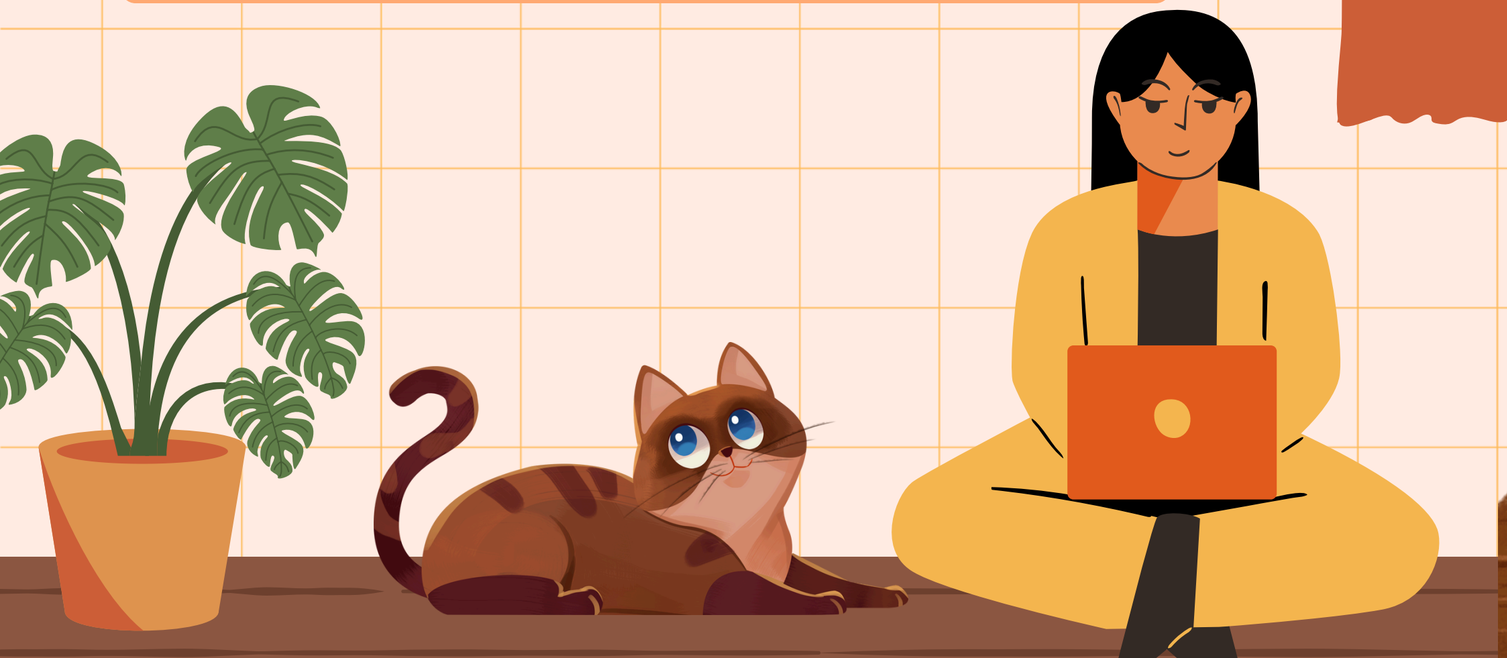
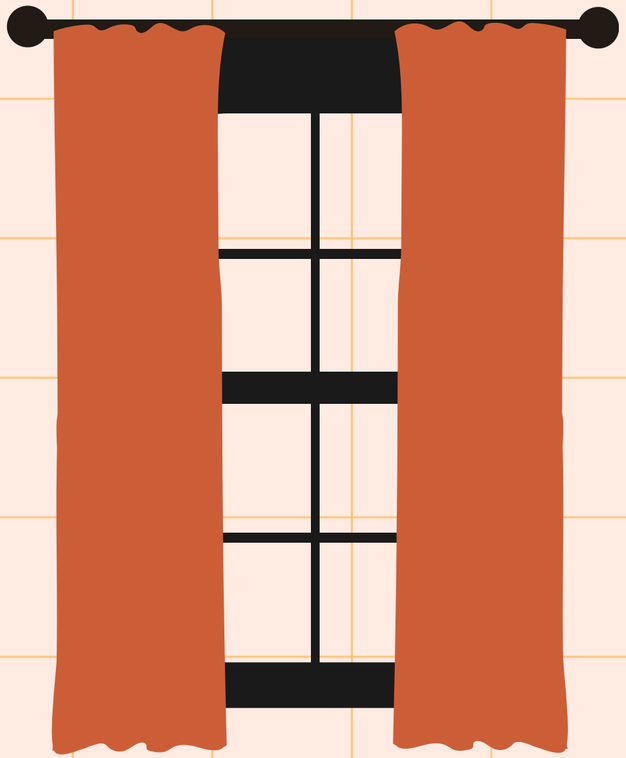
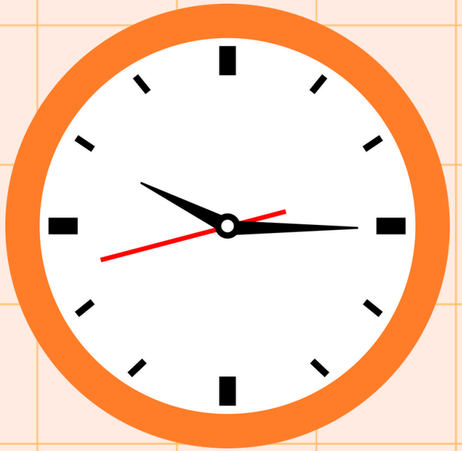
CHILDREN'S AGGRESSION

Multiple Linear Regression

Table of contents







Data Collection

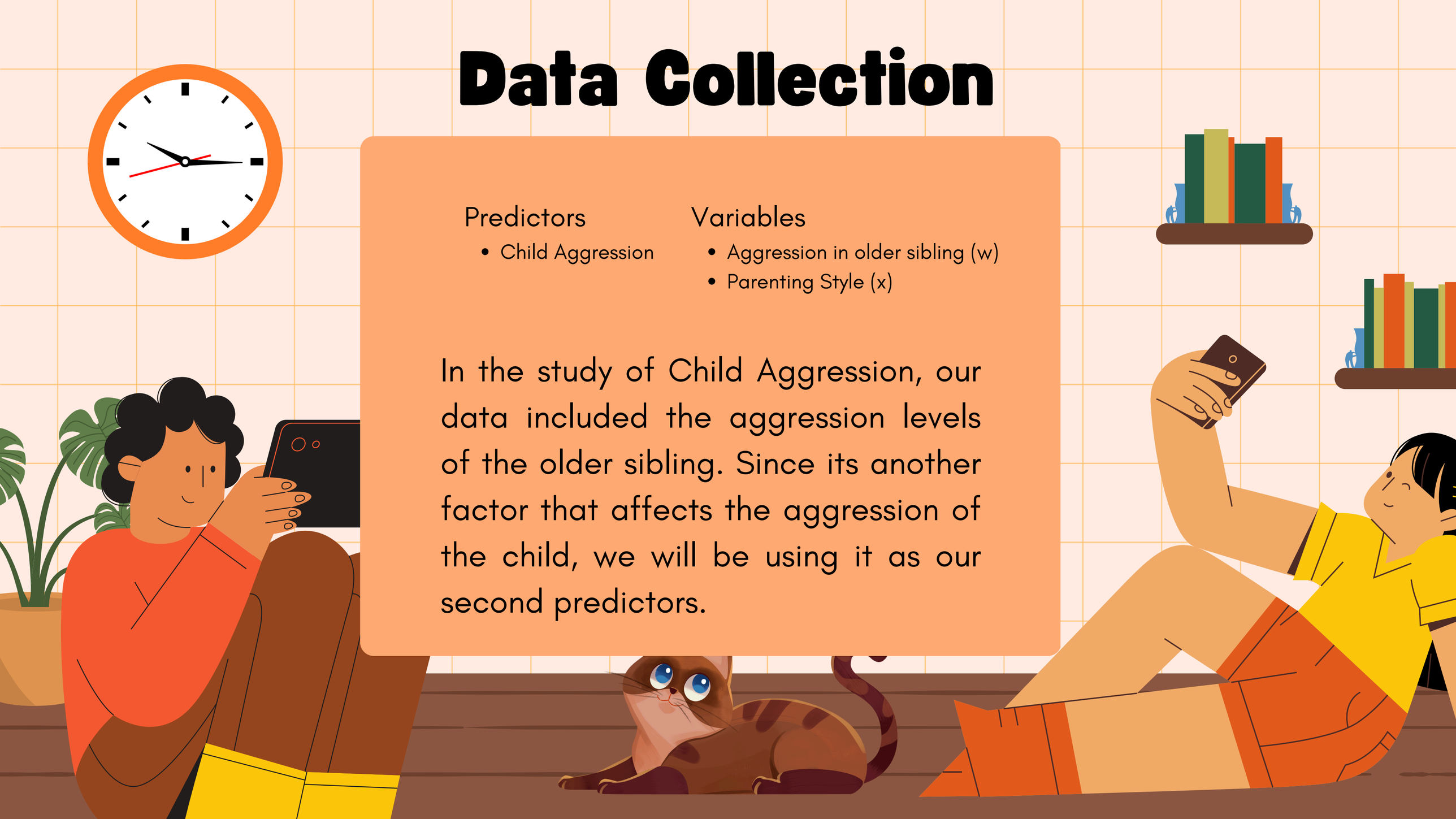
Predictors

- Child Aggression

Variables

- Aggression in older sibling (w)
- Parenting Style (x)

In the study of Child Aggression, our data included the aggression levels of the older sibling. Since its another factor that affects the aggression of the child, we will be using it as our second predictors.



Error Function

$$E(b) = \frac{1}{666}\begin{bmatrix} (y-(a+bx_1+cw_1))^2+ \\ (y-(a+bx_2+cw_2))^2+ \\ (y-(a+bx_3+cw_3))^2+ \\ \vdots \\ (y-(a+bx_{664}+cw_{664}))^2+ \\ (y-(a+bx_{665}+cw_{665}))^2+ \\ (y-(a+bx_{666}+cw_{666}))^2 \end{bmatrix}$$

$$E(b) = \frac{1}{666}\begin{bmatrix} (0.374-(a+-0.279b+-0.328c))^2+ \\ (0.771-(a+-1.248b+0.577c))^2+ \\ (-0.098-(a+-0.328b+-0.217c))^2+ \\ \vdots \\ (-0.062-(a+-0.335b+0.306c))^2+ \\ (0.104-(a+-0.475b+0.094c))^2+ \\ (-0.316-(a+-1.102b+0.149c))^2 \end{bmatrix}$$

Simplified

$$E(a,b,c) = 666a^2 - 4.974 * 10^{-14}ab + 11.022ac + 6.675a + 665b^2 + 75.616bc - 89.502b + 71.069c^2 - 17.900c + 67.859$$

Error Function

Partial Derivative

$$E'(a, b, c)$$

$$\frac{\partial E}{\partial a} = 1332a - 5.684 * 10^{-14}b + 11.022c + 6.675$$

$$\frac{\partial E}{\partial b} = -5.684 * 10^{-14}a + 1330b + 75.616c - 89.502$$

$$\frac{\partial E}{\partial c} = 11.022a + 75.616b + 142.138c - 17.900$$

Code & Verification

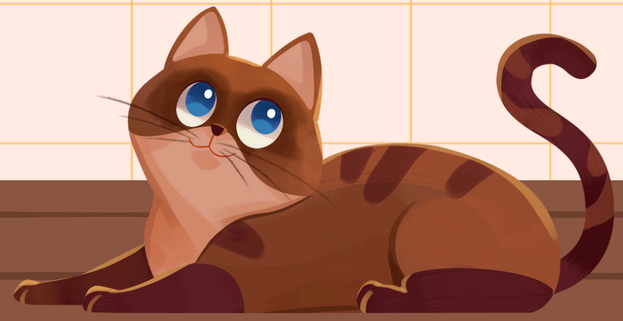
Parameters:

- feature_count = X.shape[1]
- params = [0,0,0]
- learning_rate = 0.001
- convergence_threshold = 0.00001
- max_iterations = 1000

Output:

- Number of iterations is 34
- Intercept = -0.0058
- Coefficient for (x) = 0.0620
- Coefficient for (w) = 0.0928
- Minimum error: 64.2300

```
# Convert symbolic expressions to numerical functions
grad_funcs = [sp.lambdify(symbols, gradient) for gradient in gradients]
loss_func = sp.lambdify(symbols, loss_expr)

current_loss = loss_func(*params)  # Initial loss value

for iteration in range(max_iterations):
    old_params = params.copy()
    old_loss = current_loss

    # Update parameters
    for i in range(len(params)):
        params[i] = old_params[i] - learning_rate * grad_funcs[i](*old_params)

    current_loss = loss_func(*params)
    loss_change = abs(current_loss - old_loss)
    print(f"Iteration {iteration+1}: params = {params}, loss = {current_loss}")

    if loss_change < convergence_threshold:
        break

print("Final parameters:", params)
```




Thank You