实验一 线性模型

20201060287 李昂

实验内容

使用python完成线性回归模型，逻辑回归模型问题的求解

环境要求

Python，numpy支持多维度的数组和矩阵运算，pandas数据处理和分析工具， Matplotlib 图形化工具，sklearn机器学习库

```
pip install numpy
pip install pandas
pip install matplotlib
pip install skearn -u (安装并更新sklearn和其直接依赖的包)
# 使用pip 安装sklearn时出现了安装了sklearn安装但是出现报错的问题
# 解决方法是使用这样就可以避免版本问题，导致无法引用
```

任务一：线性回归模型

题目

本任务中你将使用一元线性回归来预测食厅的利润。假设你是一家特许餐 厅的首席执行官，正在考虑在不同的城市开设一家新的分店。该连锁店已经在不 同的城市有分店，你有这些城市的利润和人口数据。你希望使用这些数据来选择下一个要扩展到的城市。

文件 ex1data.csv 包含我们的线性回归问题的数据集。Population 代表一个城市的人口，profit代表此个城市的餐厅利润，利润的负值表示亏损。

请将 70%的数据用作训练集，30%的数据用作测试集，使用留出法对以上模型进行验证。

代码

```
# -*- coding: utf-8 -*-
# @Time : 3/13/23 16:02
# @Author : ANG

# 线性回归模型分析
# 1. 从csv文件中读取数据，并进行数据预处理（pandas）
# 2. 模型训练（sklearn）
# 3. 数据可视化(matplotlib)

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


# 使用matplotlib绘制图像
def runplt():
    plt.figure()
    plt.title("profit plotted against population")
    plt.xlabel('population')
    plt.ylabel('profit')
```

```
    plt.grid(True)
    plt.xlim(0, 25)
    plt.ylim(-5, 25)
    return plt


# 从scv中读取训练集
data_set = pd.read_csv("/Users/wallanceleon/Desktop/机器学习/机器学习实验/实验一/ex1data.csv")

# 划分训练集和测试集
train_set = data_set.sample(frac=0.8, random_state=0)
test_set = data_set.drop(train_set.index)

# 训练集
train_population = train_set.loc[:, 'population'].values
train_profit = train_set.loc[:, 'profit'].values

# 测试集
test_population = test_set.loc[:, 'population'].values
test_profit = test_set.loc[:, 'profit'].values

# 构造回归对象
x = train_population.reshape((-1, 1))
y = train_profit

w = np.sum(train_population * (train_population - np.mean(train_population))) / (
            np.sum(train_population ** 2) - (1 / train_population.size) * (np.sum(train_population)) ** 2)
b = (1 / train_population.size) * np.sum(train_profit - w * train_population)

print("线性回归方程为：")
print("y = ", w, "x + ", b)

predict_y = w * train_population + b

# 显示训练集散点图和得到的回归直线
train_plot = runplt()
train_plot.plot(train_population, train_profit, 'k.')
train_plot.plot(x, predict_y, color='blue', linewidth=1)
train_plot.show()

# 显示测试集散点图和得到的回归直线
test_plot = runplt()
test_plot.plot(test_population, test_profit, 'k.')
test_plot.plot(x, predict_y, color='blue', linewidth=1)
test_plot.show()
```

分析

实验步骤主要分为以下三步：

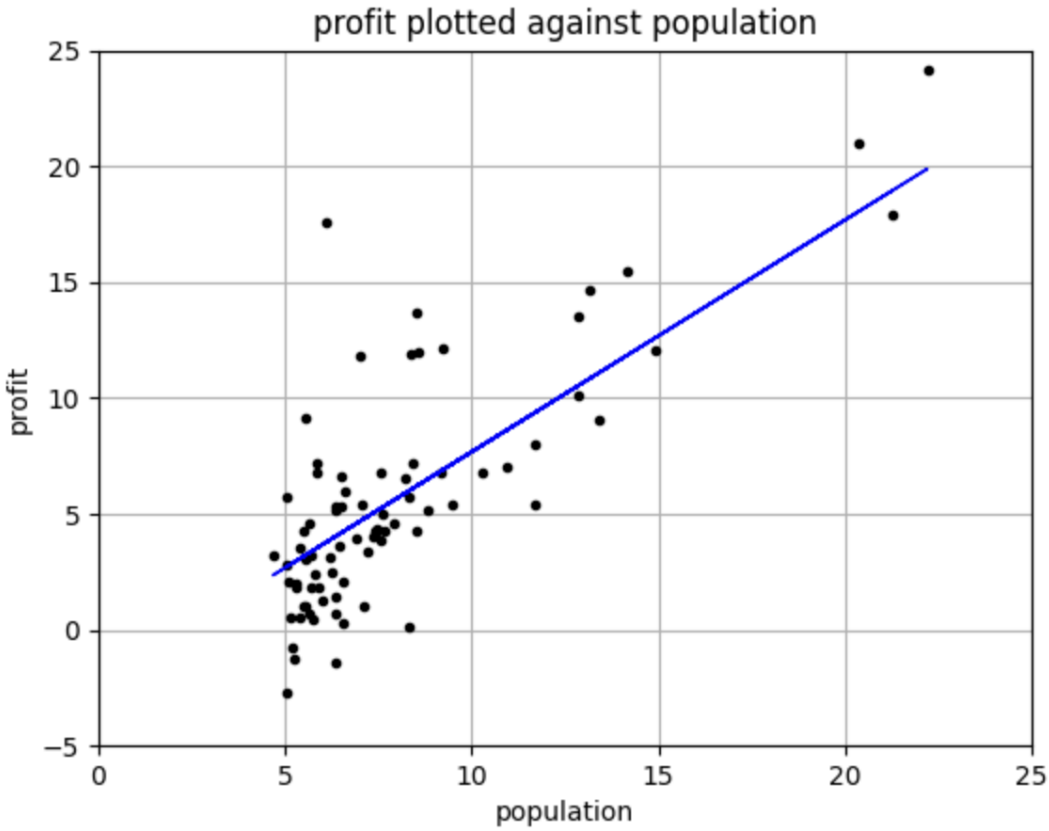
1. 从csv文件中读取数据，并进行数据预处理（pandas）
2. 模型训练（numpy）
3. 数据可视化（matplotlib）

1️⃣ 数据处理过程中使用pandas处理csv文件，这里的数据结构显示的是DataFrame类型而非常规的字典，处理过程略有不同；在进行拟合之前使用sample函数将数据集划分为训练集和测试集，并将population列表reshape便于下一步的拟合

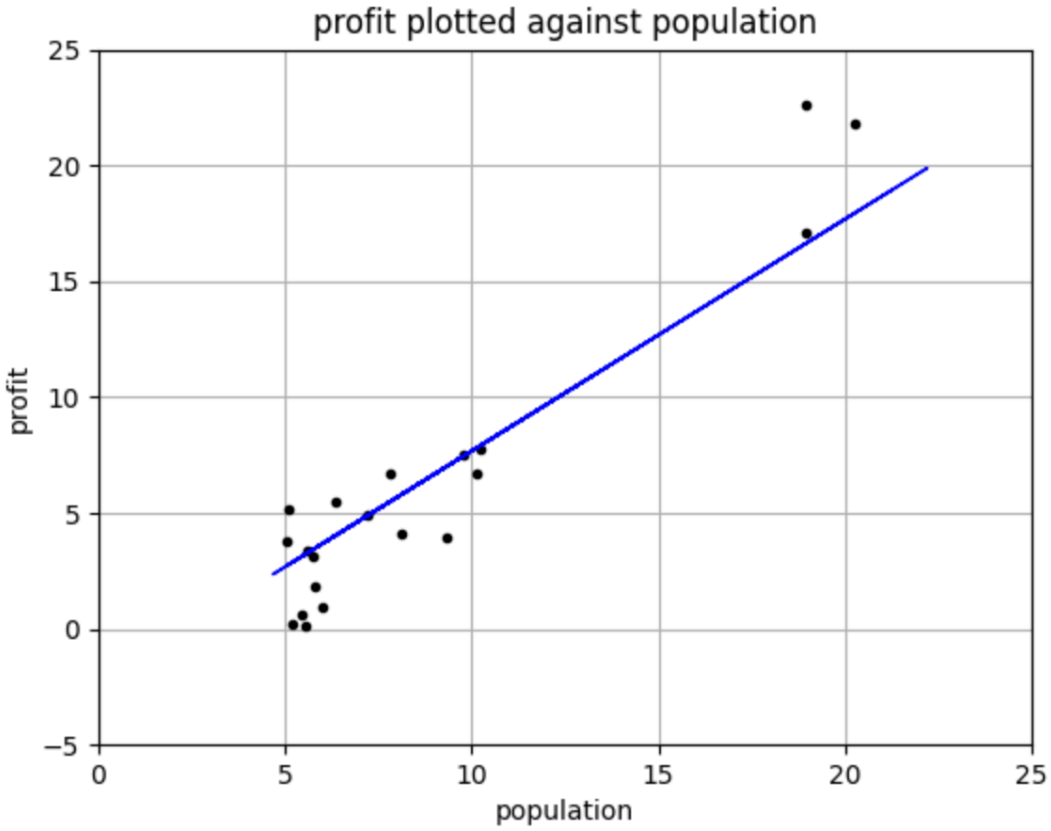
2️⃣ 由于数据仅有一纬，任务一没有使用梯度下降法，直接使用numpy库进行运算

3️⃣ 构建了runplt函数，便于可视化

实验结果



使用训练集训练得到的回归方程



使用回归方程拟合测试集

任务二：逻辑回归模型

题目

本任务中你将建立一个逻辑回归模型来预测一个学生是否被大学录取。假设你是一所大学系的管理员，你想根据两次考试的成绩来决定每个申请人的录取机会。你有以前申请者的历史数据，可以用作逻辑回归的训练集。对于每个培训示例，你都有申请人在两次考试中的分数和录取决定。你的任务是建立一个分类模型，根据这两次考试的分数来估计申请人的录取概率。

文件 ex1data2.csv 包含我们的逻辑回归问题的数据集，学生的两门成绩 Exam1， Exam2和是否被录取Accepted(1 为录取，0 为未录取)，请利用逻辑回归知识，根据学生成绩判断学生是否会被录取，并将结果可视化。

请使用 5 折交叉验证法对以上模型进行验证。

说明

本任务使用了sklearn库，但只是为了方便的进行5 折交叉验证，算法实现部分仅使用numpy完成

代码

```
# -*- coding: utf-8 -*-
# @Time : 3/13/23 16:53
# @Author : ANG

import numpy as np
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt


class LogisticRegression(object):
    """
    逻辑回归训练类
    Parameters
    ----------
    alpha : float，模型学习率
    maxiter : int，模型训练迭代次数

    """

    def __init__(self, alpha=0.3, maxiter=500):
        self.alpha = alpha  # 学习率
        self.maxiter = maxiter  # 迭代次数
        self.coef_ = None

    @staticmethod
    def sigmoid(x):
        return 1 / (1 + np.exp(-x))

    def fit(self, x, y):
        """
        梯度提升方法训练模型特征系数
        :param x: numpy or list类型，特征变量
        :param y: numpy or list类型，目标列
        """
        x = np.mat(x)  # 将数据类型转换为numpy
        y = np.mat(y).transpose()
        m, n = np.shape(x)
        self.coef_ = np.ones((n, 1))  # 初始化特征系数（n*1）向量，[1,1,1,……]
        for k in range(self.maxiter):
            h = self.sigmoid(x * self.coef_)
            error = (y - h)
            self.coef_ = self.coef_ + self.alpha / m * x.transpose() * error  # 更新特征系数

    def predict_proba(self, x):
        """
        模型预测，返回Postive的概率
        :param x: numpy or list类型，特征变量
        :return: list类型，预测正类结果
        """
        if self.coef_ is None:
            raise ValueError('模型未进行训练')
        x = np.mat(x)  # 将数据类型转换为numpy
        return self.sigmoid(x * self.coef_)


# 从csv中读取训练集,并区分数据和标记
data_set = pd.read_csv("/Users/wallanceleon/Desktop/机器学习/机器学习实验/实验一/ex1data2.csv")
score = data_set.loc[:, ['Exam1', 'Exam2']].values
Is_Accepted = data_set.loc[:, 'Accepted'].values

# 使用5折交叉验证法拆分测试集、训练集
score_train, score_test, Is_Accepted_train, Is_Accepted_test = train_test_split(score, Is_Accepted, test_size=0.2,
                                                                                random_state=0)

LogReg = LogisticRegression()
LogReg.fit(score_train, Is_Accepted_train)

# 预测
prepro = LogReg.predict_proba(score_test)
w = LogReg.coef_
print(w)

# 显示数据集散点图
plt.figure()
plt.title("Accepted plotted against Exam1 and Exam2")
plt.xlabel('Exam1')
plt.ylabel('Exam2')
plt.scatter(score[:, 0], score[:, 1], c=Is_Accepted, cmap=plt.cm.get_cmap('viridis'))
plt.show()
```

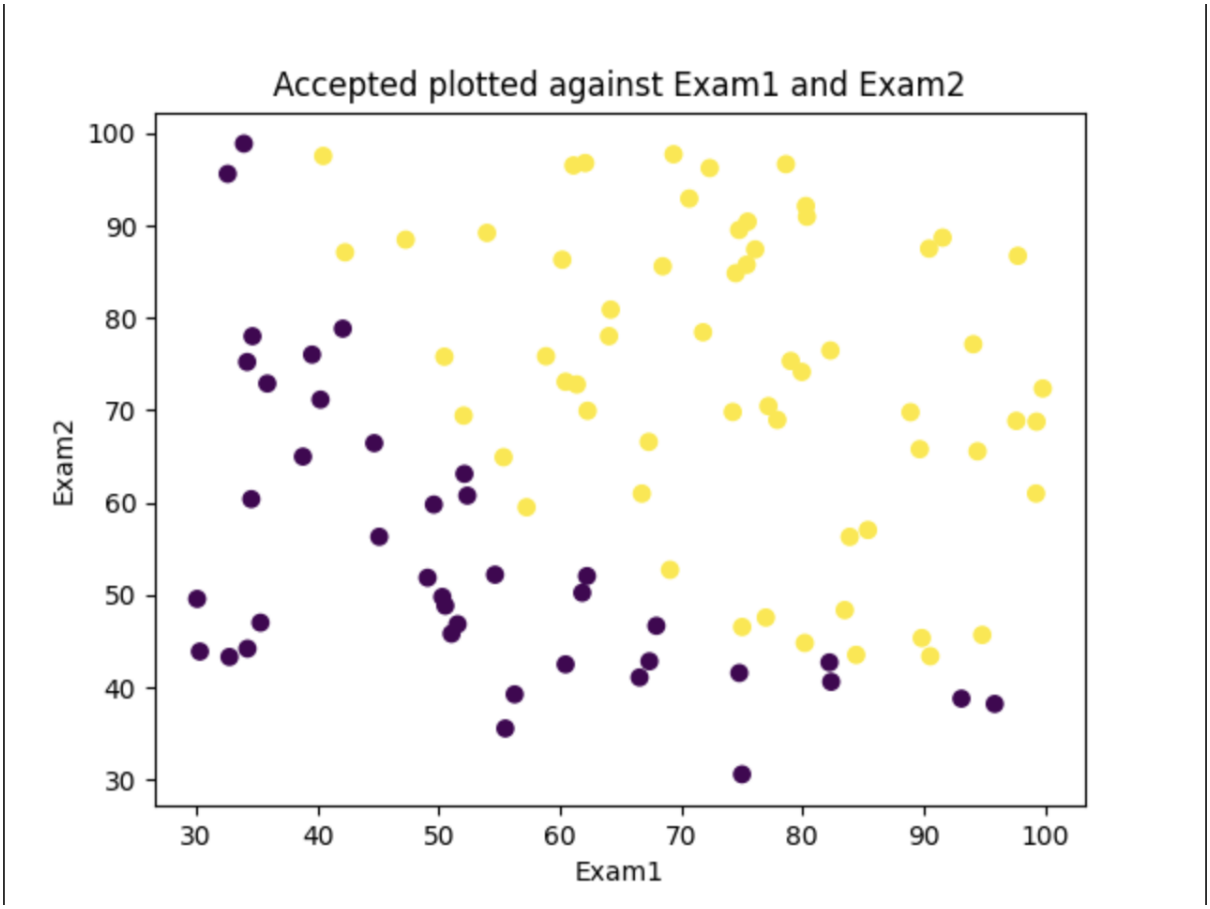
```
# 显示训练集散点图和得到的分类结果
plt.figure()
plt.title("Accepted plotted against Exam1 and Exam2")
plt.xlabel('Exam1')
plt.ylabel('Exam2')
plt.scatter(score_train[:, 0], score_train[:, 1], c=Is_Accepted_train, cmap=plt.cm.get_cmap('viridis'))
plt.show()

# 显示测试集散点图和分类结果
plt.figure()
plt.title("Accepted plotted against Exam1 and Exam2")
plt.xlabel('Exam1')
plt.ylabel('Exam2')
plt.scatter(score_test[:, 0], score_test[:, 1], c=Is_Accepted_test, cmap=plt.cm.get_cmap('viridis'))
plt.show()
```

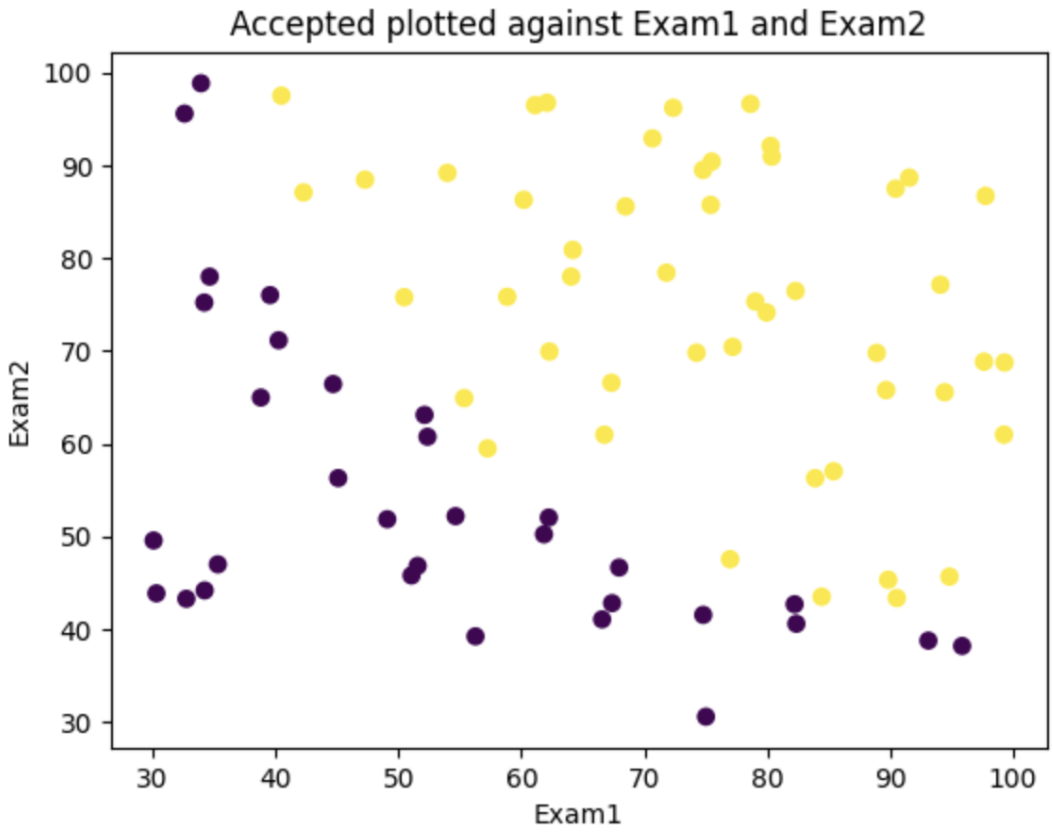
分析

💡 将实现LogisticRegression的具体步骤进行了封装，通过pandas获取数据集并区分数据和标记，使用sklearn完成5 折交叉验证，训练得到分类器，对测试集进行处理

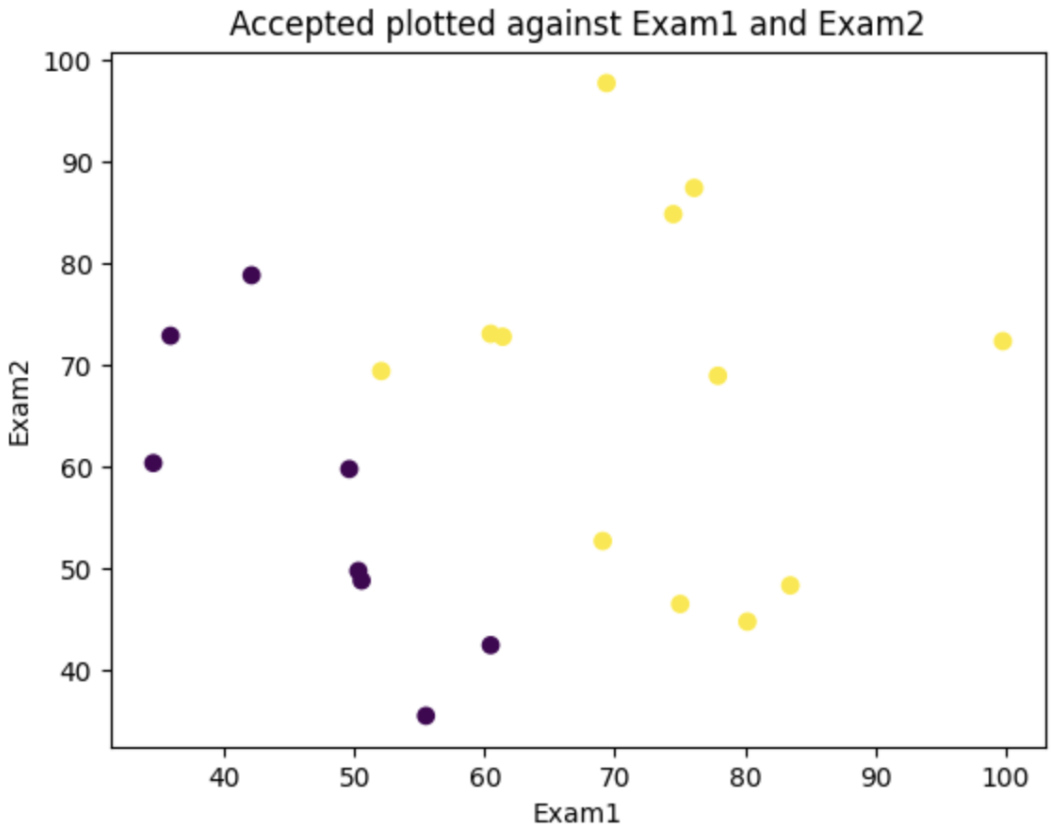
实验结果



全部数据集合（黄色为录取，紫色为未录取）



用于训练的数据集及分类结果


Accepted plotted against Exam1 and Exam2
Exam2
Exam1
100
90
80
70
60
50
40
40
50
60
70
80
90
100

测试集分类结果