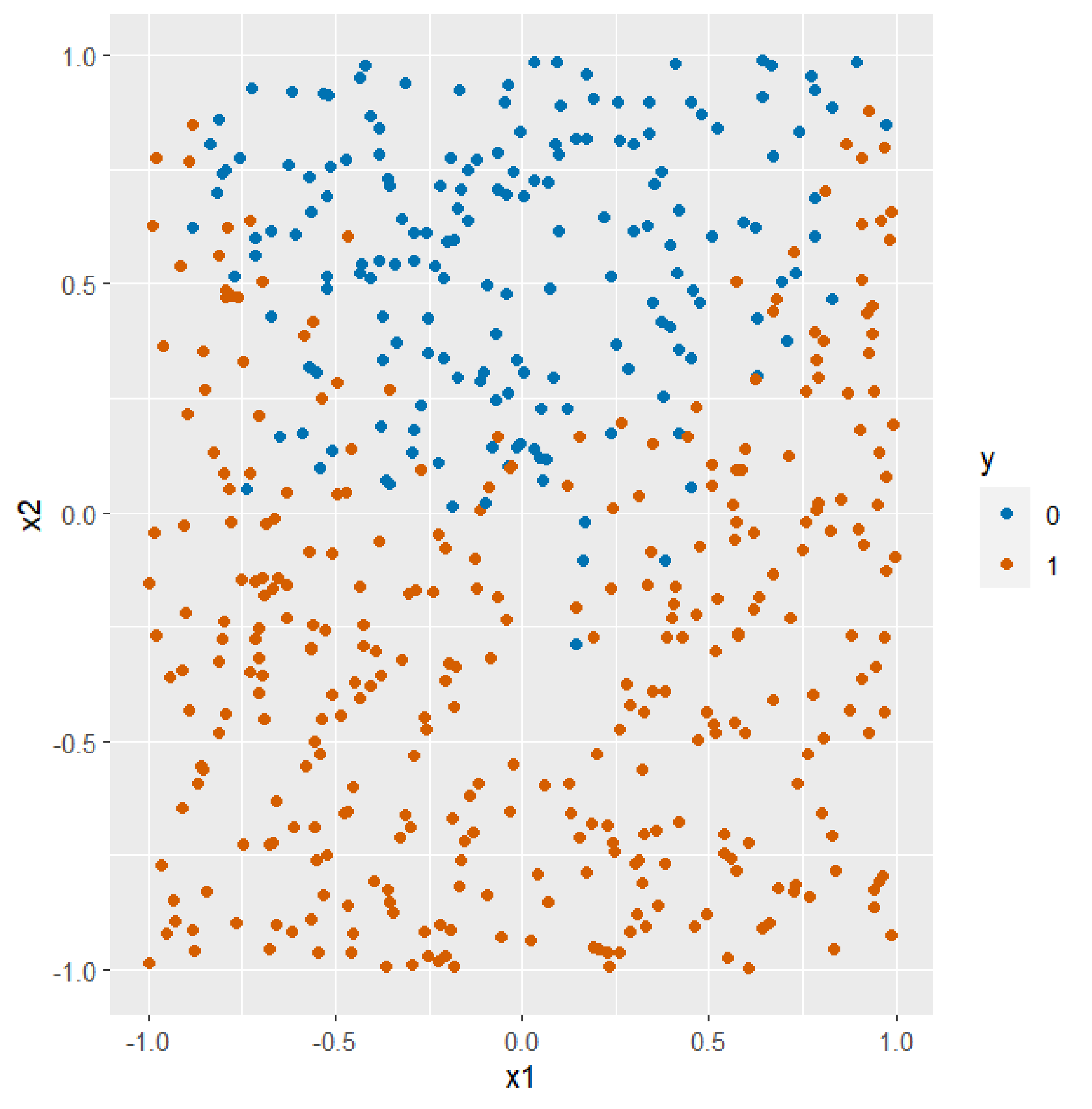
Technical University of Dortmund Summer Semester 2023
Data Science Exercise Sheet 02
Group member name(s): Ahmed Arian Sajid, Sabrina Sultana, Sharmin Ahmad
Group member UID(s): 235061, 235062, 230239

Advanced Statistical Learning

Answer to Exercise 1.a

Let us visualize the following code;

```
x1 = runif(500, -1, 1)
x2 = runif(500, -1, 1)
y = ifelse(x1^2 + rnorm(500, 0, 0.2) < x2, 0, 1)
```



A logistic model with the hypothesis space $\mathcal{H} := \{f(x) = x^T\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 | \theta \in \mathbb{R}^3\}$ will not result in a good fit, because this is a linear classifier. We can see from the plot that the classes cannot be split linearly, i.e., any linear function can't split it.

Answer to Exercise 1.b

As we can see, the data-generating process of y has the square of $x1$; we need to include the effect of x_1^2 to our hypothesis space $\mathcal{H}$, which would result in a better fit of logistic regression. So, a new Hypothesis space would be:

$$\mathcal{H} := \{f(x) = x^T\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 | \theta \in \mathbb{R}^4\}$$

Answer to Exercise 1.c

To find the decision boundary, at first, we have to calculate the coefficients using the Hypothesis space from 1.b and the given data-generating process, we are using the following R Code to find the coefficients:

```
set.seed(123)
x1 <- runif(500, -1, 1)
x2 <- runif(500, -1, 1)
x3<-x1^2
y <- ifelse(x1^2 + rnorm(500, 0, 0.2) < x2, 0, 1)
fit<-glm(y ~ x1 + x2 + x3, family = "binomial")
(theta <- coef(fit))

(Intercept)          x1          x2          x3
-0.1178140   0.1665535  -8.5038775   8.6178639
```

here given $\pi(x) = 0.5$
therefore,

$$\begin{aligned}
&\frac{1}{1 + \exp(-x^T\theta)} = 0.5 \\
&\implies 1 + \exp(-x^T\theta) = 2 \\
&\implies \exp(-x^T\theta) = 1 \\
&\implies (-x^T\theta) = \log(1) \\
&\implies x^T\theta = 0 \\
&\implies \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 = 0 \\
&\implies \theta_0 + \theta_1 x_1 + \theta_3 x_1^2 = -\theta_2 x_2 \\
&\implies -\frac{\theta_0}{\theta 2} - \frac{\theta_1}{\theta_2} x_1 - \frac{\theta_3}{\theta_2} x_1^2 = x_2
\end{aligned}$$

using the above values of coefficients, we get:

$$\begin{aligned}
&-\frac{-0.1178}{-8.5039} - \frac{0.1666}{-8.5039} x_1 - \frac{8.6179}{-8.5039} x_1^2 = x_2 \\
&\implies -0.014 + 0.02 x_1 + 1.013 x_1^2 = x_2
\end{aligned}$$

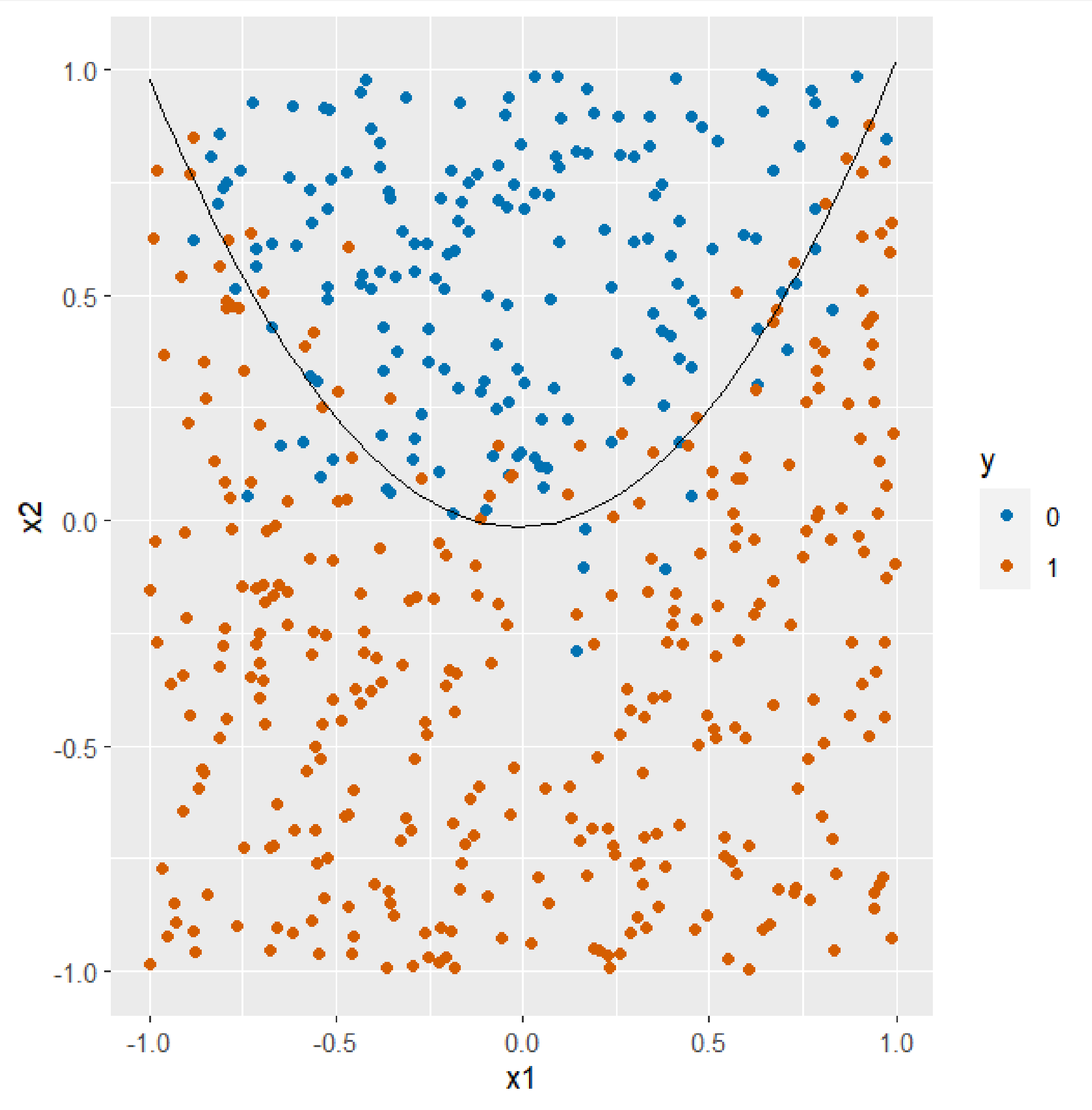
which is the decision boundary.

Answer to Exercise 1.d

We visualize the data and decision boundary found from 1.c using the following R Code:

```
set.seed(123)
x1 <- runif(500, -1, 1)
x2 <- runif(500, -1, 1)
y <- ifelse(x1^2 + rnorm(500, 0, 0.2) < x2, 0, 1)
df <- data.frame(x1 = x1, x2 = x2, y = factor(y))
boundary <- function(x){-0.014 + 0.02*x + 1.013*x^2}

library(ggplot2)
ggplot(df, aes(x = x1, y = x2, color = y)) +
geom_point() +
stat_function(fun = boundary,aes(x = x1,y=x2), color = "black")+
scale_color_manual(values = c("#0072B2", "#D55E00"))
```



Answer to Exercise 2

- (1 point) Do an internet research and shortly explain the ideas of One-versus-One and Oneversus-All. The original logistic regression is a probability estimator. Explain why this is not true for the extended multi-class variants.

  Solution:
  One-versus-One: OVO classification scheme divide an m class problem into m(m-1) / 2 binary problems. By using this classification approach, Each problem is split by a binary classifier which is responsible of distinguishing between a different pair of classes.
  One-versus-All: OVA classification scheme divides an m class problem into m binary problems. Each problem is split by a binary classifier which is responsible of distinguishing one of the classes from all other classes

  The logistic regression is not true for the extended multi-class variants because in One-versus-All can result in imbalanced class distributions, where one class has many more samples than the other and In One-versus-One does not directly provide probability estimates for each class.

- **Solution:**
  Let us consider the following data situation to train a One-versus-All logistic regression,

```
data <- data.frame(x1 = runif(500, -1, 1),
                   x2 = runif(500, -1, 1),
                   r = rnorm(500, 0, 0.2))

data$y <- ifelse(data$x1+data$r < 0 & data$x1+data$r < data$x2, 1,
          ifelse(data$x1 + data$r > 0 & data$x1 + data$r > -data$x2, 2, 3))

model1 <- glm(y == 1 ~ x1 + x2, data = data, family = binomial())
model2 <- glm(y == 2 ~ x1 + x2, data = data, family = binomial())
model3 <- glm(y == 3 ~ x1 + x2, data = data, family = binomial())

prob1 <- predict(model1, newdata = data, type = "response")
prob2 <- predict(model2, newdata = data, type = "response")
prob3 <- predict(model3, newdata = data, type = "response")

pred <- ifelse(prob1 > prob2 & prob1 > prob3, 1,
           ifelse(prob2 > prob1 & prob2 > prob3, 2, 3))
```

- (1 point) Make a prediction for the two new observations (0, 0.5) and (0.5, -0.5). Use the softmax transformation to transform the resulting scores into probabilies.
  Solution:

```
new_data <- data.frame(x1 = c(0, 0.5),
                       x2 = c(0.5, -0.5))
 scores1 <- predict(model1, newdata = new_data, type = "response")
 scores2 <- predict(model2, newdata = new_data, type = "response")
 scores3 <- predict(model3, newdata = new_data, type = "response")

 scores <- cbind(scores1, scores2, scores3)
```

  Using softmax transformation to transform the resulting scores into probabilies.

```
probs <- apply(scores, 1, function(x) exp(x) / sum(exp(x)))
```

- (2 points) Visualise both the data and the decision boundary

Solution:

```
library(ggplot2)
ggplot(data = data.frame(x1, x2, y)) +
geom_point(aes(x = x1, y = x2, color = as.factor(y)))

coef <- coef(model)
b0 <- coef[1]
b1 <- coef[2]
b2 <- coef[3]
intercept <- -b0/b2
slope <- -b1/b2

ggplot(data = data.frame(x1, x2, y)) +
geom_point(aes(x = x1, y = x2, color = as.factor(y))) +
geom_abline(intercept = intercept, slope = slope, color = "black", size = 1)
```