

## Exercise Sheet 3

1. Let  $\Omega$  be our sample space, and define two independent random variables on it:  $Y_1$  and  $Y_2$ . Let  $Y_i \sim \text{Bernoulli}(\theta)$  for  $i = 1, 2$ . Note that Bernoulli a distributed r.v.  $Y_i$  takes values 0 or 1 with the parameter  $\theta$  corresponding to  $P(Y_i = 1)$ . We consider only two parameters  $\theta \in \Theta = \{\frac{1}{3}, \frac{2}{3}\}$ .

- What possible realisations of  $(Y_1, Y_2)$  could you get?
  - For each of them, write down the likelihood for both parameter values.
  - What is the maximum likelihood estimator for each sample?
  - Interpret.
  - Once you have obtained  $\hat{\beta}$  how would you define your Bayes classifier?
2. For the logit model we had

$$P(Y_i = y|X_i = x_i; \beta) = \frac{1}{1 + e^{-x_i^T \beta}}$$

and log-likelihood

$$l(\beta) = \sum_{i=1}^n [y_i \log P(Y_i = 1|X_i = x_i; \beta) + (1 - y_i) \log(1 - P(Y_i = 1|X_i = x_i; \beta))].$$

- Explain the principle of MLE. How do we find  $\beta$ ?
  - For our numerical procedure we require the  $l'(\beta) = \frac{dl(\beta)}{d\beta}$ . Derive it.
  - Does the solution look somewhat familiar? Explain.
3. We are trying to predict individual **defaults** ( $y$ ) using account **balance** ( $x_1$ ) and **income** ( $x_2$ ) as inputs/predictors. You can use generative AI as guidance for the following exercises but make sure to always double-check!
- Install and load the R library **ISLR** and use the data **Default**.
  - Split the dataset into a training and a validation set.
  - Plot **defaults** as a function of **balance**.
  - Add a logistic regression line to this plot.
  - Now estimate the logistic regression for the model with both  $x_1$  and  $x_2$  as predictors.
  - Plot the defaults ( $y$ ) with different colours in  $(x_1, x_2)$  space.
  - Derive and add the Bayes decision boundary to this plot.
4. *Advanced:* Implement the Maximum Likelihood estimator for  $\hat{\beta}$  based on the likelihood (and its gradient) in Question 1. *Hint:* use the function `optim(b, fn = ll, gr = grad)` where **b** is the start value, **ll** is the likelihood function (as a function of **b**) and **grad** is the gradient of **ll** also as a function of **b**.