Exercise 1: PDP and ICE in case of interactions

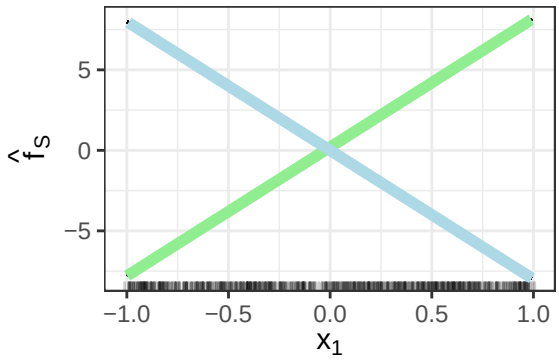
Let us assume that we fitted the following linear regression model with two features:

$$\hat{f}(\mathbf{x}) = \hat{f}(x_1, x_2) = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2 + \hat{\beta}_0. \tag{1}$$

a) Analytically derive the PD function of feature $S = \{1\}$ (with $C = \{2\}$).
Hint: In the lecture slides we derived the PD function for a linear regression model without an interaction term.
Hint: In the end, your solution should include terms like the expected value of X_2.

b) Let us assume that the estimated coefficients in Equation 1 are $\hat{\beta}_0 = 0$, $\hat{\beta}_1 = -8$, $\hat{\beta}_2 = 0.2$ and $\hat{\beta}_3 = 16$. Furthermore, $X_1 \sim Unif(-1, 1)$ is uniformly distributed between -1 and 1, and $X_2 \sim B(1, 0.5)$ origins from a Bernoulli distribution. Compute the exact PD function of feature X_1 using your derived function of a).

c) The following plot displays the ICE curves of X_1. The rugs show the marginal distribution of X_1. Please note that since X_2 is binary, we do not receive n individual ICE curves, but indeed only two unique ones. Derive the conditional expectation functions $\hat{f}_1^{(i)}(x_1)$ of X_1 for group $X_2 = 1$ and for group $X_2 = 0$ using Equation 1 given the estimated coefficients of b). Which color coding (light green or light blue) reflects the group $X_2 = 1$, which one group $X_2 = 0$?



d) Add the PDP you derived in b) to the plot. Use this example to explain briefly why it is advisable to display not only the PDP but also the ICE curves, when visualizing the feature effects of a specific model.

e) **Implementation: ICE and PDP.** Write functions that, given a fitted model object, a data matrix $\mathbf{X}$ and a feature index j, a specified grid of values does:

1. `ice(model, X, j, grid)` - returns an $n \times g$ matrix of ICE predictions (one curve per row, one grid value per column);
2. `pdp(model, X, j, grid)` - returns the length-g PD vector by averaging the ICE matrix column-wise.

f) **Implementation: centred variants.** Extend the code with

1. `c_ice(model, X, j, grid, ref = 1)` - centres every ICE curve by subtracting its value at the reference grid index `ref` (the "vertical shift" in the slides which is often set to the minimum value);
2. `c_pdp(model, X, j, grid, ref = 1)` - averages the centred ICE curves to obtain a centred PDP which is zero at the reference point.

Use the definitions of c-ICE and c-PDP from the lecture slides to guide your implementation.

Exercise 2: Derivative, Forward, and Non-Linearity Effects

Let us consider the following non-linear regression model:

$$\hat{f}(x_1, x_2) = \theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2 + \theta_{1,2} x_1 x_2 \tag{2}$$

with parameters $\theta_0 = 0$, $\theta_1 = 1$, $\theta_2 = 0.5$, and $\theta_{1,2} = 2$.

a) Derive the analytic expression for the *derivative marginal effect* (dME) of feature x_1.

b) Derive the expression for the *forward marginal effect* (fME) of x_1 with step size $h_1 > 0$.

c) Let $x_1 = 1$, $x_2 = 2$, and $h_1 = 1$. Compute $\text{dME}_1(1, 2)$ and $\text{fME}_1(1, 2; h_1 = 1)$ and explain why these two results differ and interpret the additional terms appearing in the fME.

d) **Non-Linearity Measure (NLM):** We want to test how well the secant (local linear approximation) fits the model output along the direction of x_1. The NLM defined below quantifies how well the local linear approximation (secant) explains the variation in the model output along the direction of h_1. For T grid points $(\mathbf{x}^{(i)})_{i=1}^T$, it is defined as:

$$\text{NLM} = 1 - \frac{\underbrace{\sum_{i=1}^{T} \left[\hat{f}(\mathbf{x}^{(i)}) - g(t_i)\right]^2}_{\text{SSR: error of secant}}}{\underbrace{\sum_{i=1}^{T} \left[\hat{f}(\mathbf{x}^{(i)}) - \bar{f}\right]^2}_{\text{SST: total variance}}}.$$

- $t_i \in (0, 1)$ parametrises the path (here only in x_1 direction): $\mathbf{x}^{(i)} = (x_1 + t_i h_1,\ x_2)$.
- $g(t_i) = \hat{f}(x_1, x_2) + t_i \cdot \text{fME}_1(x_1, x_2; h_1)$ is the secant (straight line) through the two end-points.
- $\bar{f} = \dfrac{1}{T} \displaystyle\sum_{i=1}^{T} \hat{f}(\mathbf{x}^{(i)})$ is the mean prediction on the path.
- Hence NLM is an R^2: 1 means perfect linear fit, values $\ll 1$ reveal curvature.

Compute the NLM for $x_1 = 1$, $x_2 = 2$, and stepsize $h_1 = 0.5$. Interpret your result in relation to the fit of a local linear approximation using the $T = 9$ equidistant points below $t_i = 0.1, 0.2, \ldots, 0.9$:

i	t_i	$x_1^{(i)} = 1 + 0.5t_i$	$\hat{f}(\mathbf{x}^{(i)})$	$g(t_i)$
1	0.1	1.05	7.3025	7.3250
2	0.2	1.10	7.6100	7.6500
3	0.3	1.15	7.9225	7.9750
4	0.4	1.20	8.2400	8.3000
5	0.5	1.25	8.5625	8.6250
6	0.6	1.30	8.8900	8.9500
7	0.7	1.35	9.2225	9.2750
8	0.8	1.40	9.5600	9.6000
9	0.9	1.45	9.9025	9.9250

e) **Implementation:** Write three functions in either R or Python:

- A function `dME(f, x, j)` that computes a central difference approximation of the derivative marginal effect for feature x_j at input x.
- A function `fME(f, x, h)` that computes the forward marginal effect using a vector of stepsizes h.
- A function `NLM(f, x, h, t_vals)` that computes the Non-Linearity Measure as described in d) where `t_vals` are the grid points used to estimate the NLM.

Use them to confirm the results from sub-tasks c) and d).