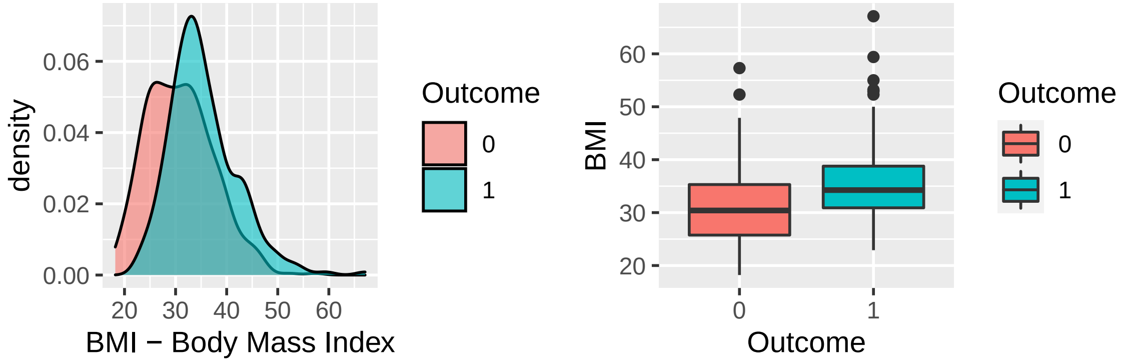
No significant difference can be observed in blood pressure level of non diabetic and diabetic women.



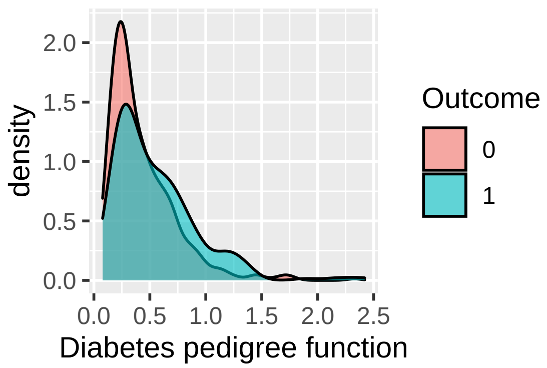
Diabetic women show a slightly higher skin thickness level, but the difference is minimal.

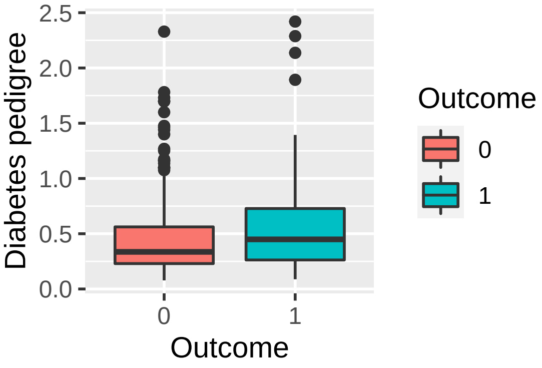


We can observe the same insulin level of diabetic and non diabetic women but in case of diabetic ones, the insulin level is more stable.

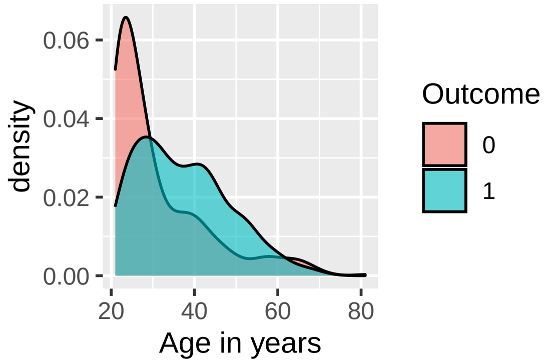


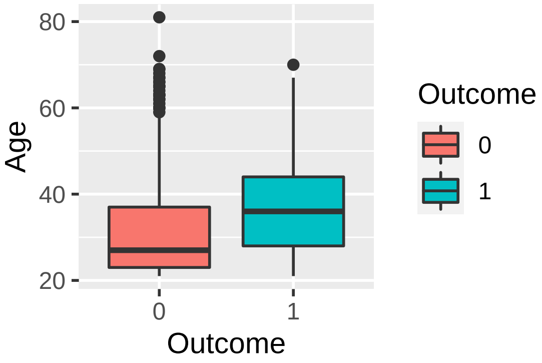
Diabetic women have a higher BMI but we cannot clearly distingush if to be or not to be diabetic only with this predictor.





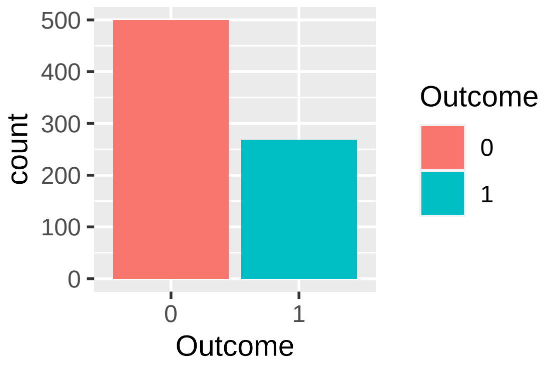
We can observe that diabetic women have a slightly higher Diabetis Pedigree Function.





Diabetic women seems to be older but this predictor doesn't clearly split diabetic and non diabetic women.

The following graph shows the proportion of positive and negative classes, where 0 means non-diabetic and 1 means diabetic case.



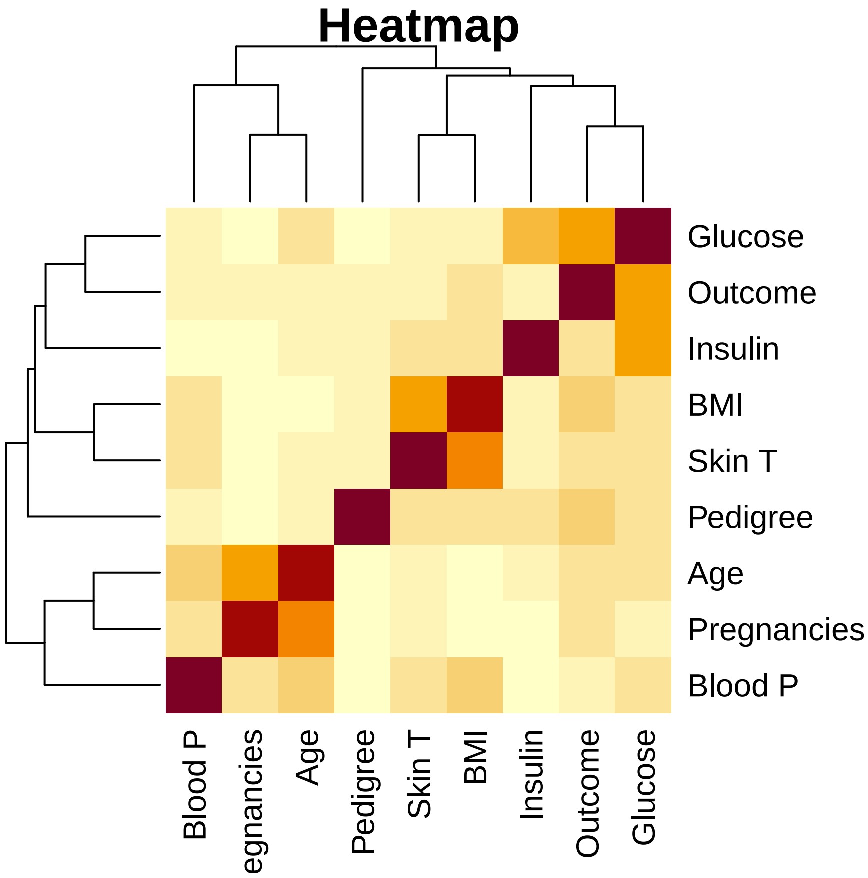
We can clearly see that the dataset **prevalence** is the **positive class 0**, that means more cases without diabetes.

Correlations

We will now have a look at the correlation of each pair of features. This table shows the correlation values.

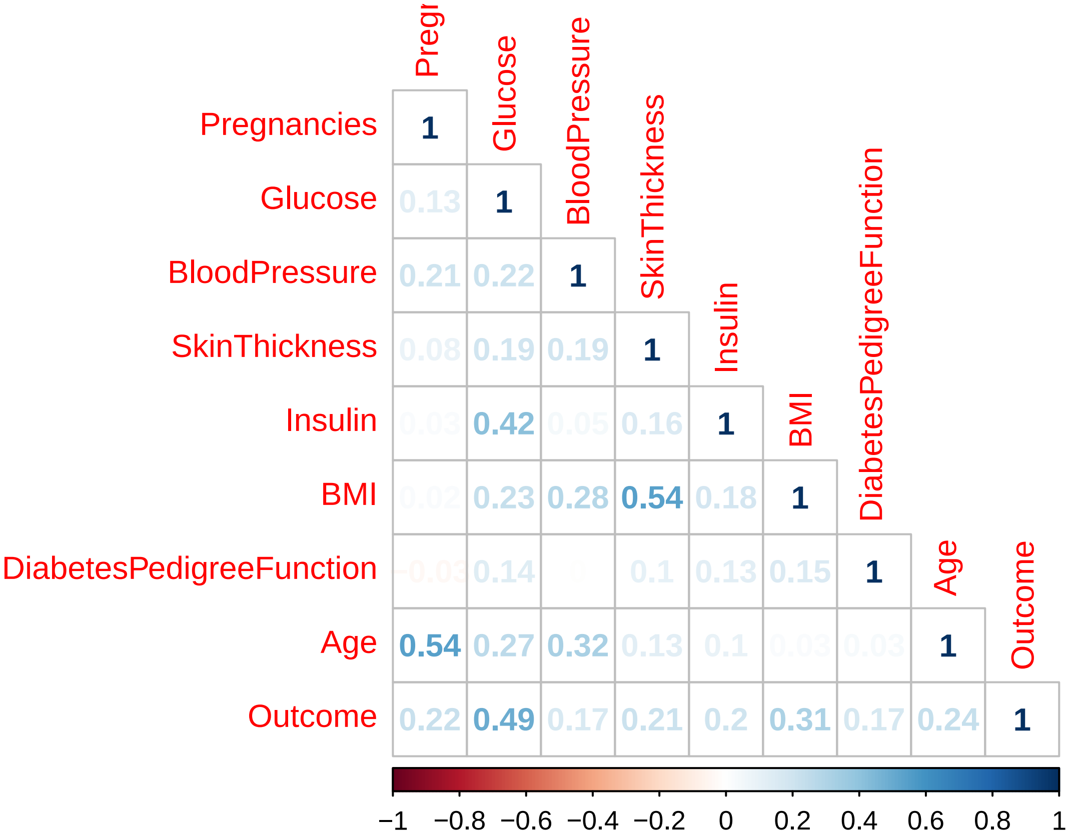
	Pregnancies	Glucose	Blood P	Skin T	Insulin	BMI	Pedigree	Age	Outcome
Pregnancies	1.0000	0.1282	0.2086	0.0818	0.0250	0.0215	-0.0335	0.5443	0.2219
Glucose	0.1282	1.0000	0.2189	0.1926	0.4195	0.2314	0.1373	0.2669	0.4928
Blood P	0.2086	0.2189	1.0000	0.1919	0.0454	0.2811	-0.0024	0.3249	0.1657
Skin T	0.0818	0.1926	0.1919	1.0000	0.1556	0.5433	0.1022	0.1261	0.2149
Insulin	0.0250	0.4195	0.0454	0.1556	1.0000	0.1804	0.1265	0.0971	0.2038
BMI	0.0215	0.2314	0.2811	0.5433	0.1804	1.0000	0.1535	0.0257	0.3122
Pedigree	-0.0335	0.1373	-0.0024	0.1022	0.1265	0.1535	1.0000	0.0336	0.1738
Age	0.5443	0.2669	0.3249	0.1261	0.0971	0.0257	0.0336	1.0000	0.2384
Outcome	0.2219	0.4928	0.1657	0.2149	0.2038	0.3122	0.1738	0.2384	1.0000

This heatmap shows the predictors correlations but with a visual format.



Dark colors indicate high correlation and light colors the opposite.

A correlogram is a different way to see the correlation between two variables and maybe even easier to follow up.



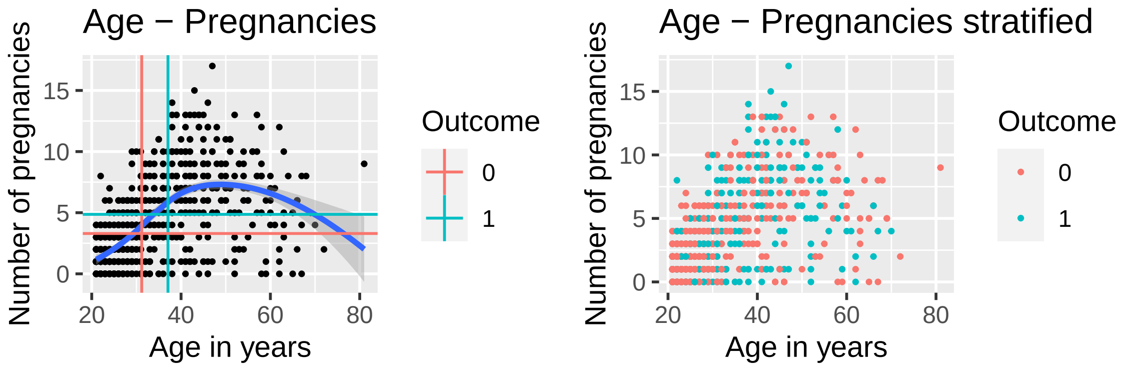
The features with higher correlation are:

Age - Pregnancies
BMI - SkinThickenss
Outcome - Glucose

We see that **Glucose** is the variable with highest correlation with **Outcome**, then goes **BMI**, **Age** and **Pregnancies**.

```
##     Glucose         BMI         Age Pregnancies
##   0.4927824   0.3122490   0.2383560   0.2218982
```

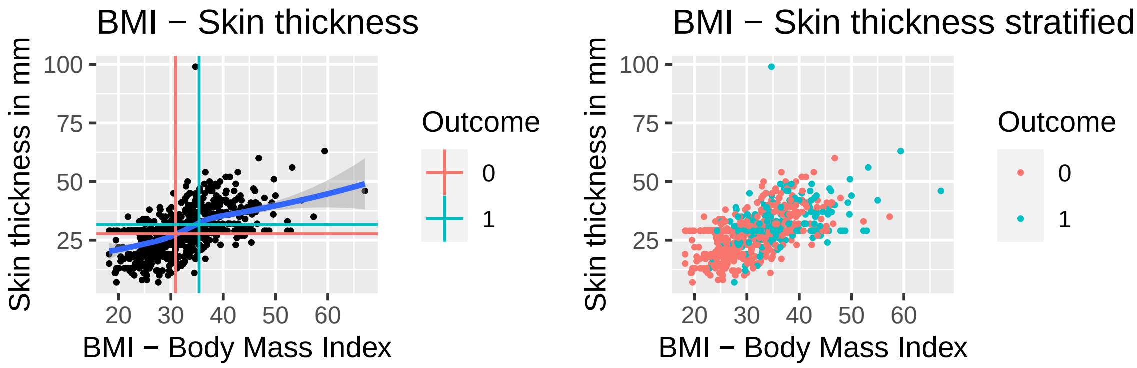
Let's have a look at the relationship of the top 3 pair of highest correlated predictors.



The blue line shows that the number of pregnancies increase with age. One could also expect that after fertility period, the number of pregnancies keep flat. However, we see that this value goes down and at the same time we have less data in the scatter plot with an increase of confidence interval. Several reasons could be behind this number of pregnancies drop, like lower natality rates because of any crisis, women with a higher number of pregnancies has a lower life expectancy or sample data too small and by chance these values are low.

Red and green lines indicate the average value of each predictor stratified by outcome. The average age of diabetic people is higher as well as the average number of pregnancies.

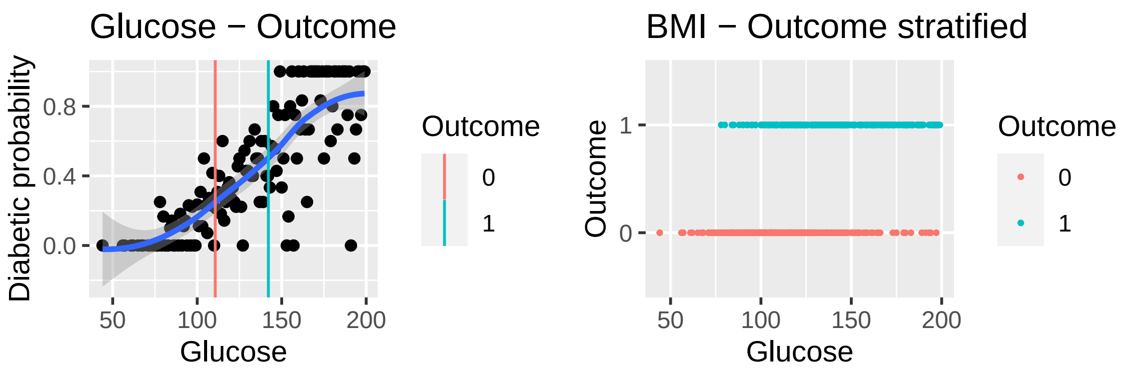
No clear boundary can be drawn that separates non diabetic and diabetic women based on number of Pregnancies vs Age.



This graph shows that the higher the BMI is, the thicker the skin is.

We can also observe that diabetic people, as average, roughly have a BMI 20% higher than non diabetics.

Non diabetic women have lower BMI and skin thickness.



Outcome	AvgGlucose
0	110.6820
1	142.1306

The blue line shows the probability to be diabetic as per the glucose level.

Diabetics average glucose level is 30% higher than non diabetics.

2.2 Data normalization

Our dataset contains features with different units and dimensions. Some of the algorithms, not all, are based on predictors distance. We do not want our models to be affected by the magnitude of these variables. The algorithm should not be biased towards variables with higher magnitude. To overcome this problem, we will normalize the variables.

We will calculate the mean and standard deviation of the variable, and then, for each observation we will substract the mean and then divide by the standard deviation of that variable.

$$z_i = \frac{x_i - \overline{x}}{\sigma}$$

2.3 Data splitting

We will split the dataset in **train**, **test** and **validation** set to build different machine learning algorithms.

The **train set** is the sample of data to train the model. The **test set** is the data used for fine-tuning some models. The **validation set** simulates a set of theoretical real values which is only used once to see how the algorithm performs with unkknown data.

Our dataset is not large, but we only have to predict a two class categorical outcome. Deciding the partition proportion of each set is a compromise. A large training set should let us build a more accurate model but there will be less data left for cross-validation and that will not help accuracy because we will end up with with a train and test set with different data distributions. A model trained on a vastly different data distribution than the test set will perform inferiorly with the validation set.

We will create the **validation_set** as 10% of the total diabetes dataset.

The **train_set** set will be 80% and the **test_set** set will be 20% of the 90% of the diabetes dataset.

The **validation_set** has 77 observations.

```
## tibble [77 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Pregnancies             : num [1:77] -0.844 2.717 -0.251 0.046 0.936 ...
##  $ Glucose                 : num [1:77] -0.613 0.767 1.194 -0.35 2.147 ...
##  $ BloodPressure           : num [1:77] -3.504 0.795 0.299 -0.032 -0.363 ...
##  $ SkinThickness           : num [1:77] 1.011 -1.15 0.784 2.035 1.125 ...
##  $ Insulin                 : num [1:77] -0.668 -0.355 1.208 0.768 1.891 ...
##  $ BMI                     : num [1:77] 1.578 -1.491 -0.124 0.676 0.763 ...
##  $ DiabetesPedigreeFunction: num [1:77] -0.872 -0.685 1.144 2.771 -0.658 ...
##  $ Age                     : num [1:77] -0.0205 2.0203 -0.4456 1.9353 0.6598 ...
##  $ Outcome                 : Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 1 1 1 ...
```

The **train_set** has 552 observations.

```
## tibble [552 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Pregnancies             : num [1:552] 0.64 -0.844 1.233 -0.844 0.343 ...
##  $ Glucose                 : num [1:552] 0.865 -1.204 2.015 -1.073 -0.186 ...
##  $ BloodPressure           : num [1:552] -0.032 -0.528 -0.693 -0.528 0.133 ...
##  $ SkinThickness           : num [1:552] 0.6702 -0.0123 -0.0123 -0.6948 -0.0123 ...
##  $ Insulin                 : num [1:552] -0.181 -0.181 -0.181 -0.54 -0.181 ...
##  $ BMI                     : num [1:552] 0.167 -0.851 -1.331 -0.633 -0.996 ...
##  $ DiabetesPedigreeFunction: num [1:552] 0.468 -0.365 0.604 -0.92 -0.818 ...
##  $ Age                     : num [1:552] 1.425 -0.191 -0.106 -1.041 -0.276 ...
##  $ Outcome                 : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 2 1 ...
```

The **test_set** has 139 observations.

```
## tibble [139 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Pregnancies             : num [1:139] -1.141 1.827 -0.844 1.53 2.123 ...
##  $ Glucose                 : num [1:139] 0.5041 0.5698 2.2125 -0.0873 0.7012 ...
##  $ BloodPressure           : num [1:139] -2.677 0.629 -1.024 0.629 1.787 ...
##  $ SkinThickness           : num [1:139] 0.6702 -0.0123 -0.6948 0.6702 0.4427 ...
##  $ Insulin                 : num [1:139] 0.3164 -0.1814 8.1651 -0.1814 0.0617 ...
##  $ BMI                     : num [1:139] 1.549 -0.778 -0.342 -0.502 0.603 ...
##  $ DiabetesPedigreeFunction: num [1:139] 5.481 2.925 -0.223 -0.63 -0.658 ...
##  $ Age                     : num [1:139] -0.0205 2.0203 2.1904 -0.3606 1.5101 ...
##  $ Outcome                 : Factor w/ 2 levels "0","1": 2 1 2 2 2 1 2 1 1 1 ...
```

All together we have 768 observations, the same number as the original diabetes dataset.

2.4 Modelling approaches

There are several common techniques to develop machine learning algorithms. In our case we will compare the performance of four different models.

2.4.1 Logistic regression model

The **logistic regression** method is a classification algorithm to assaign observations to a discrete set of classes. The logistic regression can help us to predict the risk to be diabetic as the predictions are discrete.

This is the overall accuracy result obtained after predicting the outcome with the logistic regression model fitting.

Method	Accuracy
Logistic regression model	0.7625899

The predictor's p-value indicate the meaning variables for the model. A p-value lower than 0.05 indicates that changes in the predictor's value are related to changes in the response variable.

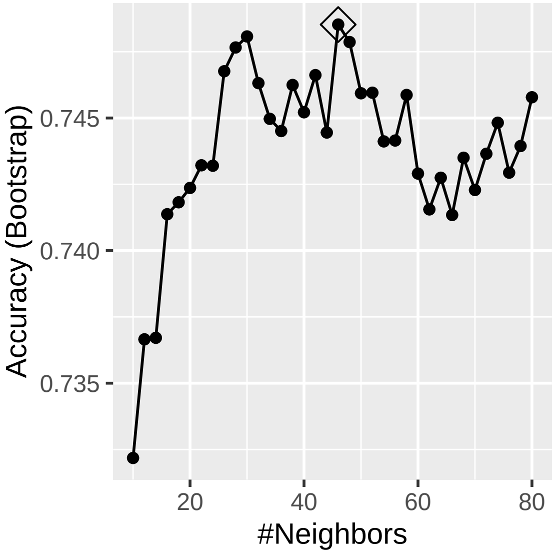
```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6769  -0.7039  -0.3855   0.6704   2.3349
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.8421     0.1149  -7.327 2.35e-13 ***
## Pregnancies                0.4574     0.1332   3.434 0.000594 ***
## Glucose                    1.1105     0.1433   7.748 9.34e-15 ***
## BloodPressure             -0.1335     0.1225  -1.089 0.276068
## SkinThickness              0.1186     0.1375   0.862 0.388590
## Insulin                   -0.1768     0.1272  -1.390 0.164583
## BMI                        0.6563     0.1519   4.320 1.56e-05 ***
## DiabetesPedigreeFunction   0.3636     0.1208   3.009 0.002621 **
## Age                        0.1918     0.1398   1.372 0.170023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 713.28  on 551  degrees of freedom
## Residual deviance: 504.36  on 543  degrees of freedom
## AIC: 522.36
##
## Number of Fisher Scoring iterations: 5
```

2.4.2 KNN model

The **KNN** (K-nearest neighbors) model is an algorithm used in machine learning that works with predictors distances.

The **K** parameter indicates the number of nearest neighbors the model considers. This parameter has to be optimized with cross-validation to obtain the highest possible accuracy.

In the next graph we can see that the best K value that generates the highest accuracy is 46.



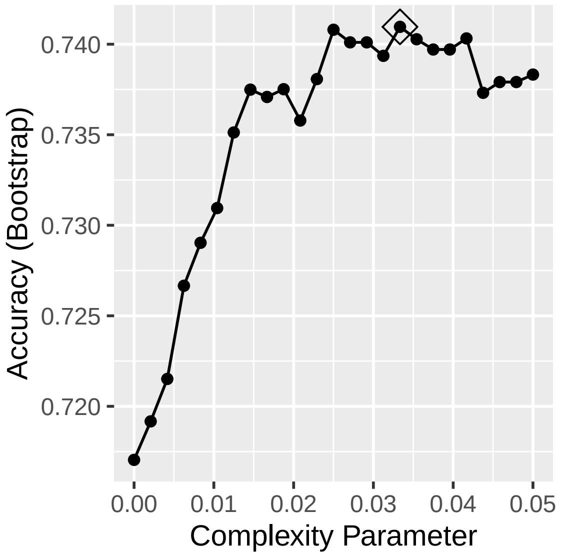
This is the overall accuracy of the KNN model.

Method	Accuracy
KNN model	0.7338129

2.4.3 CART model

The **CART** (Classification And Regression Tree) model is a predictive model, which explains how an outcome variable's values can be predicted based on other values. A CART output is a decision tree where each fork is a split in a predictor variable and each end node contains a prediction for the outcome variable.

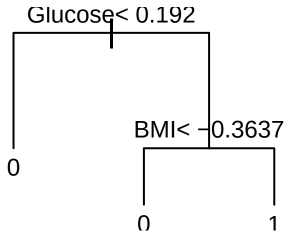
We will use cross-validation to choose **CP** (Complexity Prameter) value. The complexity parameter (CP) is used to control the size of the decision tree and to select the optimal tree size. If the cost of adding another variable to the decision tree from the current node is above the value of CP, then tree building does not continue.



This is the best setting for the CP.

```
##             cp
## 17 0.03333333
```

And this is the decision tree structure, which is based in 2 predictors only.



The variable importance of the CART model is as follows.

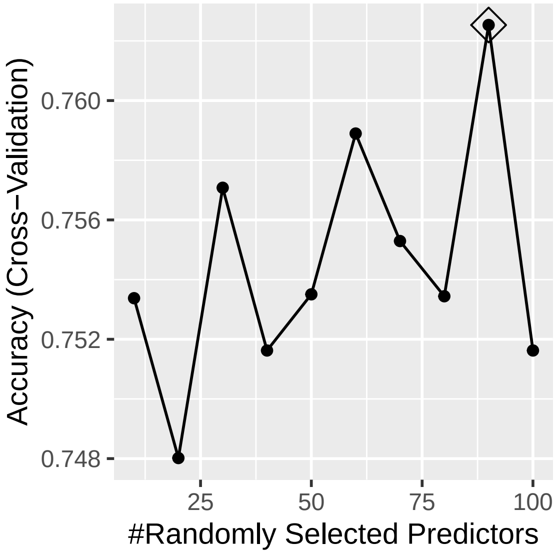
```
## rpart variable importance
##
##                          Overall
## Glucose                   100.00
## BMI                        81.78
## Age                        60.74
## Pregnancies                41.78
## Insulin                    26.77
## DiabetesPedigreeFunction   11.75
## SkinThickness               0.00
## BloodPressure               0.00
```

And this is the obtained overall Accuracy with the CART method.

Method	Accuracy
CART model	0.7553957

2.4.4 Random forest model

The **Random forest** model is an algorithm that randomly creates several small decission trees that all together make one forest. The solutions comes from a collection of decision model to improve acuracy.



We now check the variable importance of the model with the **varImp()** function of the caret package.

```
## rf variable importance
##
##                          Overall
## Glucose                  100.000
## BMI                       50.645
## DiabetesPedigreeFunction  27.788
## Age                       26.720
## BloodPressure              9.481
## Insulin                    3.339
## Pregnancies                2.572
## SkinThickness              0.000
```

Glucose seems to be the most important variable of this Random Forest model followed by **BMI**.

Method	Accuracy
Random Forest model	0.7985612

3.1 Results

This is the summary of the 4 different models results sorted by overall accuracy when training the models.

Method	Accuracy
Random Forest Model	0.7985612
Logistic regression model	0.7625899
CART Model	0.7553957
KNN Model	0.7338129

We will now assess the performance of the four models with hypothetical real results when using them with the **validation set**.

Method	Accuracy
KNN Model	0.7532468
Logistic regression model	0.7402597
CART Model	0.7402597
Random Forest Model	0.7142857

3.2 Discussion

We see that the model that better performs in terms of overall accuracy when developing the algorithm is the **Random forest model**. However, the best performing model with the **validation set** is the **KNN model**.

We can see that the **overall accuracy** of the different models is different when using the **validation set** and also the order of the most accurate models.

The reason of the different model performance when training and when using the validation set can be the small size of the Pima Indians Diabetes dataset.

We can also see that the results of the logistic regression model and the CART model are the same. We have to confirm if this is a casuality or not.

This is the result when comparing the prediction on the validation set with the logistic regression model and CART model with **identical()** function.

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

The identical() function tells us that prediction results are not the same. So, let's have a look at the written results.

These are the predicted values of the validation set using the **logistic regression** model.

```
##  [1] 0 1 1 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
## [77] 0
## Levels: 0 1
```

These are the predicted values of the validation set using the **CART** model.

```
##  [1] 0 0 1 0 1 0 0 1 1 0 0 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
## [77] 0
## Levels: 0 1
```

Now we can visually appreciate that the matrices are not the same.

Now we will have a look at the Confussion Matrix of both models.

Logistic regression model Confussion Matrix.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 43 13
##          1  7 14
## 
##                Accuracy : 0.7403
##                  95% CI : (0.6277, 0.8336)
##     No Information Rate : 0.6494
##     P-Value [Acc > NIR] : 0.05786
## 
##                   Kappa : 0.3989
## 
##  Mcnemar's Test P-Value : 0.26355
## 
##             Sensitivity : 0.8600
##             Specificity : 0.5185
##          Pos Pred Value : 0.7679
##          Neg Pred Value : 0.6667
##              Prevalence : 0.6494
##          Detection Rate : 0.5584
##    Detection Prevalence : 0.7273
##       Balanced Accuracy : 0.6893
## 
##        'Positive' Class : 0
## 
```

CART model Confussion Matrix.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46 16
##          1  4 11
##
##                Accuracy : 0.7403
##                  95% CI : (0.6277, 0.8336)
##     No Information Rate : 0.6494
##     P-Value [Acc > NIR] : 0.05786
##
##                   Kappa : 0.3647
##
##  Mcnemar's Test P-Value : 0.01391
##
##             Sensitivity : 0.9200
##             Specificity : 0.4074
##          Pos Pred Value : 0.7419
##          Neg Pred Value : 0.7333
##              Prevalence : 0.6494
##          Detection Rate : 0.5974
##    Detection Prevalence : 0.8052
##       Balanced Accuracy : 0.6637
##
##        'Positive' Class : 0
##
```

Now we can perfectly see that despite the predictive models are different, the overall accuracy in both cases is the same.

However, sensitivity and specificity are quite different, being the CART model specificity specially poor. More similar is the balanced accuracy of both models.

4. Conclusions

The four predictive models we have build perform different when training and when being used with a hypothetical real case.

The size of the dataset conditions the good performance of the predective models.

The most meaningful predictors are different for each type of model, but **Glucose** and **BMI** are always in the top 2. This conclusion has been proofed except for the KNN model as in this case there is no way to check the variable importance.

Glucose and **BMI** are the predictors with highest correlation with the **Outcome**, which is the fact to be or not to be diabetic. Higher values increase the probability to be diabetic.

Future work

It would be interesting to study if other predictive models based on different algorithms perform in a similar way and are also sensitive to the size of the dataset.

It is also possible to find better tuning parameters for the used models.

It is also an interesting exercice to study which of the predictive models shows a higher overall accuracy after a Montecarlo simulation that generates many train and test set creation. We could then choose the best performing model. The problem is that this research task can consume a lot of time.

We can repeat the previous exercice but this time with different proportions of the train and test set and observe which proportion benefits the predictive overall accuracy of the models.