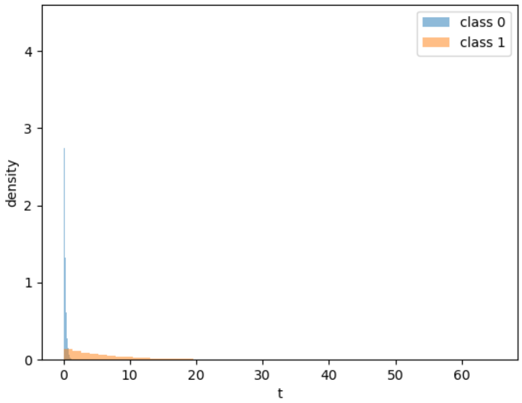
Programming Assignment 2: Binary Classification

Due October 2, 2023

The purpose of this programming assignment is to connect mathematical computation for binary classification, as performed in worksheet 3, with code, and to visualize densities and metrics graphically. You will be working with class data generated from some exponential distributions.[1]

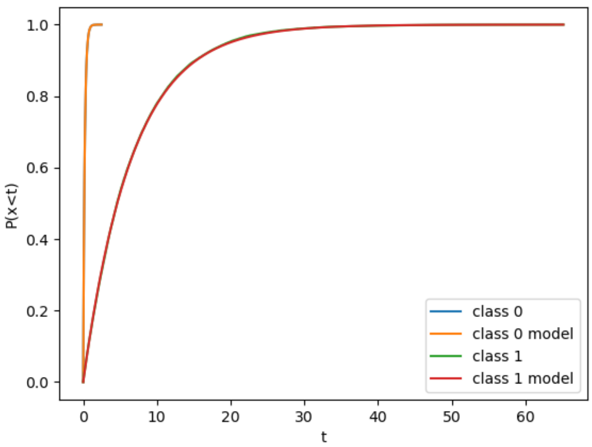


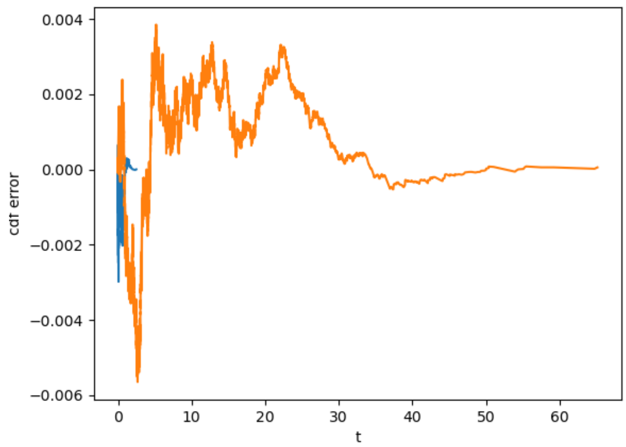
Let $(\mathcal{X} \subset \mathbb{R}, \mathbb{P}_{\mathcal{X}})$ be a random variable and suppose its cdf $F(t) := \mathbb{P}_{\mathcal{X}}(x \leq t)$ is strictly increasing. You will start by generating data $x_1, \ldots, x_m \sim_{\text{iid}} \mathbb{P}_{\mathcal{X}}$ as follows:

1. generate data $u_1, \ldots, u_m \sim_{\text{iid}} \mathbb{P}_{\mathcal{U}}$ from a uniform measure on $[0, 1]$.
2. Define $x_j := F^{-1}(u_j)$.

Sample data from exponential distributions, defining your own sampler, supposing that you can generate iid data from a uniform measure (using, e.g., 'np.random.random(n)').

Then plot its empirical cdf as well as fit a model, supposing—as it is—that your data is exponential distributed. You do not need to use any sophisticated optimization for this task: a simple parameter search will do. Still you will need to define a cost function to evaluate performance of your model and then minimize (again, via search) it over your parameter space. The parameters you find should be pretty close to those in the constructor. Reproduce the following two plots.



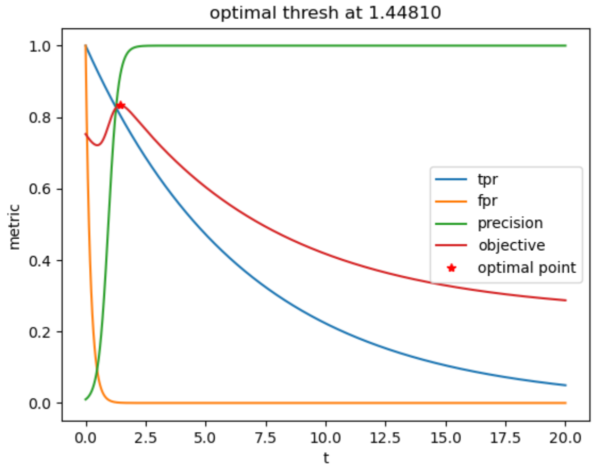


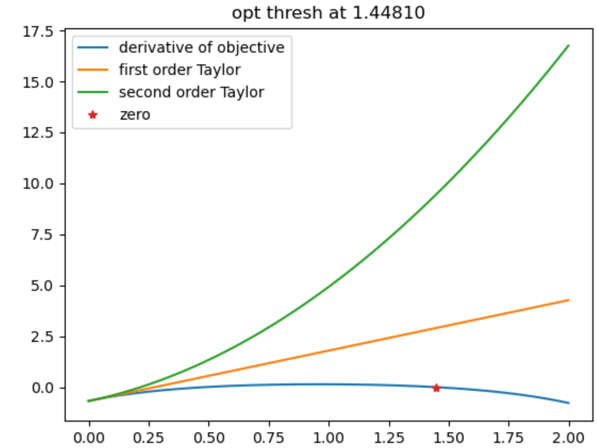
Define methods to compute various metrics (tpr, fpr, precision) and as well as objective function which is weighted sum of tpr and precision, namely $f(t) := \lambda \mathrm{tpr}(t) + (1 - \lambda)\mathrm{prec}(t)$. In our case, this objective is

$$f(t) = \lambda e^{-\gamma_1 t} + \frac{\alpha_1 e^{-\gamma_1 t}}{\alpha_0 e^{-\gamma_0 t} + \alpha_1 e^{-\gamma_1 t}}. \tag{1}$$

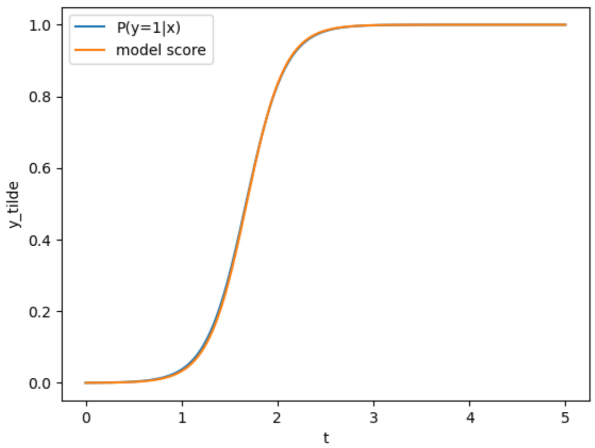
[1] Data belonging to class $y = 0$ and to class $y = 1$ will be separately defined by parameters γ_0 and γ_1, respectively.

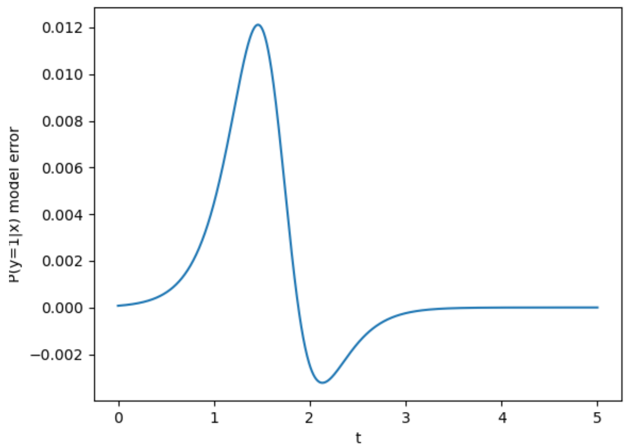
Plot precision, tpr, fpr, and the objective against threshold t. Indicate where f realizes its max. Also plot the derivative of f, along with its zero, which should match the max you found in the previous plot. You may, but do not need to, include the 1st and 2nd order Taylor expansions, which may help elucidate a problem from the previous worksheet.





Finally, plot $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1|x)$ and notice that this plot looks sigmoidal. Therefore, also fit model $\sigma_{a,b}(t) := \frac{1}{1+e^{-at+b}}$ to $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1|x)$ and plot. You should include a plot of the error.





With your submission, which should be a pdf reproducing each graphic in this document, also include in the comments on the Canvas submission page the following data, as an ordered tuple: (a, b, t^*), where t^* is the optimal threshold above (your t^* will *not* match the one above).

As typical, delete the main function in your code. You are free to play with parameters γ_y, α_y ($\mathbb{P}_{\mathcal{Y}}(y)$), and λ, but your answers in Canvas should correspond to those for the pre-loaded params.