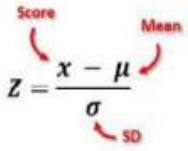
Using same example in lab 03 (about KNN) and modify the code

- Define which K-value will be the best to split the data when using cross validation (**determine the number of K folds**)
- Assign value for the K from 5 to 25 for the same problem and report the **accuracy for each k-value**
- Define **which k value** will outperform other values.

- Before fed the training data directly to the classifier, apply pre-processing for your data
  - To do that, apply zero-Score which illustrated earlier in this lab
    - Calculate the **mean** value for the feature
      - ❖ You can use built in method
    - Calculate the **Standard deviation** value for the feature (first, calculate the variance)
      - ❖ You **cannot use built in method**
    - For each x, subtract from mean value, and divide over standard deviation



    - Then apply again the assignment 2 solution to report which k fold will be the best after previous steps.

Select your own-dataset and do the following (Binary classification Task)

- According to your understanding of the concepts of **Recall** and **Precision** and the trade-off between them

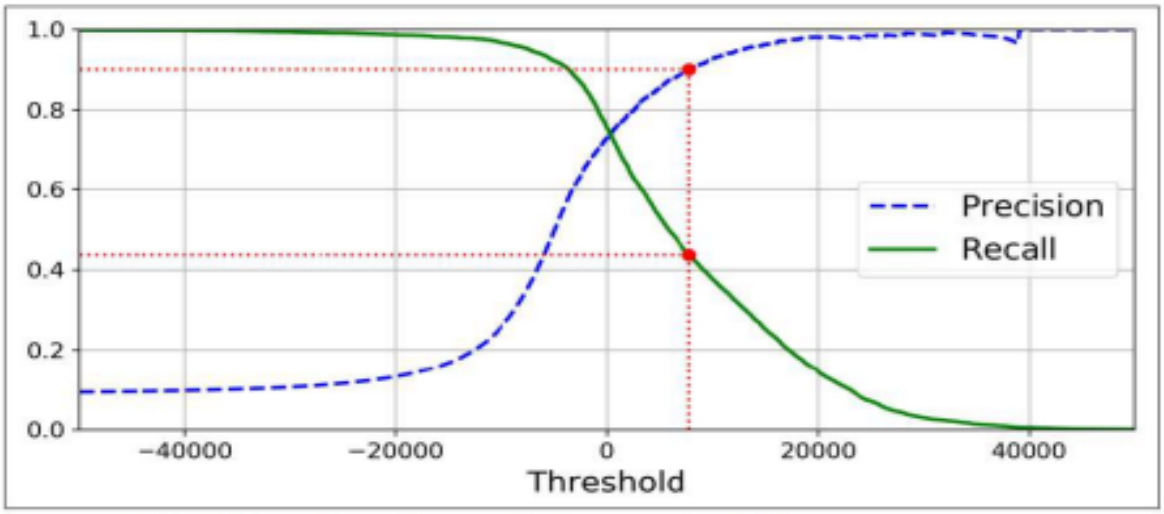


Figure 3-4. Precision and recall versus the decision threshold

1- Apply **stochastic gradient classifiers**(SGD), and report the
   - Accuracy
   - Recall
   - Precision
   - Specificity
2- Plot precision and recall as functions of the threshold value
3- After that, from the curve, define the best threshold value ($\theta 1$) that makes Recall twice Precision
4- Using this **threshold value ($\boldsymbol{\theta 1}$),** and perform the classification process using the new threshold-value $\boldsymbol{\theta 1}$
5- again, from the curve, define the best threshold value ($\theta 2$) that makes Precision twice Recall
6- Using this **threshold value ($\boldsymbol{\theta 2}$),** and perform the classification process using the new threshold-value $\boldsymbol{\theta 2}$