实验一：逻辑回归

姓名：陆世潜　　学号：PB21020629

实验内容

补充代码框架，实现一个逻辑回归模型，并使用数据集测试模型预测的效果，具体来说是根据某人的各项特征预测其贷款状态。还探究了超参数的变化对模型表现的影响。训练集占比 90%，测试集占比 10%。

数据

Loan Data Set | Kaggle

损失函数

$$L = \frac{1}{m}\sum_{i=1}^{m}(y_i * \log \hat{y}_i + (1-y_i) * \log(1-\hat{y}_i)) + \gamma * R(\boldsymbol{\theta})$$

$$\hat{\boldsymbol{y}} = (y_1, \ldots, y_m)^T = sigmoid(\boldsymbol{X} * \boldsymbol{\theta})$$

其中m为训练集的数据量，y_i为真实结果，$\hat{y}_i$为预测结果，γ为惩罚系数，$R(\cdot)$为惩罚项（L1 或 L2 范数），$\boldsymbol{\theta}$为模型参数（其长度取决于数据特征的数量），$\boldsymbol{X}$为输入数据（包含各种被编码的特征）。

参数更新的过程（lr为学习率）：

正则项为 L1 范数：

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - lr * \left(\frac{1}{m}\boldsymbol{X}^T * (sigmoid(\boldsymbol{X} * \boldsymbol{\theta}) - \boldsymbol{y}) + \gamma * sign(\boldsymbol{\theta})\right)$$

正则项为 L2 范数：

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - lr * \left(\frac{1}{m}\boldsymbol{X}^T * (sigmoid(\boldsymbol{X} * \boldsymbol{\theta}) - \boldsymbol{y}) + \gamma * 2 * \boldsymbol{\theta}\right)$$

Loss Curve

参数：

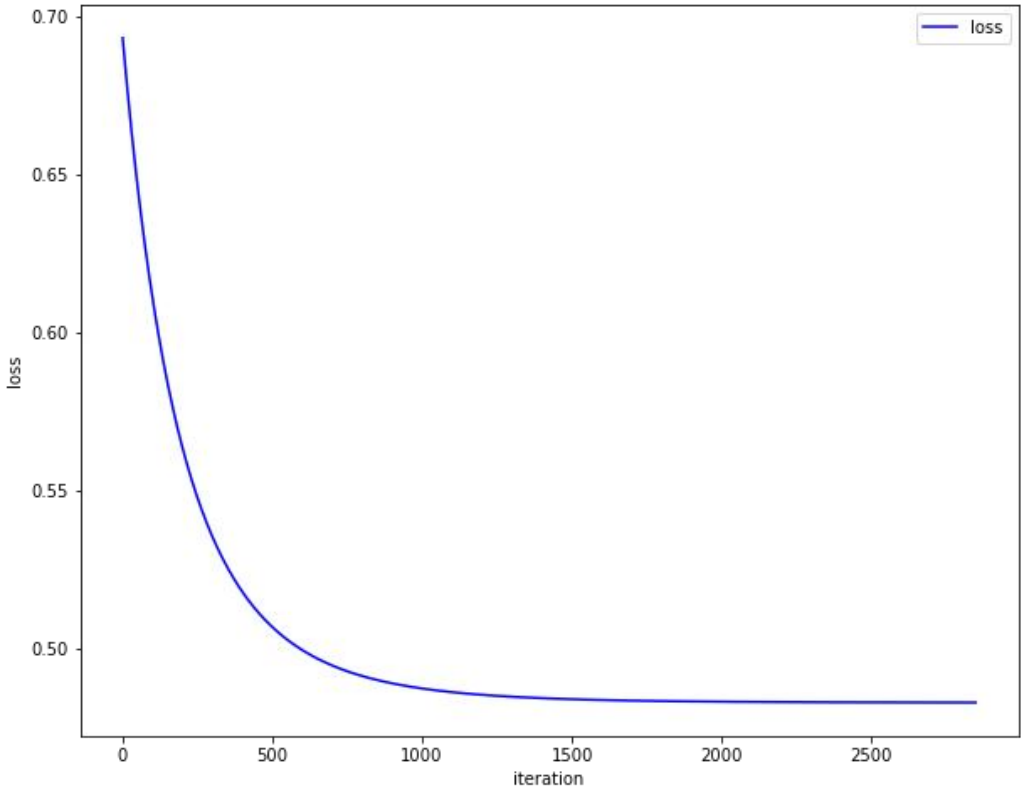
normalization method: Z-score (mean = 0, standard deviation = 1)

penalty: L2

fit intercept: True

learning rate: 0.01

gamma(penalty coefficient): 0.01



不同的实验设置以及超参数的影响

一下各项测试都是测试 10 次然后取平均值。

Normalization Methods:

Z-score 指让每列（每个特征的）数据的平均数为 0，标准差为 1。

Max-Min 指对每个数据 x，计算 x_norm = (x - min) / (max - min)。

可见 Z-score 方法效果略好一些，迭代次数也相对少很多，loss 也较小。

	accuracy	precision	recall	iteration	final_loss
Max-Min	0.811290323	0.794656707	0.983612785	7527.3	0.534795883
Z-score	0.820967742	0.806380574	0.979159544	2802.2	0.481558454

Penalty:

可见 L2 方法略好一些，但迭代次数稍多。

	accuracy	precision	recall	iter_num	final_loss
L1	0.822580645	0.803877702	0.972346045	1545.3	0.496700045
L2	0.827419355	0.812509017	0.96499218	2809.8	0.482561208

Intercept:

可见加上截距稍微好一些，迭代次数也相对少一些，loss 也较小。

	accuracy	precision	recall	iter_num	final_loss
FALSE	0.761290323	0.79143229	0.887989299	3893.6	0.533002809
TRUE	0.790322581	0.791220289	0.943881899	2858.3	0.476108347

Learning Rate:

效果上没有明显区别，学习率越高，迭代次数越少，最后的 loss 更小（不明显）。

	accuracy	precision	recall	iter_num	final_loss
0	0.324193548	—	0	1	0.693147181
0.0001	0.793548387	0.785812816	0.951542271	109757.1	0.479170414
0.001	0.79516129	0.787497213	0.951542271	18744.8	0.476243573
0.005	0.79516129	0.787497213	0.951542271	5030.8	0.475930292
0.01	0.79516129	0.787497213	0.951542271	2818.6	0.475886622
0.05	0.79516129	0.787497213	0.951542271	719.3	0.475848468
0.1	0.79516129	0.787497213	0.951542271	396.8	0.475843047
0.5	0.79516129	0.787497213	0.951542271	99	0.475838111
1	0.79516129	0.787497213	0.951542271	54.4	0.47583734
2	0.79516129	0.787497213	0.951542271	30.5	0.475836825
5	0.79516129	0.787497213	0.951542271	23	0.47583621

Gamma (penalty coefficient):

效果上没有明显区别，惩罚项系数越大，迭代次数越少，但 loss 越大。

	accuracy	precision	recall	iter_num	final_loss
0	0.796774194	0.795194716	0.956213728	4445.9	0.452547941
0.0001	0.796774194	0.795194716	0.956213728	4404.2	0.452852017
0.001	0.796774194	0.795194716	0.956213728	4094	0.455519128
0.005	0.796774194	0.795194716	0.956213728	3322	0.466199664
0.01	0.796774194	0.795194716	0.956213728	2818.6	0.477628521
0.05	0.79516129	0.794920206	0.953991506	1561.9	0.534534715
0.1	0.793548387	0.794389594	0.951427403	1103.1	0.571495285
0.5	0.796774194	0.798590599	0.949205181	369.3	0.649193832
1	0.796774194	0.799801997	0.946932454	204.6	0.668573292
2	0.798387097	0.801198101	0.946932454	107.9	0.680078397
5	0.798387097	0.801198101	0.946932454	44	0.687710609

选出最合适的参数和方法，得到最佳 accuracy = 0.8137096774193548

normalization method = Z-score; penalty = L2; fit intercept = True; lr = 0.01; γ = 0.01