Report

- Pre-Process:
  - Before we started to learn the model, we needed to modify the sex column.
    that is, because the algorithm cannot handle categorical attribute.
    we considered two different techniques:
    - Converting the data to sequential numbers.
    - Using dummy columns.

    Cons of the former is that using different numbers for separating between examples creates inner order between M, F and I while we didn't get any information about such order.
    because of that we preferred using dummy columns and implemented it by OneHotEncoder method.
    In that method, we converted the sex column to three columns – 'M', 'F' and 'I' and labeled them with binary notations for detecting if the example is M or not, F or not and I or not.
  - We concatenated a column of 1's to the train data to include the bias in the matrix multiplication instead of adding it externally.

- Normalize:
  - We tried to normalize the data by both min-max and zscore methods.
    the zscore normalize got us performed better by 15% so and therefore sticked with that method.

- Hyper-Parameters:
  - The hyper parameters are η in Perceptron, η and λ in SVM and τ in PA.
  - For detecting good hyper parameters, we ran the models with plenty of permutations until we got the exact values we used.
    we were inspired by grid search algorithm for doing so.

- Implementation:
  - Our code includes three phases:
    - Pre-processing.
    - Modeling.
    - Checking (in our case, printing the predictions).
  - Pre-processing:
    - Encoding the sex column.
    - Adding 1's column.
  - Modeling:
    - As perceptron is a degenerative case of SVM, we used one update function for either of them.
      when we wanted to model perceptron, we sent $\lambda = 0$.
    - PA has its own updating function.
  - Checking:
    - Printing the predicted Y labels.