数据预处理

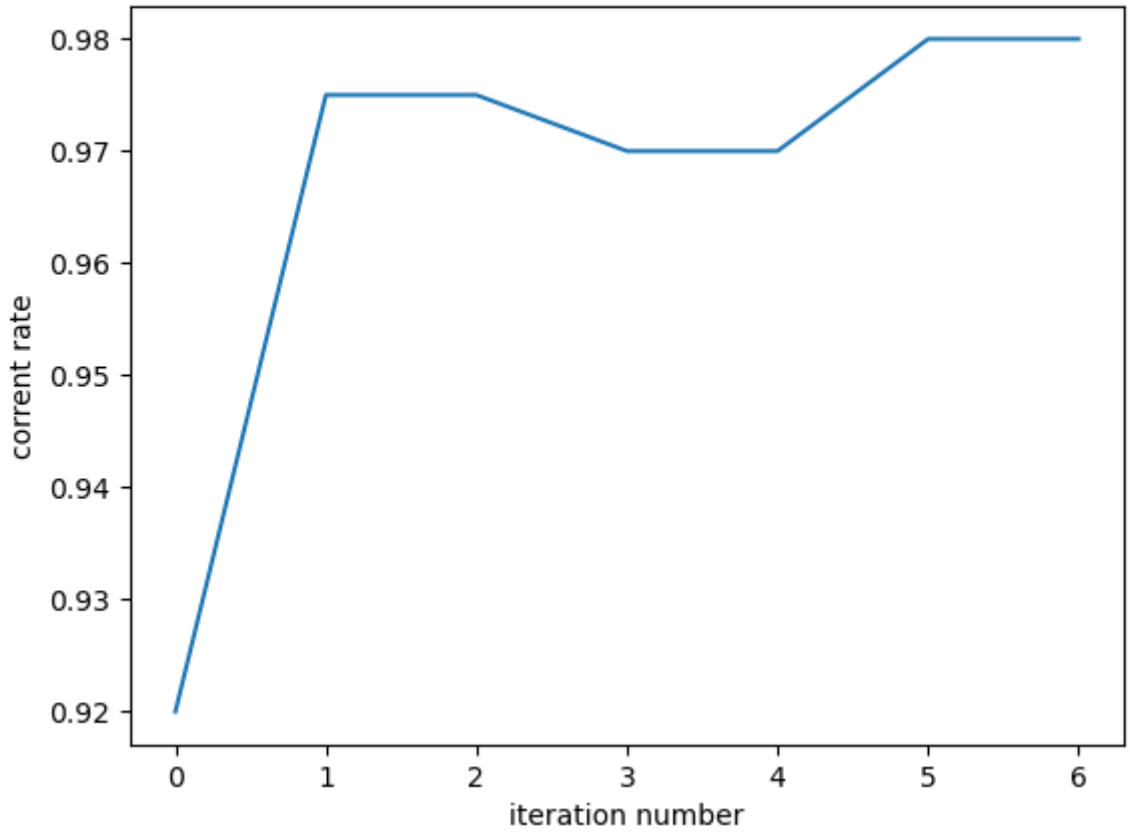
运行：
python3 ./dataProcess.py
将"./archive/train.csv"中的 price_ranage 列，按照是否小于等于1 分为0 和1。
将所有参数归一化，方便朴素贝叶斯法划定范围。
的2000个数据按照8:1:1分成trainData, validData, testData 保存在data目录下。

朴素贝叶斯

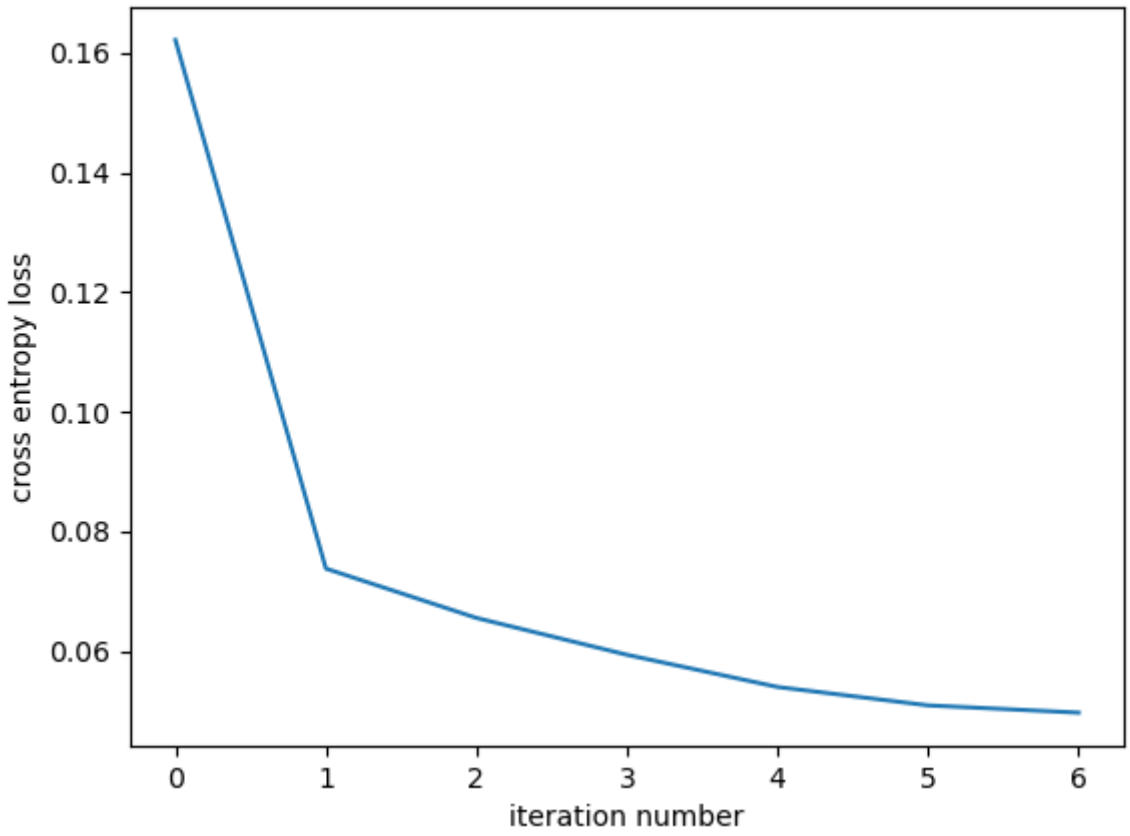
运行：
python3 ./naiveBayesian.py
对于参数 $X_1, X_2 \dots X_N$, 对应的结果 $Y_1, Y_2 \dots Y_N$ 。对X的第j项$X^{(j)}$ （1≤j≤J），可以取 $x_{j1}, x_{j2} \dots x_{jL}$ 。对于Y可以取$c_1, c_2 \dots c_K$
可以估计 $P(Y = c_k) = \frac{\Sigma_{i=1}^{N} I(y_i = c_k) + 1}{N + K}$ (加1防止结果为0)
根据假设 $P(X = x | Y = c_k) = \Pi_{j=1}^{J} P(X^{(j)} = x^{(j)} | Y = c_k)$
可以估计 $P(X = x | Y = c_k) = \Pi_{j=1}^{J} \frac{P(X^{(j)} = x^{(j)}, Y = c_k)}{P(Y = c_k)} = \Pi_{j=1}^{J} \frac{\Sigma_{i=1}^{N} I(X_i^{(j)} = x^{(j)}, Y_i = c_k) + 1}{\Sigma_{i=1}^{N} I(Y_i = c_k) + N}$ (加1防止结果为0)
则 $P(Y = c_k | X = x) \propto P(X = x | Y = c_k) P(Y = c_k)$
则 $\hat{y} = argmax_{c_k} (P(X = x | Y = c_k) P(Y = c_k))$
将X中bool值的参数分为两类，将连续的变量按0.1为间隔区分为10类。根据以上公式可以在 **trainBayesian** 中算出每个 $P(Y = c_k), P(X^{(j)} = x^{(j)} | Y = c_k) (k = 1, 2 \dots K, j = 1, 2 \dots J)$ ，并用pickle存入文件
在 **testBayesian** 计算$\hat{y}$ 并检验是否等于 y, 输出正确率

逻辑回归

运行：
python3 ./logisticRegression.py
假设结果y=1的概率符合$P(Y = 1 | x) = \frac{e^{w \cdot x + b}}{1 + e^{w \cdot x + b}}$
则似然函数为 $\Pi_{i=1}^{N} (P(Y = 1 | x_i))^{y_i} (1 - P(Y = 1 | x_i))^{1 - y_i}$
则对数似然函数为 $L(w, b) = \Sigma_{i=1}^{N} (y_i log(P(Y = 1 | x_i)) + (1 - y_i) log(1 - P(Y = 1 | x_i)))$
化简得 $L(w, b) = \Sigma_{i=1}^{N} (y_i (w \cdot x_i + b) - log(1 + e^{w \cdot x + b}))$ 。求L最大。
求偏导得
$\frac{\partial L}{\partial b} = \Sigma_{i=1}^{N} (y_i - P(Y = 1 | x))$
$\frac{\partial L}{\partial w} = \Sigma_{i=1}^{N} (y_i - P(Y = 1 | x)) \cdot x_i$
在 **trainRL**函数中每次迭代更新参数
$w \leftarrow w - speed \cdot \Sigma \frac{\partial L}{\partial w}$, $b \leftarrow b - speed \cdot \Sigma \frac{\partial L}{\partial b}$
每次将所有训练集用过一次后用验证集求交叉熵损失，直到损失小于指定ϵ。用pickle保存参数W和b


0.98
0.97
0.96
0.95
0.94
0.93
0.92
corrent rate
0
1
2
3
4
5
6
iteration number

正确率随迭代次数变化


0.16
0.14
0.12
0.10
0.08
0.06
cross entropy loss
0
1
2
3
4
5
6
iteration number

交叉熵损失随迭代次数变化
在 ***testRL** 中检验时，只需将 x带入 $P(Y=1|x)$ 结果大于0.5 则预测Y=1, 否则Y=0

SVM

运行：
python3 ./SVM.py
运用sklearn中的SVM库，参数C为松弛变量，默认为1，kernel为核函数，默认rbf，gamma为核函数参数，默认为auto

Empirical Study

利用装饰器 **timer** 为训练和测试函数计时（已经去掉读取数据和存储时间）

方法	train time (ms)	test time (ms)	accuracy
naive bayesian	42	9	93.5%
logistic regression	133	3	96%
SVM	59	11	94%

朴素贝叶斯法训练最快（如果实现时用向量化应该会更快），测试时间较长，准确率较低。
逻辑回归训练时间较长，测试时间因为计算量很小很短，准确率最高。
SVM方法因为调用了sklearn的库，所以训练时间也较快，测试时间最长，准确率介于两者之间。

518021911058 沈瑜石