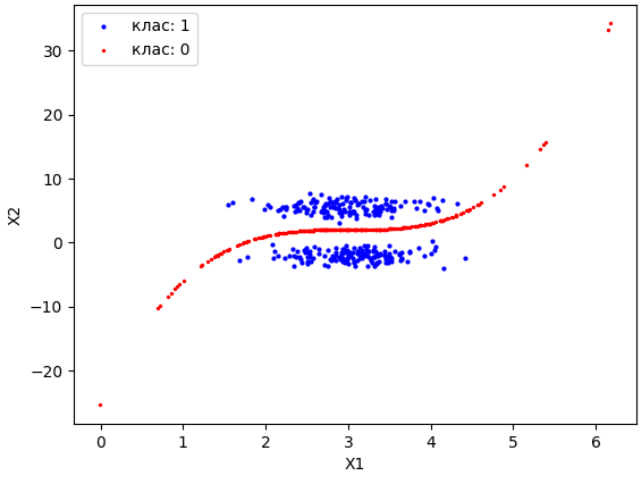
Лабораторна робота №2.

Кисляк Марини. Варіант 6.

1. Намалюємо дані розділені по класах:

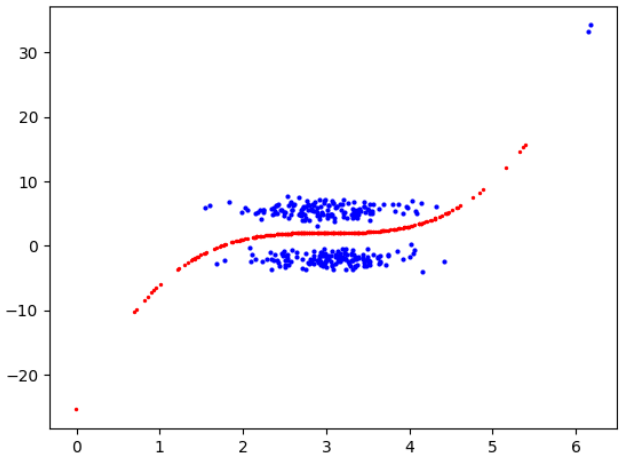


2. Реалізовуємо НМ.

```
def f(x):
    return 2/(1 + np.exp(-x)) - 1 # гіперболічний тангенс

def df(x):
    return 0.5*(1 + x)*(1 - x) # похідна

W1 = np.array([[-0.2, -0.4, 0.3], [-0.3, 0.2,0.7]]) # ваг для першого шару
W2 = np.array([0.5,0.1]) # ваги для 2 шару

def go_forward(inp):
    sum = np.dot(W1, inp)
    out = np.array([f(x) for x in sum])

    sum = np.dot(W2, out)
    y = f(sum)
    return (y, out)

def train(epoch):
    global W2, W1
    lmd = 0.01          # крок навчання
    N = 10000           # кількість ітерацій
    count = len(epoch)
    for k in range(N):
        x = epoch[np.random.randint(0, count)]  # випадковий вибір вхідного сигрналу

        y, out = go_forward(x[0:3])             # прохід по НМ

        e = y - x[-1]                           # помилка

        delta = e*df(y)                         # Градієнт

        W2[0] = W2[0] - lmd * delta * out[0]    # коригування ваґів
        W2[1] = W2[1] - lmd * delta * out[1]    # коригування ваґів

        delta2 = W2*delta*df(out)               # вектор градієнтів

        # коригування ваґів 1 шару
```

```
        W1[0, :] = W1[0, :] - np.array(x[0:3]) * delta2[0] * lmd
        W1[1, :] = W1[1, :] - np.array(x[0:3]) * delta2[1] * lmd
```

3. Після навчання, запускаємо код з визначеними вагами на наших даних і подивимось наскільки правильно модель розділяє на класи.



Як бачимо, лише два спостереження були класифіковані неправильно. Отже, можна зробити висновок, що при крокові навчання 0.01 та кількості ітерацій 10000 модель працює нормально.