華東理工大學

EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

《机器学习》

实验报告本

班　级：**软件 212**

学　号：**10180824**

姓　名：**徐壮壮**

指导教师：**戴蕾**

信息科学与工程学院

2024 年　04 月　06 日

《机器学习》实 验 报 告

实验 1 名称：逻辑回归实验	**实验地点**：信息楼 318
所使用的工具软件及环境： Python 版本：Python 3.10 及以上	
一、实验目的 ➢ 掌握回归的基本原理及 python 实现 ➢ 掌握逻辑回归的基本原理及 python 实现 ➢ 能够运用逻辑回归实现具体分类任务	
二、实验内容 针对某一具体分类问题，运用所学知识，设计逻辑回归模型并编程解决，最终需要输出分类精度和分类可视化图。	
三、实验要求 1. 具体应用领域自选； 2. 具体分类数据自选； 3. 具体分类任务（二分类或多分类）自选。	

四、实验步骤

选择二分类逻辑回归任务，数据集为学生两次测试成绩及录取与否结果，根据某个学生的两次测试评分，来决定他们是否会录取。 1 表示被录取，0 表示不被录取；

1. 读取数据集

2. 数据集预览

3. 训练集与测试集划分

4. 使用 sklearn 的逻辑回归函数来拟合

4.1 计算其准确率

4.2 绘制决策边界

5. 自己写损失函数与梯度函数

5.1 计算其准确率

5.2 绘制决策边界

6. 使用双曲线来简单拟合

6.1 计算边界

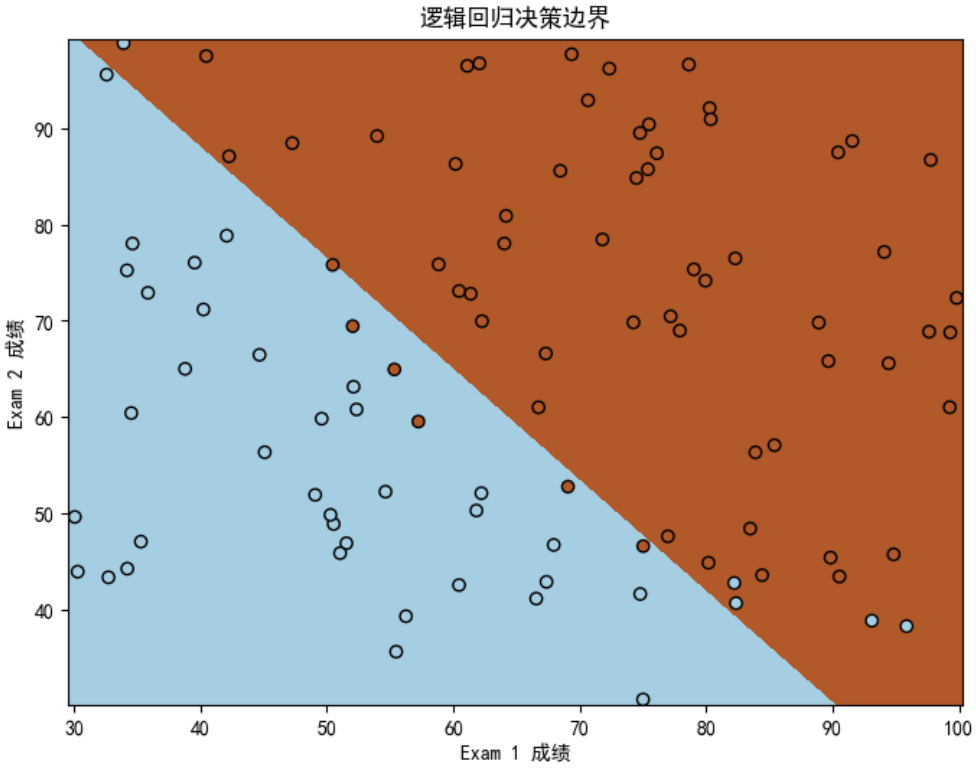
6.2 计算准确率

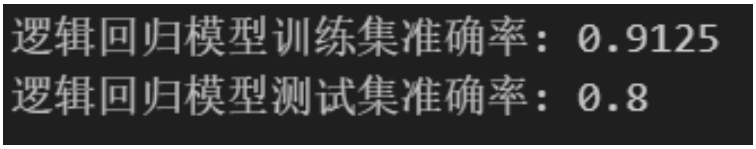
6.3 绘制决策边界

五、 程序设计的核心代码

1. sklearn 逻辑回归模型拟合

```
# 特征提取
X = data.iloc[:, :2]  # 特征矩阵(第一次和第二次测试分数)
y = data.iloc[:, 2]   # 目标向量(录取结果)

# 将数据分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 创建逻辑回归模型并进行训练
model = LogisticRegression()
model.fit(X_train, y_train)

# 在测试集上评估模型
accuracy_train = model.score(X_train, y_train)
accuracy = model.score(X_test, y_test)
print(f"逻辑回归模型训练集准确率: {accuracy_train}")
print(f"逻辑回归模型测试集准确率: {accuracy}")
```

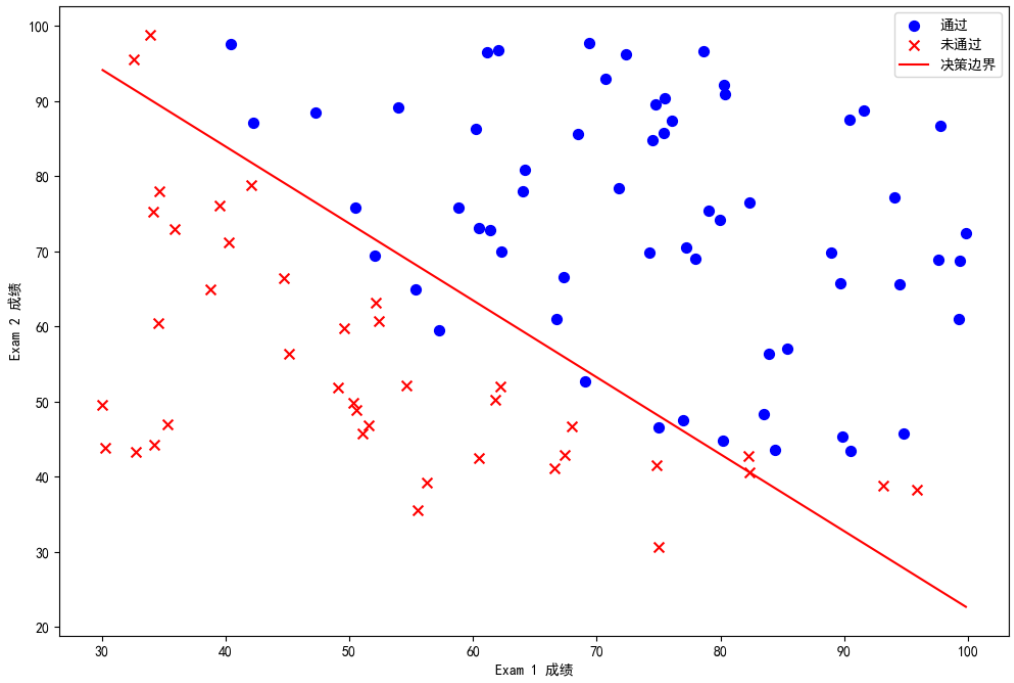




sklearn 逻辑回归模型决策边界图及准确率

2. 自己写损失函数与梯度函数

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

# 定义代价函数
def cost(w, X, y):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))
    return np.sum(first - second) / (len(X))

# 梯度下降函数
def gradient(w, X, y):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)

    parameters = int(w.ravel().shape[1])
    grad = np.zeros(parameters)
    error = sigmoid(X * w.T) - y

    for i in range(parameters):
        term = np.multiply(error, X[:, i])
        grad[i] = np.sum(term) / len(X)

    return grad


result = opt.fmin_tnc(func=cost, x0=w, fprime=gradient, args=(X, y))

# 预测函数
def predict(w, X):
    probability = sigmoid(X * w.T)
    return [1 if x >= 0.5 else 0 for x in probability]

# 计算准确率
w_min = np.matrix(result[0])
predictions = predict(w_min, X)
correct = [
    1 if ((a == 1 and b == 1) or (a == 0 and b == 0)) else 0
    for (a, b) in zip(predictions, y)
]
accuracy = (sum(map(int, correct)) % len(correct))
print('accuracy = {0}%'.format(accuracy))
```



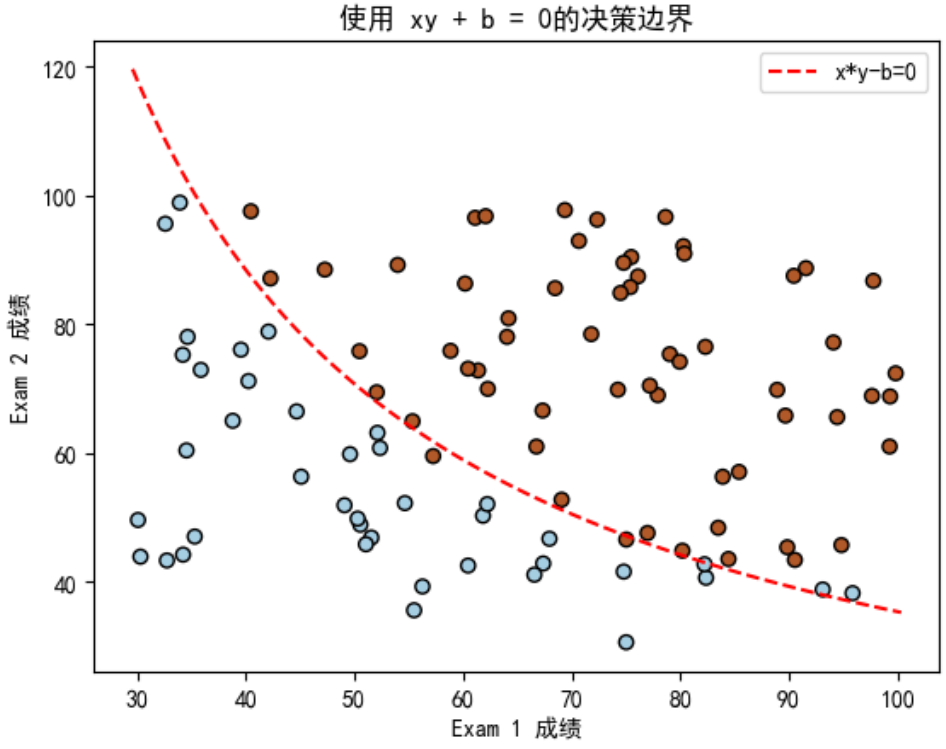

```
初始代价函数计算结果 0.6931471805599453
代价函数计算结果 0.20349770158947447
accuracy = 89%
```

自己写损失函数与梯度函数的决策边界图及准确率

3. 因为分界类似与双曲线，所以简单使用双曲线拟合

```
# 将数据分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 读取数据
numsX = X.values
numsy = y.values

passed = [] # 通过的
failed = [] # 没有通过的

# 计算边界
for i in range(len(numsX)):
    if numsy[i] == 1:
        passed.append(numsX[i][0]*numsX[i][1])
    else:
        failed.append(numsX[i][0]*numsX[i][1])

passed.sort()
failed.sort()
print(passed[0],failed[-1])

# 简单取边界的折中值
b= (passed[0] + failed[-1])/2
```

```
# 计算准确率
X_test_nums = X_test.values
y_test_nums = y_test.values

correct_train = 0
# 训练集准确率
for i in range(len(X_train)):
    if X_train.values[i][0]*X_train.values[i][1] > b:
        if y_train.values[i] == 1:
            correct_train += 1
    else:
        if y_train.values[i] == 0:
            correct_train += 1

correct = 0
# 测试集准确率
for i in range(len(X_test_nums)):
    if X_test_nums[i][0]*X_test_nums[i][1] > b:
        if y_test_nums[i] == 1:
            correct += 1
    else:
        if y_test_nums[i] == 0:
            correct += 1
train_accuracy = correct_train/len(X_train)
test_accuracy = correct/len(X_test_nums)
print(f"双曲线模型训练集准确率: {train_accuracy}")
print(f"双曲线模型测试集准确率: {test_accuracy}")
```

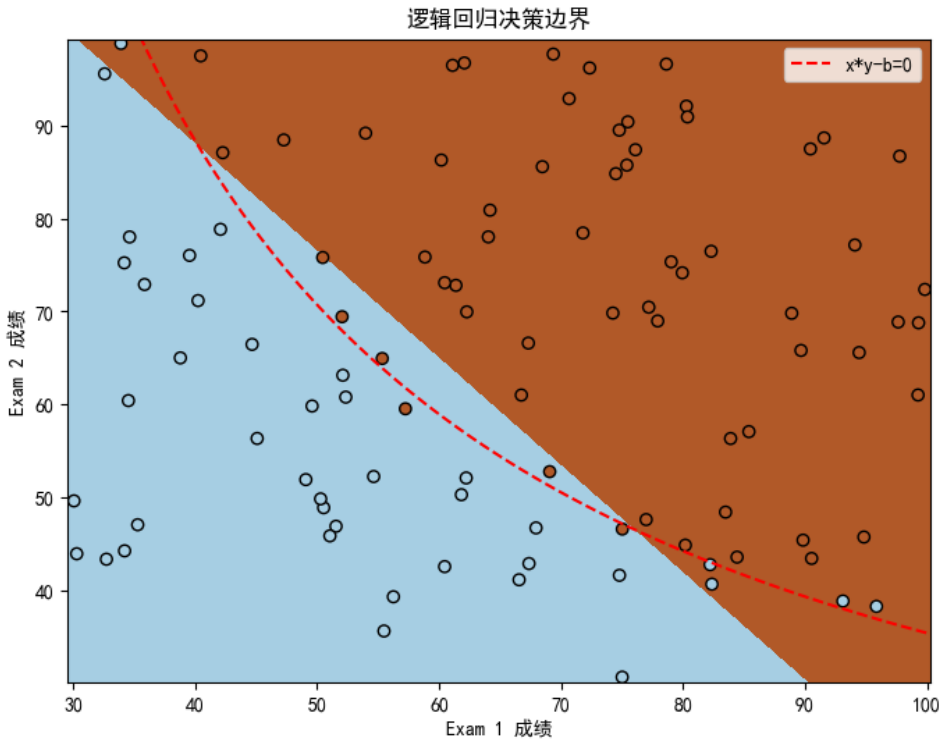



```
双曲线模型训练集准确率: 0.975
双曲线模型测试集准确率: 0.9
```

x * y - b = 0 的决策边界图及准确率

六、实验结果

比较逻辑回归决策边界的准确率与 $xy + b = 0$ 的决策边界的准确率如下：




```
逻辑回归模型训练集准确率：0.9125
逻辑回归模型测试集准确率：0.8
双曲线模型训练集准确率：0.975
双曲线模型测试集准确率：0.9
```

其中阴暗分明界是逻辑回归模型决策边界，虚线是双曲线模型决策边界；两者都可进行分类，且精度都在 80%以上。但是可以看出：双曲线模型的精度更高。

七、实验体会

通过此次实验，我体会到，对于二分类样本数据来说，逻辑回归模型是在通过不断计算损失函数与梯度函数进行迭代，不断减少代价函数与修正参数权重来寻找一个准确率较为高的模型来进行分类。

但是，逻辑回归模型只是解决线性问题，且对于一些数据倘若表现为一点非线性的特征时，此时无论迭代多少次，都无法找到最优解；例如本数据集的问题，但是，如果选择合适的非线性模型，如我针对与特征选取了双曲线模型，仅仅进行一次计算，得出的双曲线模型的精度就大于逻辑回归模型迭代 2000 次的精度。

由此可见，对于不同的数据特征，适当选取合适的模型，比一直对某个特定的模型调参更为重要。

八、教师评语

该学生_____完成了实验任务，算法设计_____，实验结果_____，实验体会_____。

因此总体评价为_____。

教师签字：

年　　月　　日