STATISTICAL MACHINE LEARNING (WS2024/25)
HOMEWORK: CONFIDENCE INTERVAL

Assignment (2 points) *Assume a predictor $h\colon \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{0, 1, \ldots, Y-1\}$, is applied on i.i.d. sample $((x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, l)$.*

Given:

- *A sequence of class predictions $(h(x^1), \ldots, h(x^l))$,*
- *A sequence of true class labels $(y^1, \ldots, y^l)$,*
- *A loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$,*
- *Ann error level $\delta \in [0, 1]$,*

Your tasks is to compute a $(1-\delta)$-confidence interval (R_L, R_R), such that the probability $\mathbb{P}(R(h) \in (R_L, R_R))$ is at least $(1-\delta)$. Here, $R(h) = \mathbb{E}[\ell(y, h(x)]$ represents the expected risk of the predictor.

You have to fill in code for the following Python function:

```
def confidence_interval(true_y, pred_y, loss, delta ):
    # Input:
    #  true_y is np.array(l,dtype=int) representing the true class labels;
    #   a label is an integer from 0 to Y-1
    #  pred_y is np.array(l,dtype=int) representing the predicted class labels;
    #   a label is an integer from 0 to Y-1
    #  Loss is np.array((Y,Y)) whose Loss[y,yy] represents  the loss
    #   incurred when the true label is y and prediction is yy
    #  delta is a scalar from (0,1) representing the probability of failure
    #
    # Output:
    #  R_L, R_R are scalars such that R_L<=R_R

    return R_L, R_R
```