## Solution 1: Gradient Boosting

- (a) The loss is calculated by the negative log-likelihood by:  $L(y, f) = -\ell(f) = -(const (\log_2(y) f)^2/2)$  (1P)
  - The pseudo residuals are then calculated by:  $\tilde{r}(f) = -\partial L(y, f)/\partial f = (\log_2(y) f)$  (1P)
- (b) Use  $\tilde{y} = \log_2(y) = (0, 1, 2)$  (1P)
  - (i)  $\hat{f}^{[0]}(\boldsymbol{x}) = \bar{\tilde{y}} = 1$  as this is the optimal constant model for squared error. (1P)
  - (ii)  $\tilde{r}^{[1]} = \log_2(y) \hat{f}^{[0]}(x) = (-1, 0, 1)$  (1P)
  - (iii)  $R_t^{[1]}$ , t = 1, 2 will split using  $\boldsymbol{x}_1$ , as  $\boldsymbol{x}_2$  carries no information. Since  $x_1^{(1)} = x_1^{(2)}$ ,

$$R_1 = -0.5I(x_1 \ge 0.5)$$

and

$$R_2 = 1I(x_1 < 0.5).$$

(2P)

(iv) 
$$\hat{f}^{[1]}(\mathbf{x}) = \hat{f}^{[0]}(\mathbf{x}) + 1(-0.5, -0.5, 1) = (0.5, 0.5, 2)$$
 (1P)

(v) 
$$\tilde{r}^{[2]} = \log_2(y) - \hat{f}^{[1]}(\boldsymbol{x}) = (-0.5, 0.5, 0)$$
 (1P)

- (c) Nothing, because there is no information that can be used to further improve the model. (1P)
- (d) (i) M grows: capacity will increase and the algorithm may eventually overfit (1P)
  - (ii) n grows: capacity will stay the same and the algorithm may underfit (1P)