Assigment 3

Yi Hung Chen

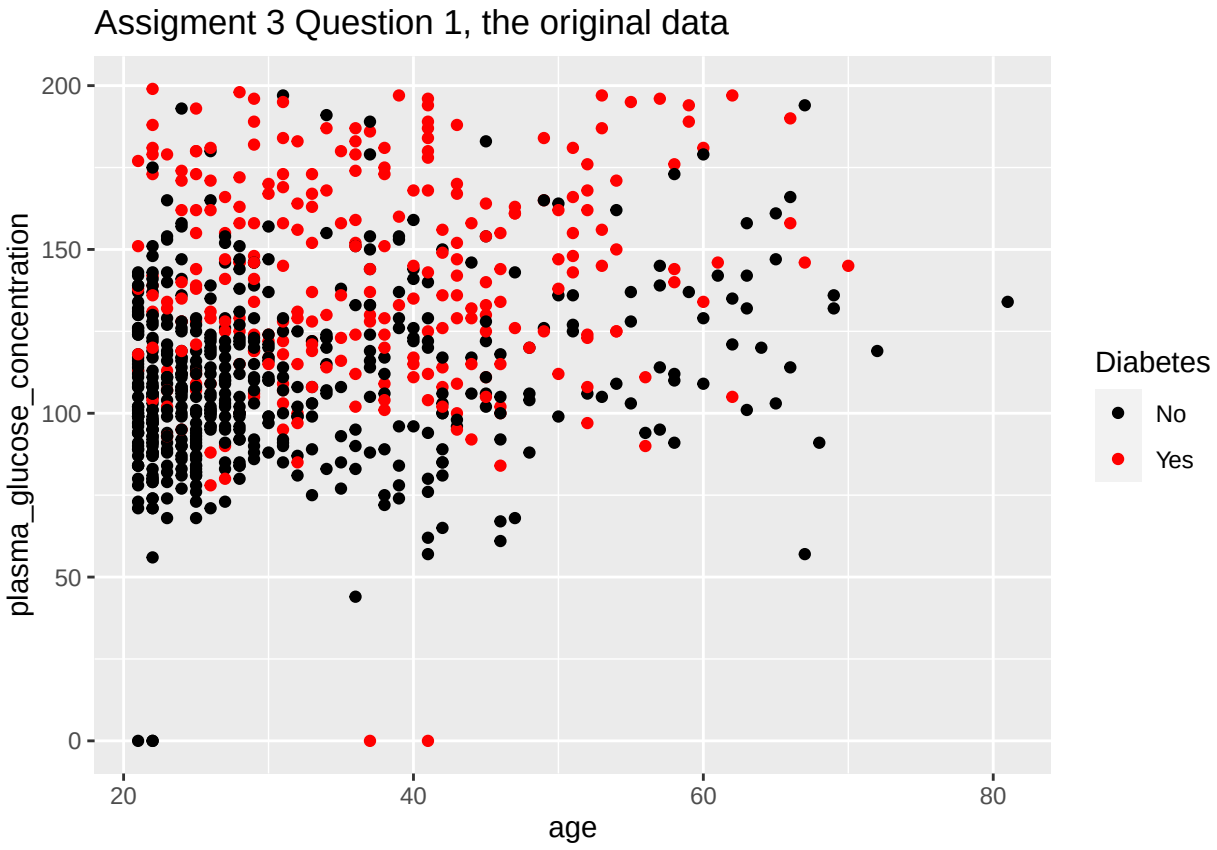
2022-11-12

Assignment 3

First, the data from the Excel file *pima-indians-diabetes* will be imported and the column names are changed

Ex.3.1

Make a scatter plot showing a Plasma glucose concentration on Age where observations are colored by Diabetes levels.



Q. Do you think that Diabetes is easy to classify by a standard logistic regression model that uses these two variables as features? In my opinion, it is not easy to classify Diabetes using standard logistic regression model (using age and Plasma glucose concentration as model features). As we can observed on the plot, although the one who "does not" have Diabetes are more concentrate on the bottom left side of the graph, it is still not easy to classify if the age gets older.

Ex.3.2

Train a logistic regression model with y =Diabetes as target, x_1 =Plasma glucoseconcentration and x_2 =Age as features. Make a prediction for all observations by using r = 0.5

```
model_1 <- glm(diabetes ~ plasma_glucose_concentration +
    age, data = diabetes_data_1, family = binomial)
summary(model_1)$coef
```

1. Use "glm" function with family = binomial to train the logistic regression model

```
##                                Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept)                -5.89785793 0.462449826 -12.753509 2.980232e-37
## plasma_glucose_concentration  0.03558250 0.003288130  10.821500 2.722834e-27
## age                            0.02450157 0.007379078   3.320411 8.988507e-04
```

Q. Report the probabilistic equation of the estimated model According to the coefficient, the probabilistic equation is

$$p = \frac{e^{-5.89785793+0.03558250 plasmaglucoseconcentration+0.02450157age)}}{1+e^{-5.89785793+0.03558250*plasmaglucoseconcentration+0.02450157*age}}$$

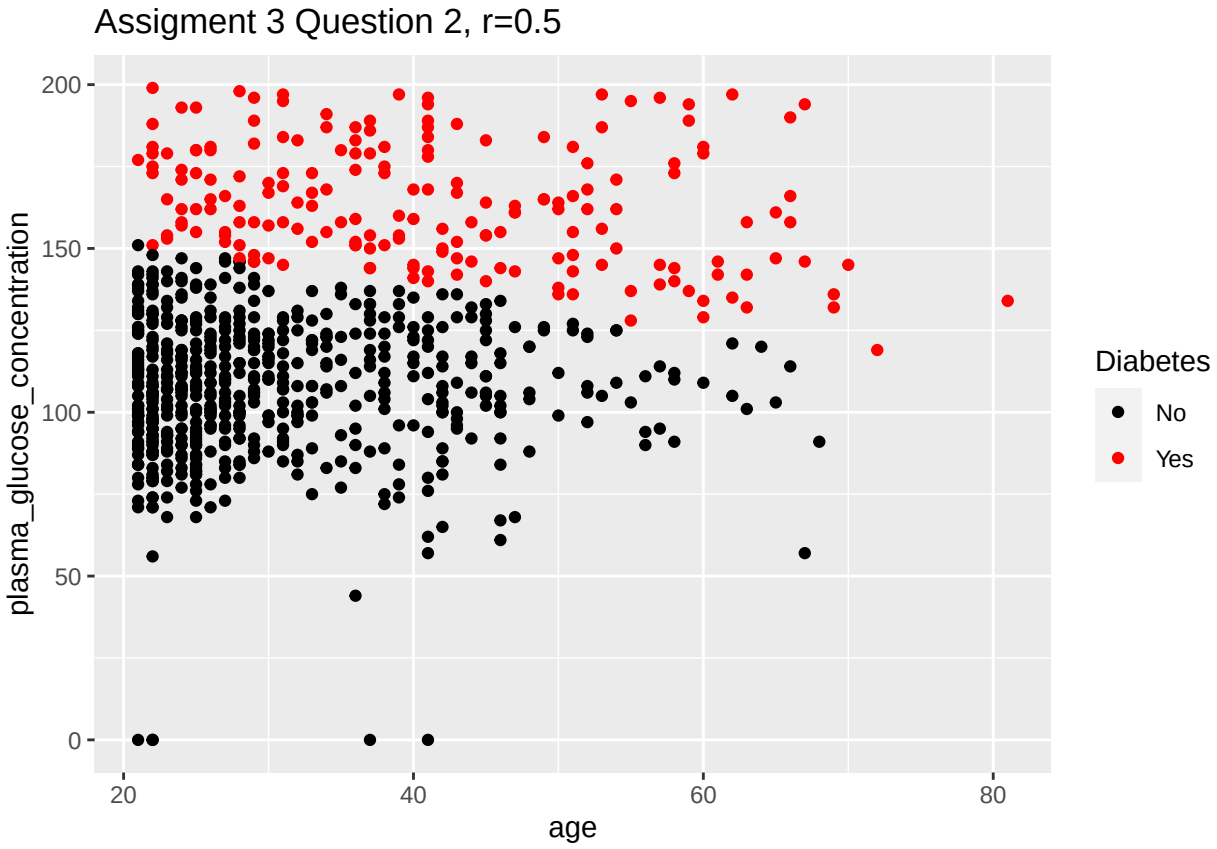
```
missmissclassification_ex2
```

2. Compute training misclassification error

```
## [1] 0.2659713
```

The missmissclassification error is 0.2659713

3. Plot the scatter plot showing the predicted values of Diabetes



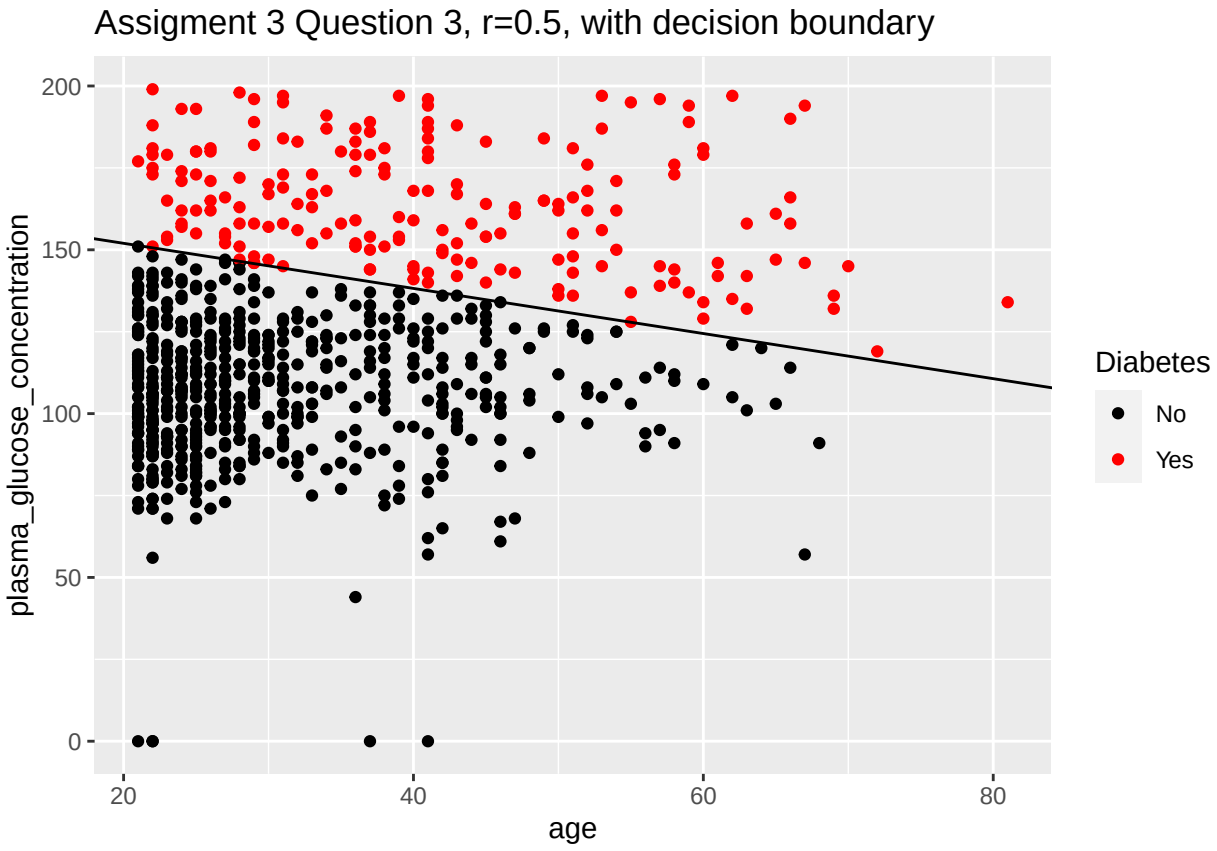
Q. Comment on the quality of the classification by using these results In my opinion, the quality of the classification is mediocre. Although the overall missclafication rate (26.59713%) is not high, the prediction of older people is not ideal.

EX.3.3

(a) Report the equation of the decision boundary between the two classes of step 2 The decision boundary equation of step 2 is

$$plasma\ glucose\ concentration = \frac{5.897858}{0.0355825} + \frac{-0.02450157}{0.0355825} age = 165.7516 - 0.6885848 age$$

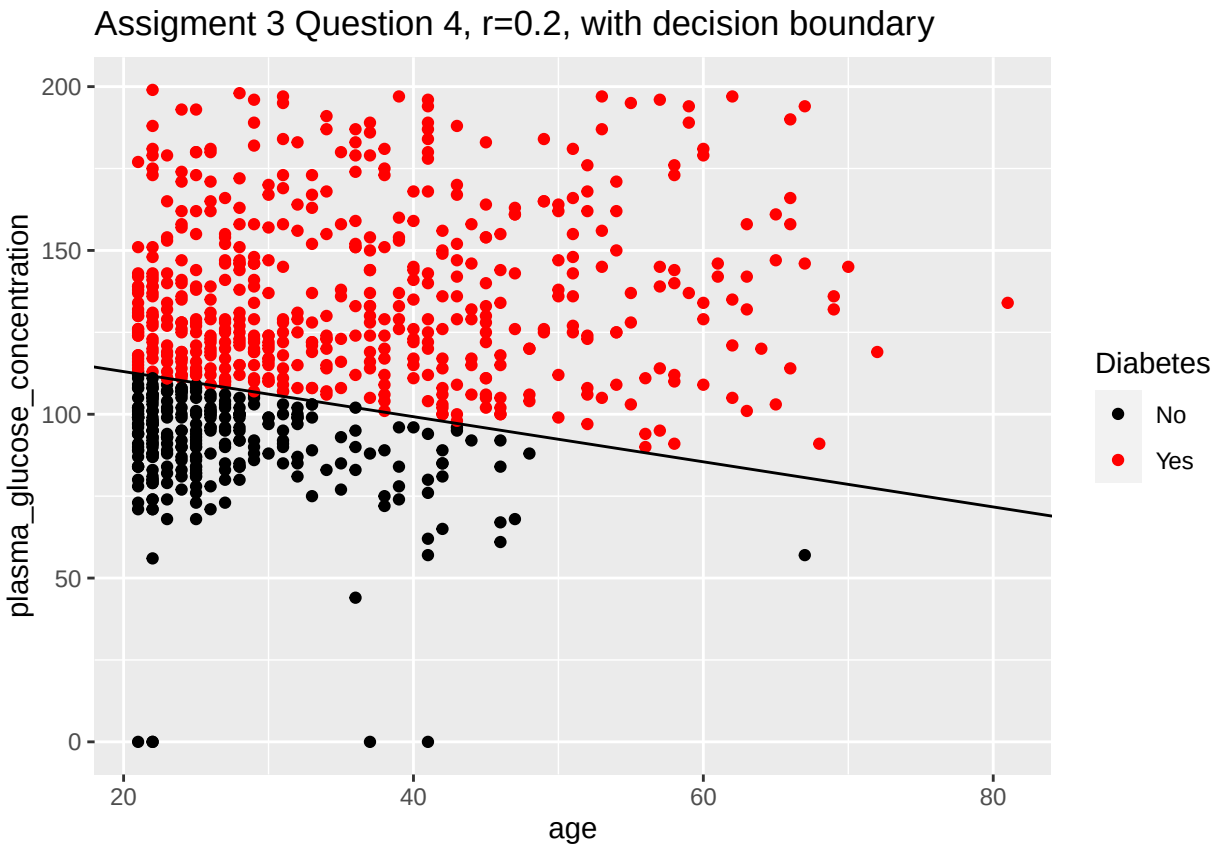
(b) Add a curve showing this boundary to the scatter plot

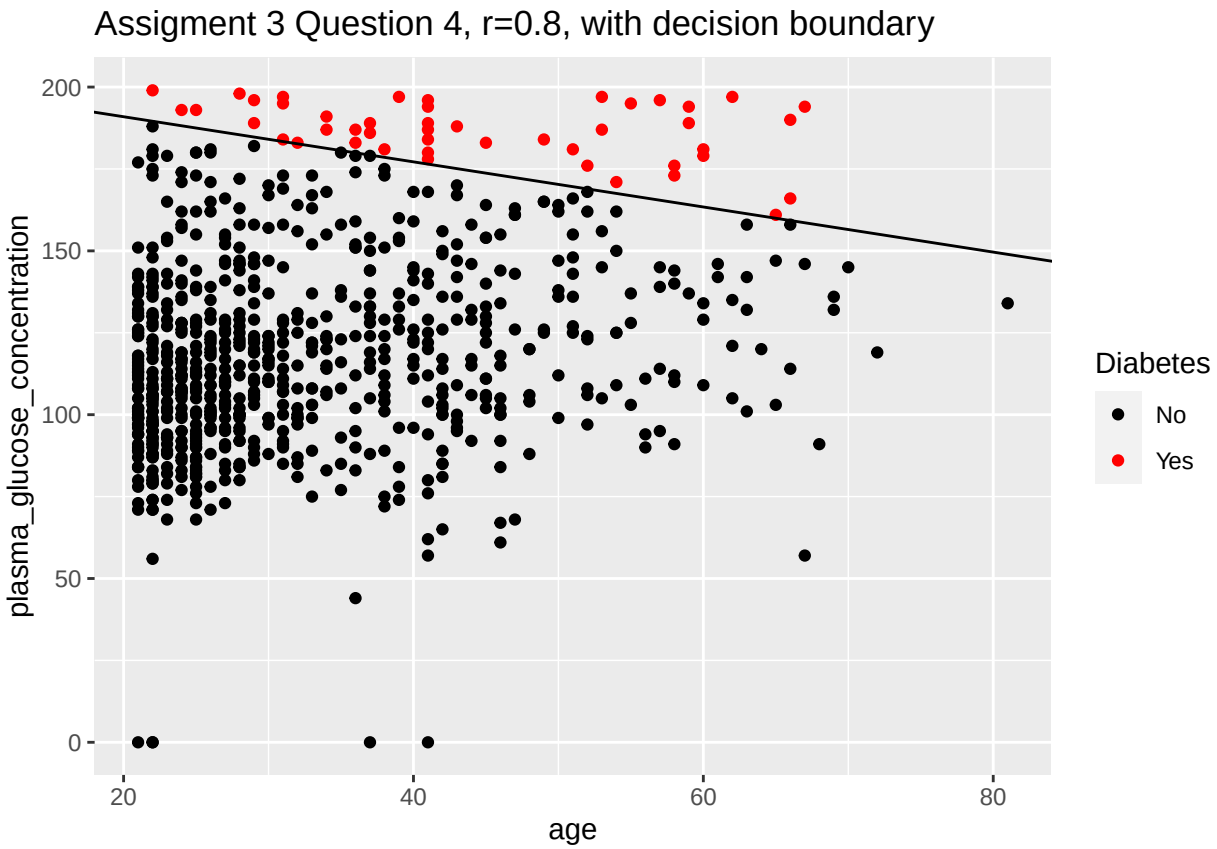


Q. Comment whether the decision boundary seems to catch the data distribution well. As the graph shown, the decision boundary seperate the prdiction of Diabetes very well in this case.

EX.3.4

Make same kind of plots as in step 2 but use thresholds r = 0.2 and r = 0.8

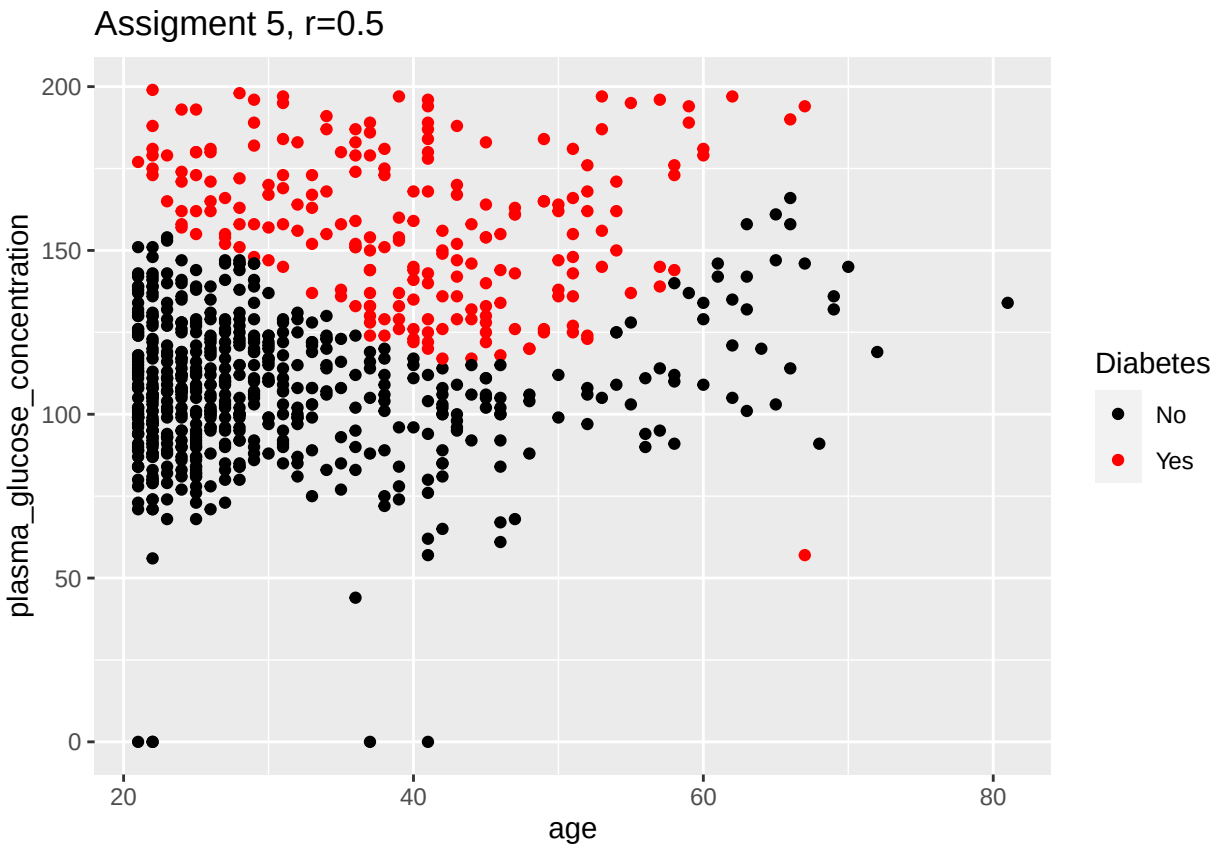




Q. comment on what happens with the prediction when r value changes When r = 0.8, the decision boundary of having diabetes will move toward to the top, which leaves less people being predict to have Diabetes. For r = 0.2 the opposite happen the decision boundary of having diabetes will move toward to the bottom, , which predict less people having Diabetes.
Both cases will result with higher missclasification errors, which means the prediction became less accurate.

EX.3.5

Perform a basis function expansion trick by computing new features




```
## The missclassification is 0.2464146
```

Q. What can you say about the quality of this model compared to the previous logistic regression model? How have the basis expansion trick affected the shape of the decision boundary and the prediction accuracy? According to the missclasification error(0.2464146), this model is by far the best quality one compares to other model in this assignment. The basis expansion trick change the decision boundary to a "U" shape and the prediction is closer to the original data. However, due to higher dimension of the features, the decision boundary is hard to visualize on a 2-D graph, but we can still observe by looking at the color difference.
To sum up, using the basis expansion improves the prediction accuracy.

Appendix

```
# =====Assignment 3===== =====Set Up=====
diabetes_data <- read.csv("D:/pima-indians-diabetes.csv")
colnames(diabetes_data) <- c("number_of_times_pregnant",
    "plasma_glucose_concentration", "blood_pressure", "triceps_skinfold_thickness",
    "serum_insulin", "bmi", "diabetes_pedigree_function",
    "age", "diabetes")
library(ggplot2)
# ===== ÉX 3.1===== create 'diabetes_data_1' so the
# original data wont be affect
diabetes_data_1 <- diabetes_data

# use 'as.factor' so 1 means has diabetes, 0 means no
# diabetes
diabetes_data_1$diabetes <- as.factor(ifelse(diabetes_data_1$diabetes ==
    1, "Yes", "No"))  # use 'as.factor' so 1 means has diabetes, 0 means no diabetes
plot_assigment3_q1 <- ggplot(diabetes_data_1, aes(x = age,
    y = plasma_glucose_concentration, color = diabetes)) +
    geom_point() + labs(title = "Assigment 3 Question 1, the original data",
    colour = "Diabetes") + scale_color_manual(values = c("#000000",
    "#ff0000"))

plot_assigment3_q1
# ===== ÉX 3.2===== =====1=====
model_1 <- glm(diabetes ~ plasma_glucose_concentration +
    age, data = diabetes_data_1, family = binomial)
summary(model_1)$coef
diabetes_data_1$probabilities <- predict(model_1, diabetes_data_1,
    type = "response")
# The type='response' option tells R to output
# probabilities of the form P(Y = 1|X), as opposed to
# other information such as the logit.

diabetes_data_1$predicted_classes_0.5 <- as.factor(ifelse(diabetes_data_1$probabilities >
    0.5, "Yes", "No"))

# =====2=====
missclass = function(X, X1) {
    n = length(X)
    return(1 - sum(diag(table(X, X1)))/n)
}
missmissclassification_ex2 <- missclass(diabetes_data_1$diabetes,
    diabetes_data_1$predicted_classes_0.5)
missmissclassification_ex2
# =====3=====
plot_assigment3_q2 <- ggplot(diabetes_data_1) + geom_point(aes(x = age,
    y = plasma_glucose_concentration, color = predicted_classes_0.5)) +
    labs(title = "Assigment 3 Question 2, r=0.5", colour = "Diabetes") +
    scale_color_manual(values = c("#000000", "#ff0000"))

plot_assigment3_q2
```

```r
# ===== ÉX 3.3===== To correct the intercept on the
# plot if the threshold is not 0.5
inverse_logit <- function(threshold) {
    return(-log((1 - threshold)/threshold))

}
decision_boundary <- function(a, b, c, ...) {
    # function to plot decision boundary
    slope <- -a/b
    intercept <- -c/b
    geom_abline(slope = slope, intercept = intercept, ...)

}
plot_assigment3_q3 <- ggplot(diabetes_data_1) + geom_point(aes(x = age,
    y = plasma_glucose_concentration, color = predicted_classes_0.5)) +
    labs(title = "Assigment 3 Question 3, r=0.5, with decision boundary",
        colour = "Diabetes") + scale_color_manual(values = c("#000000",
    "#ff0000")) + decision_boundary(model_1$coefficients[3],
    model_1$coefficients[2], model_1$coefficients[1] - inverse_logit(0.5))
plot_assigment3_q3


# ===== ÉX 3.4===== ===== r=0.2 =====
diabetes_data_1$predicted_classes_0.2 <- as.factor(ifelse(diabetes_data_1$probabilities >
    0.2, "Yes", "No"))

plot_assigment3_q4_r0.2 <- ggplot(diabetes_data_1) + geom_point(aes(x = age,
    y = plasma_glucose_concentration, color = predicted_classes_0.2)) +
    labs(title = "Assigment 3 Question 4, r=0.2, with decision boundary",
        colour = "Diabetes") + scale_color_manual(values = c("#000000",
    "#ff0000")) + decision_boundary(model_1$coefficients[3],
    model_1$coefficients[2], model_1$coefficients[1] - inverse_logit(0.2))

plot_assigment3_q4_r0.2

# ===== r=0.8 =====
diabetes_data_1$predicted_classes_0.8 <- as.factor(ifelse(diabetes_data_1$probabilities >
    0.8, "Yes", "No"))

plot_assigment3_q4_r0.8 <- ggplot(diabetes_data_1) + geom_point(aes(x = age,
    y = plasma_glucose_concentration, color = predicted_classes_0.8)) +
    labs(title = "Assigment 3 Question 4, r=0.8, with decision boundary",
        colour = "Diabetes") + scale_color_manual(values = c("#000000",
    "#ff0000")) + decision_boundary(model_1$coefficients[3],
    model_1$coefficients[2], model_1$coefficients[1] - inverse_logit(0.8))
plot_assigment3_q4_r0.8

# ===== ÉX 3.5=====
diabetes_data_ex5 <- diabetes_data
# Create new data frame so it won't affect the
# previous data frame

# Add new features
```

```r
diabetes_data_ex5$z1 <- (diabetes_data_ex5$plasma_glucose_concentration)^4
diabetes_data_ex5$z2 <- (diabetes_data_ex5$plasma_glucose_concentration)^3 *
    diabetes_data_ex5$age
diabetes_data_ex5$z3 <- (diabetes_data_ex5$plasma_glucose_concentration)^2 *
    (diabetes_data_ex5$age)^2
diabetes_data_ex5$z4 <- (diabetes_data_ex5$plasma_glucose_concentration)^1 *
    (diabetes_data_ex5$age)^3
diabetes_data_ex5$z5 <- (diabetes_data_ex5$age)^4

# Do the model using glm with new features
model_2 <- glm(diabetes ~ plasma_glucose_concentration +
    age + z1 + z2 + z3 + z4 + z5, data = diabetes_data_ex5,
    family = binomial)

diabetes_data_ex5$probabilities <- predict(model_2, diabetes_data_ex5,
    type = "response")
diabetes_data_ex5$predicted_classes_0.5 <- as.factor(ifelse(diabetes_data_ex5$probabilities >
    0.5, "Yes", "No"))
plot_assigment3_q5 <- ggplot(diabetes_data_ex5, aes(x = age,
    y = plasma_glucose_concentration)) + geom_point(aes(x = age,
    y = plasma_glucose_concentration, color = predicted_classes_0.5)) +
    scale_color_manual(values = c("#000000", "#ff0000")) +
    labs(title = "Assigment 5, r=0.5", colour = "Diabetes")
plot_assigment3_q5

missmissclassification_ex5 <- missclass(diabetes_data_ex5$diabetes,
    diabetes_data_ex5$predicted_classes_0.5)
cat("The missclassification is", missmissclassification_ex5)
```