CSE 5523: Machine Learning - Homework #1

Due: 11:59pm 02/13/2024

Homework Policy: Please submit your solutions in a single PDF file named HW_1_name.number.pdf (e.g., HW_1_zhang.12807.pdf) to Carmen. You may write your solutions on paper and scan it, or directly type your solutions and save them as a PDF file. *Submission in any other format will not be graded.* Working in groups is fine, but each member must submit their own writeup. Please write the members of your group on your solutions. **For coding problems, please append your code to your submission and report your results (values, plots, etc.) in your written solution**. You will lose points if you only include them in your code submissions.

1) **Nearest neighbor & distance metrics [10 points].** Given four labeled training data instances, $([5,2]^\top$, "square"), $([4,0.5]^\top$, "cross"), $([6,4]^\top$, "circle"), $([4,5]^\top$, "diamond"), where each of them is a point in the 2D space (i.e., $\mathbb{R}^2$) and labeled by a "shape", please determine the label (i.e., "shape") of the test data instance $[3,2]^\top$ based on the nearest neighbor rule. See Figure 1 for illustration.

(a) [2 points] What is the label if L_1 distance is used?

$\|[3,2]^T-[5,2]^T\|_1 = |3-5|+|2-2| = 2$

$\|[3,2]^T-[4,0.5]^T\|_1 = |3-4|+|2-0.5| = 2.5$

$\|[3,2]^T-[6,4]^T\|_1 = |3-6|+|2-4| = 5$

$\|[3,2]^T-[4,5]^T\| = |3-4|+|2-5| = 4$

Square

(b) [2 points] What is the label if L_2 distance is used?

$\|[3,2]^T-[5,2]^T\|_2 = \sqrt{(3-5)^2+(2-2)^2} = 2$

$\|[3,2]^T-[4,0.5]^T\|_2 = \sqrt{(3-4)^2+(2-0.5)^2} = 1.803$

$\|[3,2]^T-[6,4]^T\|_2 = \sqrt{(3-6)^2+(2-4)^2} = 3.606$

$\|[3,2]^T-[4,5]^T\|_2 = \sqrt{(3-4)^2+(2-5)^2} = 3.162$

Cross

(c) [2 points] What is the label if L_∞ distance is used? L_∞ norm of one vector is defined as the largest magnitude among each element of a vector.

$\|[3,2]^T-[5,2]^T\|_\infty = (|3-5|^\infty + |2-2|^\infty)^{\frac{1}{\infty}} = (|2|^\infty)^{\frac{1}{\infty}} = 2$ CROSS

$\|[3,2]^T-[4,0.5]^T\|_\infty = (|3-4|^\infty + |2-0.5|^\infty)^{\frac{1}{\infty}} = (|1.5|^\infty)^{\frac{1}{\infty}} = 1.5$

$\|[3,2]^T-[6,4]^T\|_\infty = (|3-6|^\infty + |2-4|^\infty)^{\frac{1}{\infty}} = (|3|^\infty)^{\frac{1}{\infty}} = 3$

$\|[3,2]^T-[4,5]^T\|_\infty = (|3-4|^\infty + |2-5|^\infty)^{\frac{1}{\infty}} = (|3|^\infty)^{\frac{1}{\infty}} = 3$

(d) [2 points] What is the label if cosine distance is used? Given two vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, their cosine distance is defined as $1 - \frac{\boldsymbol{x}^\top \boldsymbol{x}'}{\|\boldsymbol{x}\|_2 \|\boldsymbol{x}'\|_2}$.

$\|[3,2]^T-[5,2]^T\|_{\cos} = 1 - \frac{[3\ 2][5\ 2]^T}{\sqrt{3^2+2^2}\sqrt{5^2+2^2}} = 0.0215$

$\|[3,2]^T-[4,0.5]^T\|_{\cos} = 1 - \frac{[3\ 2][4\ 0.5]^T}{\sqrt{3^2+2^2}\sqrt{4^2+0.5^2}} = 0.106$ circle

$\|[3,2]^T-[6,4]^T\|_{\cos} = 1 - \frac{[3\ 2][6\ 4]^T}{\sqrt{3^2+2^2}\sqrt{6^2+4^2}} = 0$

$\|[3,2]^T-[4,5]^T\|_{\cos} = 1 - \frac{[3\ 2][4\ 5]^T}{\sqrt{3^2+2^2}\sqrt{4^2+5^2}} = 0.047$

(e) [2 points] What is the label if the following Mahalanobis distance is used? Given two vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, their Mahalanobis distance is defined as $(\boldsymbol{x}-\boldsymbol{x}')^\top \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}')$. Here, please consider $\boldsymbol{A}$ as $\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}$.

$\|[3,2]^T-[5,2]^T\|_{\text{Mahalanobis}} = [3-5\ \ 2-2]\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 3-5 \\ 2-2 \end{bmatrix} = 81$ diamond

$\|[3,2]^T-[4,0.5]^T\|_{\text{Mahalanobis}} = [3-4\ \ 2-0.5]\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 3-4 \\ 2-0.5 \end{bmatrix} = 20.25$

$\|[3,2]^T-[6,4]^T\|_{\text{Mahalanobis}} = [3-6\ \ 2-4]\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 3-6 \\ 2-4 \end{bmatrix} = 49$

$\|[3,2]^T-[4,5]^T\|_{\text{Mahalanobis}} = [3-4\ \ 2-5]\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 3-4 \\ 2-5 \end{bmatrix} = 0$

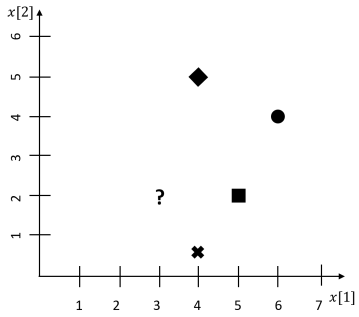


Figure 1: A training set with four labeled data instances and a test data instance (i.e., the question mark).

2) **Maximum Likelihood Estimation [10 points].** Consider a random variable $\mathbf{X}$ (possibly a vector) whose distribution may be written as $f(\mathbf{x};\theta)$, where θ is called the parameter. *Maximum likelihood estimation* is one of the most important parameter estimation techniques. Let $\mathbf{X_1}, ..., \mathbf{X_n}$ be iid (independent and identically distributed) random variables distributed according to $f(\mathbf{x};\theta)$. By independence, the joint distribution of the observations is the product

$$\prod_{i=1}^{n} f(\mathbf{X_i};\theta) \tag{1}$$

Viewed as a function of θ, this quantity is called the likelihood of θ. It is often more convenient to work with the *log-likelihood*,

$$\sum_{i=1}^{n} \log f(\mathbf{X_i};\theta) \tag{2}$$

A maximum likelihood estimate (MLE) of θ is any parameter

$$P(X;\lambda) = \frac{e^{-\lambda}\lambda^{X}}{X!}$$

$$\hat{\theta} \in \arg\max_{\theta} \sum_{i=1}^{n} \log f(\mathbf{X_i};\theta) \tag{3}$$

where "arg max" denotes the set of all values achieving the maximum. If there is a unique maximizer, it is called the maximum likelihood estimate. Let $\mathbf{X_1}, ..., \mathbf{X_n}$ be iid Poisson random variables with intensity parameter λ. Determine the maximum likelihood estimator of λ.

$$\lambda \in \arg\max_{\lambda} \sum_{i=1}^{n} \log f(X_i;\lambda) = \arg\max_{\lambda} \sum_{i=1}^{n} \log \frac{e^{-\lambda}\lambda^{X_i}}{X_i!}$$

$$= \arg\max_{\lambda} \sum_{i=1}^{n} (\log e^{-\lambda}\lambda^{X_i} - \log X_i!)$$

$$= \arg\max_{\lambda} \sum_{i=1}^{n} (\log e^{-\lambda} + \log \lambda^{X_i} - \log X_i!)$$

$$= \arg\max_{\lambda} \sum_{i=1}^{n} (-\lambda + X_i \log\lambda - \log X_i!)$$

$$= \arg\max_{\lambda} \sum_{i=1}^{n} \frac{\partial}{\partial\lambda}(-\lambda + X_i \log\lambda - \log X_i!)$$

$$= \arg\max_{\lambda} \sum_{i=1}^{n} \left(-1 + \frac{X_i}{\lambda}\right)$$

$$\sum_{i=1}^{n}\left(\frac{X_i}{\lambda} - 1\right) = 0$$

$$\sum_{i=1}^{n} \frac{X_i}{\lambda} = \sum_{i=1}^{n} 1 \Rightarrow \sum_{i=1}^{n} \frac{X_i}{\lambda} = n$$

$$\lambda = \frac{1}{n}\sum_{i=1}^{n} X_i$$

3) **Linear regression with input dependent noise [20 points].**

From the probabilistic perspective of linear regression, we have a model in which the output is linearly dependent on the input with respect to the parameters.

$$y = \boldsymbol{w}^\top \boldsymbol{x} + \epsilon \tag{4}$$

We will see in the class if the noise is from a normal distribution (i.e., $\epsilon \sim \mathcal{N}(0, \sigma^2)$), maximizing likelihood estimation leads to the sum of squares loss (or residual sum of square loss) formulation.
Here, we will consider a simple case where y and x are scalars (i.e., $x, y \in \mathbb{R}$) but the noise is dependent on the input such that $\epsilon \sim \mathcal{N}(0, \sigma^2 x^2)$. That is, the standard deviation scales linearly with the input. We observe N independent training examples denoted as $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^N$. The model is

$$y = wx + \epsilon.$$

Let's assume x is IID sampled from a known distribution with a probability density $p(x)$.

(a) [5 points] Write an expression for the likelihood of data $p(D_{\text{tr}}; w)$ in terms of $p(x)$, σ, x_i, y_i, w, N, and π.

$Y = W^T X$

$P(Y|X) \sim N(W^T X, \sigma^2)$

$Y = W^T X + \varepsilon$

$\varepsilon \sim N(0, \sigma^2 X^2)$

$P(Y|X) \sim N(W^T X, \sigma^2 X^2)$

$$P(Y|X;W) = \frac{1}{\sqrt{2\pi\sigma^2 X^2}} e^{-\frac{(Y - W^T X)^2}{2\sigma^2 X^2}}$$

$$P(D_{tr}; W) = \prod_{i=1}^{N} P(X_i, y_i; W)$$

$$= \prod_{i=1}^{N} P(X_i) P(y_i | X_i; W)$$

$$= \sum_{i=1}^{N} \log P(X_i) P(y_i | X_i; W)$$

$$= \sum_{i=1}^{N} P(X_i) \log \frac{1}{\sqrt{2\pi\sigma^2 X_i^2}} e^{-\frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2}}$$

(b) [5 points] Find the maximum likelihood estimate w^{ML} of weights w.

$$= \alpha \sum_{i=1}^{N} \left[\log \frac{1}{\sqrt{2\pi\sigma^2 X_i^2}} + \log e^{-\frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2}} \right]$$

$$= \alpha \sum_{i=1}^{N} \left[\log \frac{1}{\sqrt{2\pi\sigma^2 X_i^2}} - \frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2} \right]$$

$$\underset{W}{\text{argMax}}\, \alpha \sum_{i=1}^{N} \left[\log \frac{1}{\sqrt{2\pi\sigma^2 X_i^2}} - \frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2} \right]$$

$$= \underset{W}{\text{argMin}} \sum_{i=1}^{N} \frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2}$$

$$\frac{\partial}{\partial W} \left[\frac{(y_i - W^T X_i)^2}{2\sigma^2 X_i^2} \right] = 0$$

$$= \frac{-2(y_i - W^T X_i)}{2\sigma^2 X_i^2} X_i = 0$$

$$= \sum_{i=1}^{N} \frac{-y_i}{\sigma^2 X_i} + \sum_{i=1}^{N} \frac{W^T X_i}{\sigma^2 X_i}$$

$$W^T \sum_{i=1}^{N} \frac{1}{\sigma^2} = \sum_{i=1}^{N} \frac{y_i}{\sigma^2 X_i}$$

$$W^T = \sum_{i=1}^{N} \frac{y_i}{\sigma^2 X_i} \frac{\sigma^2}{N} = \frac{\sum_{i=1}^{N} \frac{y_i}{X_i}}{N}$$

(c) **[5 points]** Assuming that the prior distribution of w is $\mathcal{N}(0, \alpha^2)$, express the posterior distribution of w, i.e., $p(w|D_{\text{tr}})$ in terms of $p(x)$, σ, α, x_i, y_i, w, N, and π.

$$P(D_{tr}|w) = P(x_i) \frac{1}{\sqrt{2\pi\sigma^2 x_i^2}} e^{-\frac{(y_i - w^T x_i)^2}{2\sigma^2 x_i^2}} \qquad P(w) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(w-0)^2}{2\sigma^2}}$$

$$P(w|D_{tr}) = \frac{P(D_{tr}|w)P(w)}{P(D_{tr})}$$

$$= \frac{\prod_{i=1}^{N} \left(\frac{1}{\sqrt{2\pi\sigma^2 x_i^2}} e^{-\frac{(y_i - w x_i)^2}{2\sigma^2 x_i^2}} \right) \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{w^2}{2\sigma^2}} \right) P(x_i)}{P(D_{tr})}$$

(d) **[5 points]** Find the MAP estimate w^{MAP} of weights w.

$$\log P(w|D_{tr})$$

$$= \sum_{i=1}^{N} \log \left(\frac{1}{\sqrt{2\pi\sigma^2 x_i}} e^{-\frac{(y_i - w x_i)^2}{2\sigma^2 x_i^2}} \right) \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{w^2}{2\sigma^2}} \right)$$

$$= \sum_{i=1}^{N} \left[\log \frac{1}{\sqrt{2\pi\sigma^2 x_i}} - \frac{(y_i - w x_i)^2}{2\sigma^2 x_i^2} + \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{w^2}{2\sigma^2} \right]$$

$$\arg\max_w \log P(w|D_{tr}) = \arg\min_w \left[\frac{(y_i - w x_i)^2}{2\sigma^2 x_i^2} + \frac{w^2}{2\sigma^2} \right]$$

$$\frac{\partial}{\partial w} \left[\frac{(y_i - w x_i)^2}{2\sigma^2 x_i^2} + \frac{w^2}{2\sigma^2} \right] = 0$$

$$= \frac{2(y_i - w^T x_i)}{2\sigma^2 x_i^2} x_i + \frac{2w}{2\sigma^2}$$

$$= \sum_{i=1}^{N} \frac{-y_i}{\sigma^2 x_i} + \sum_{i=1}^{N} \frac{w^T x_i}{\sigma^2 x_i} + \frac{w}{\sigma^2}$$

$$w \sum_{i=1}^{N} \frac{1}{\sigma^2} + w \frac{1}{\sigma^2} = \sum_{i=1}^{N} \frac{y_i}{x_i \sigma^2}$$

$$w \left(\frac{N}{\sigma^2} + \frac{1}{\sigma^2} \right) = \sum_{i=1}^{N} \frac{y_i}{x_i \sigma^2}$$

$$w = \frac{\sum_{i=1}^{N} \frac{y_i}{x_i}}{1 + N}$$

4) **Weighted Linear Regression [10 points].** Consider a linear regression problem in which we want to weigh different training examples differently. Specifically, suppose we want to minimize

$$E_D(w) = \frac{1}{2}\sum_{i=1}^{N} r_i(w^\top x_i - y_i)^2. \tag{5}$$

In class, we worked out what happens for the case where all the weights (r_i's) are one. In this problem, we will generalize some of those ideas to the weighted setting. In other words, we will allow the weights r_i to be different for each of the training examples.

Suppose we have a training set $\{(x_i, y_i); i = 1, \cdots, N\}$ of N independent examples, but in which the y_i's were observed with different variances. Specifically, suppose that

$$p(y_i|x_i; w) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\Big(-\frac{(y_i - w^T x_i)^2}{2(\sigma_i)^2}\Big). \tag{6}$$

In other words, y_i has mean $w^T x_i$ and variance $(\sigma_i)^2$ (where the σ_i's are fixed, **known** constants). Show that finding the maximum likelihood estimate of w reduces to solving a weighted linear regression problem. State clearly what the r_i's are in terms of the σ_i's.

$$P(D|w) = \prod_{i=1}^{N} P(y_i|x_i; w) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - wx_i)^2}{2(\sigma_i)^2}}$$

$$\log P(D|w) = \sum_{i=1}^{N} \log\left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - wx_i)^2}{2(\sigma_i)^2}}\right]$$

$$\operatorname*{argmax}_w \log P(D|w) = \operatorname*{argmin}_w \sum_{i=1}^{N}\left[\frac{(y_i - wx_i)^2}{2(\sigma_i)^2}\right]$$

since we want to $\operatorname*{argmin}_w \frac{1}{2}\sum_{i=1}^{N} r_i (w^T x_i - y_i)^2$

so $r_i = \frac{1}{(\sigma_i)^2}$

5) **Naive Bayes as linear classifiers [15 points].**

Given a training set $D_{\text{tr}} = \{(\boldsymbol{x}_n \in \mathbb{R}^D, y_n \in \{+1, -1\})\}_{n=1}^N$ for binary classification, let us fit a Naive Bayes model to it; i.e., $p(y|\boldsymbol{x}) \propto p(y) \prod_{d=1}^D p(x[d]|y)$, where $p(x[d]|y)$ is a Gaussian distribution $\mathcal{N}(\mu_{d,y}, \sigma_d^2)$. That is, we assume that for each dimension d, the two one-dimensional Gaussian distributions (one for each class) share the same variance. For $p(y)$, let $p(+1) = \lambda$. Please show the followings.

(a) [6 points] $p(y|\boldsymbol{x})$ can be re-written as $\frac{1}{1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x} + b)}}$ $(y \in \{+1, -1\})$

$$P(y=1|x) = \frac{P(x|y=1)\,P(y=1)}{P(x)}$$

$$= \frac{P(x|y=1)\,P(y=1)}{P(x, y=1) + P(x, y=-1)}$$

$$= \frac{P(x|y=1)\,P(y=1)}{P(x|y=1)P(y=1) + P(x|y=-1)P(y=-1)}$$

$$= \frac{1}{1 + \frac{P(x|y=-1)P(y=-1)}{P(x|y=1)P(y=1)}}$$

$$= \frac{1}{1 + e^{\log\left[\frac{P(x|y=-1)P(y=-1)}{P(x|y=1)P(y=1)}\right]}}$$

Let $z = \log \frac{P(x|y=1)P(y=1)}{P(x|y=-1)P(y=-1)} = y(w^\top x + b)$

$$= \frac{1}{1+e^{-z}}$$

$$P(y=1|x) = \frac{1}{1+e^{-y(w^\top x+b)}}$$

(b) [6 points] What are the corresponding $\boldsymbol{w}$ and b?

Your answers should be based on $\mu_{d,y}$, σ_d, and λ. For $\boldsymbol{w} \in \mathbb{R}^D$, you may simply write the expression of $w[d]$ for a specific d.

$$y(w^\top x + b) = \log \frac{P(x|y=1)P(y=1)}{P(x|y=-1)P(y=-1)} = \log \frac{P(y=1)\prod_{d=1}^D P(x[d]|y)}{P(y=-1)\prod_{d=1}^D P(x[d]|y)} = \log\frac{\lambda}{1-\lambda} + \sum_{d=1}^D \log \frac{\frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{(x[d]-\mu_{d,y=1})^2}{2\sigma_d^2}}}{\frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{(x[d]-\mu_{d,y=-1})^2}{2\sigma_d^2}}}$$

$$= \log\frac{\lambda}{1-\lambda} + \sum_{d=1}^D \left[\log e^{-\frac{(x[d]-\mu_{d,y=1})^2}{2\sigma_d^2}} - \log e^{-\frac{(x[d]-\mu_{d,y=-1})^2}{2\sigma_d^2}}\right] = \log\frac{\lambda}{1-\lambda} + \sum_{d=1}^D \left(-\frac{(x[d]-\mu_{d,y=1})^2}{2\sigma_d^2} + \frac{(x[d]-\mu_{d,y=-1})^2}{2\sigma_d^2}\right)$$

$$= \log\frac{\lambda}{1-\lambda} + \sum_{d=1}^D \frac{x^2[d] + \mu_{d,y=-1}^2 - 2x[d]\mu_{d,y=-1} - x^2[d] - \mu_{d,y=1}^2 + 2x[d]\mu_{d,y=1}}{2\sigma_d^2}$$

$$w[d] = \sum_{d=1}^D \frac{2x[d]\mu_{d,y=1} - 2x[d]\mu_{d,y=-1}}{2\sigma_d^2} = x[d]\frac{\mu_{d,y=1} - \mu_{d,y=-1}}{\sigma^2}$$

$$b = \log\frac{\lambda}{1-\lambda} + \sum_{d=1}^D \frac{\mu_{d,y=-1}^2 - \mu_{d,y=1}^2}{2\sigma^2}$$

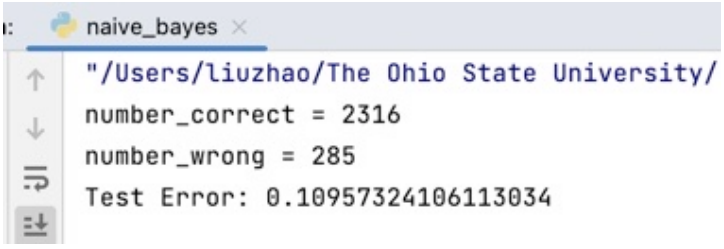
6) **Naive Bayes Classifier [35 points].**

Download the files **spambase.train** and **spambase.test**.
The file spambase.train contains 2000 training data and spambase.test has 2601 test data. Both datasets have 58 columns: the first 57 columns are input features, corresponding to different properties of an email, and the last column is an output label indicating spam (1) or non-spam (0). Please fit the Naive Bayes model using the training data.
As a pre-processing step, quantize each variable to one of two values, say 1 and 2, so that values below the median map to 1, and others map to 2. Please aggregate the training and test test to obtain the median value of each variable, and then use the median to quantize both data sets.

(a) Report the test error (misclassification percentage) of Naive Bayes classifier. As a sanity check, what would be the test error if you always predicted the same class, namely, the majority class from the training data?

Baseline Test error : 38.56 %

Naive Bayes test error : 10.957 %



```
 1 import numpy as np
 2 import pandas as pd
 3
 4 SPAM = 1
 5 NON_SPAM = 0
 6
 7
 8 if __name__ == '__main__':
 9     train = pd.read_csv("spambase.train", header=
   None)
10     test = pd.read_csv("spambase.test", header=None
   )
11
12     train = pd.concat([train, test], ignore_index=
   True, sort=False)
13
14     number_correct = 0
15     number_wrong = 0
16
17     # number of rows in Test dataset
18     test_rows = test.shape[0]
19     # number of columns in Test dataset
20     test_columns = test.shape[1]
21
22     # number of rows in Training dataset
23     train_rows = train.shape[0]
24     # number of columns in Training dataset
25     train_columns = train.shape[1]
26
27     # Training dataset with Y = 1
28     train_y_1 = train[train[57].isin([1])]
29     number_train_y_1 = train_y_1.shape[0]
30
31     # Training dataset with Y = 0
32     train_y_0 = train[train[57].isin([0])]
33     number_train_y_0 = train_y_0.shape[0]
34
35     # Probability(Y = 1)
36     probability_y_1 = number_train_y_1 / train_rows
37     # Probability(Y = 0)
38     probability_y_0 = number_train_y_0 / train_rows
```

```
39
40      medians = []
41      for column in range(train_columns - 1):
42          medians.append(train[column].median())
43          # print(f"column = {column}, median = {
    train[column].median()}")
44
45      """
46      median_condition_probability = np.array([[1, 2
    , 3]])
47
48      for column in range(train_columns - 1):
49          median = medians[column]
50
51          number_match_1 = train_y_1[train_y_1[column
    ] <= median].shape[0]
52          theta_1 = number_match_1 / number_train_y_1
53
54          number_match_0 = train_y_0[train_y_0[column
    ] <= median].shape[0]
55          theta_0 = number_match_0 / number_train_y_0
56
57          row_median_proba = np.array([[median,
    theta_1, theta_0]])
58
59          if column == 0:
60              median_condition_probability =
    row_median_proba
61          else:
62
63              median_condition_probability = np.
    concatenate((median_condition_probability,
    row_median_proba))
64      """
65      #print(median_condition_probability)
66      #number_match_1 = train_y_1[train_y_1[11] < 0.
    145000000000000002].shape[0]
67      #print(f"number_match_1 = {number_match_1}")
68      #print(f"number_train_y_1 = {number_train_y_1
    }")
69      #print(f"number_train_y_1 = {number_train_y_0
```

```
69 }")
70 
71     for row in range(test_rows):
72         predict_value = 0
73         real_value = test.iloc[row, test_columns
    - 1]
74 
75         # For spam (Y=1) which indicates label = 1
76         probability_spam = 1
77         # For Non-spam (Y=0) which indicates label
    = 0
78         probability_non_spam = 1
79 
80         for column in range(test_columns - 1):
81             new_value = test.iloc[row, column]
82 
83             median = medians[column]
84 
85             number_match_1 = train_y_1[train_y_1[
   column] <= median].shape[0]
86             theta_1 = number_match_1 /
   number_train_y_1
87 
88             number_match_0 = train_y_0[train_y_0[
   column] <= median].shape[0]
89             theta_0 = number_match_0 /
   number_train_y_0
90 
91             if new_value > median:
92                 theta_1 = 1 - theta_1
93                 theta_0 = 1 - theta_0
94 
95             probability_spam = probability_spam *
   theta_1
96             probability_non_spam =
   probability_non_spam * theta_0
97 
98         probability_spam = probability_spam *
   probability_y_1
99         probability_non_spam =
   probability_non_spam * probability_y_0
```

```
100
101         #print(f"row = {row}, probability_spam = {
    probability_spam}, probability_non_spam = {
    probability_non_spam}")
102
103         if probability_spam >=
    probability_non_spam:
104             predict_value = 1
105         else:
106             predict_value = 0
107
108         #print(f"row = {row}, real_value = {
    real_value}, predict_value = {predict_value}")
109
110         if predict_value == real_value:
111             number_correct = number_correct + 1
112         else:
113             number_wrong = number_wrong + 1
114
115     print(f"number_correct = {number_correct}")
116     print(f"number_wrong = {number_wrong}")
117
118     print(f"Test Error: {number_wrong / test_rows}
    ")
119
120
```