**《机器学习基础》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **计算机科学与技术2022级** | | | **姓名** | **楼洋** |
| **实验题目** | **对数几率回归算法实践** | | | | | |
| **实验时间** | **2024/3/26** | | **实验地点** | **DS3402** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 □设计性 □综合性** | | |
| 教师评价：  **□**算法/实验过程正确； **□**源程序/实验内容提交 **□**程序结构/实验步骤合理；  **□**实验结果正确； **□**语法、语义正确； **□**报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的  掌握线性模型、对数几率回归算法原理。  提交：在PTA上提交 实验报告与程序代码的压缩包文件 | | | | | | |
| 二、实验项目内容  1. 理解对数几率回归算法原理；  2. 编程实现对数几率回归算法；  3. 将算法应用于西瓜数据集、鸢尾花数据集分类问题。  **实验平台**之一**：**华为云，网址：https://www.huaweicloud.com/ | | | | | | |
| 1. 实验过程或算法（源程序） 2. 对数几率回归算法设计；   逻辑回归模型，由条件概率分布P(y|x)表示。其中随机变量x 为样本属性的向量，向量的每个元素描述一个样本的对应属性，随机变量y 取值为0 或1，描述其分类标签。P(y=0|x)表示该样本为负类的概率，P(y=1|x)表示正类的概率。  对于每一个输入的样本属性向量x，对其做一个高维到一维的映射：  其中w 是转换矩阵，参数值需要训练得到；z 称为预测值，是一个实数。然而二分类任务的输出标签为y={0,1}，因此需要将z 映射成0/1 值。最理想的z->y 映射为“单位跃阶函数”，但考虑到其不具有连续、光滑、可微等优良数学特性，采用对数几率函数替代：  而对于这个函数寻找适当的w和b需要通过梯度下降来实现，对于逻辑回归模型的梯度下降，实现梯度下降前，必须求逻辑回归的成本函数：  然后通过成本函数可以实现梯度下降，从而计算出w,b：  建立对数几率回归模型代码如下：  class LogisticRegression:      # 初始化学习率和迭代次数      def \_\_init\_\_(self, learning\_rate=0.01, iterations=1000):          self.learning\_rate = learning\_rate          self.iterations = iterations        # sigmoid函数      def sigmoid(self, z):          return 1 / (1 + np.exp(-z))        # fit函数 用于梯度下降求解w和b      def fit(self, X, y):          m, n = X.shape          self.weights = np.zeros(n)          self.bias = 0            for \_ in range(self.iterations):              model = np.dot(X, self.weights) + self.bias              prediction = self.sigmoid(model)                dw = (1 / m) \* np.dot(X.T, (prediction - y))              db = (1 / m) \* np.sum(prediction - y)                self.weights -= self.learning\_rate \* dw              self.bias -= self.learning\_rate \* db        # 预测函数 用于对测试样本进行预测      def predict(self, X):          """进行预测"""          model = np.dot(X, self.weights) + self.bias          prediction = self.sigmoid(model)          return [1 if i > 0.5 else 0 for i in prediction]        def accuracy\_score(self,y\_true, y\_pred):          """计算准确率"""          correct\_predictions = sum(y\_pred[i] == y\_true[i] for i in range(len(y\_true)))          accuracy = correct\_predictions / len(y\_true)          return accuracy      def coef\_(self):          """返回w"""          return self.weights        def intercept\_(self):          """返回b"""          return self.bias        def train\_test\_split(self,X, y, test\_size=0.2, random\_state=None):          """将数据集划分为训练集和测试集"""          if random\_state:              np.random.seed(random\_state)            # 随机打乱索引          indices = np.arange(X.shape[0])          np.random.shuffle(indices)            # 根据test\_size计算测试集大小          test\_set\_size = int(X.shape[0] \* test\_size)            # 划分测试集和训练集          test\_indices = indices[:test\_set\_size]          train\_indices = indices[test\_set\_size:]            # 划分数据集          X\_train = X[train\_indices]          X\_test = X[test\_indices]          y\_train = y[train\_indices]          y\_test = y[test\_indices]            return X\_train, X\_test, y\_train, y\_test   1. 数据集获取；   数据集皆为网络上获取，详见附件Iris.csv和watermelon\_3a.csv  对于数据集导入的读取采用csv库,数据处理如下：  1.对于西瓜数据集的处理  f=open(r'data/watermelon\_3a.csv',encoding='GB2312')  reader=csv.reader(f)  rows=[row for row in reader]        #rows[i]=[编号,密度,含糖率,好瓜]  X=[]                                #样本集  y=[]                                #标签值  for i in range(1,len(rows)):      X.append([float(rows[i][1]),float(rows[i][2])])      y.append(int(rows[i][4]))  X=np.array(X)                       #将list转变成numpy数组  y=np.array(y)  2.对于鸢尾花数据集的处理  f=open(r'data/Iris.csv',encoding='GB2312')  reader=csv.reader(f)  rows=[row for row in reader]        #rows[i]=[编号,SepalLengthCm,SepalWidthCm,PetalLengthCm,PetalWidthCm,Species]  X=[]                                #样本集  y=[]                                #标签值  for i in range(1,len(rows)):      X.append([float(rows[i][1]),float(rows[i][2]),float(rows[i][3]),float(rows[i][4])])      if rows[i][5]=='Iris-setosa':          y.append(0)      else:          y.append(1)  X=np.array(X)                       #将list转变成numpy数组  y=np.array(y)   1. 模型训练及调参。   对于西瓜数据集的训练，由于样本数较少，直接将整个样本作为训练集，进行训练，并进行调参，算法如下：  model = LogisticRegression()  model.fit(X, y)  predictions = model.predict(X)  y\_pred=np.array(predictions)  accuracy = model.accuracy\_score(y, y\_pred)  print(f"Accuracy: {accuracy}")  对于鸢尾花数据集的训练，由于样本数较多，通过模型中的train\_test\_split函数将数据集划分为训练集和测试集，从而获取更好的效果，进行训练，并进行调参，算法如下：  model = LogisticRegression()  model.fit(X\_train, y\_train)  predictions = model.predict(X\_test)  y\_pred=np.array(predictions)  accuracy = model.accuracy\_score(y\_test, y\_pred)  print(f"Accuracy: {accuracy}") | | | | | | |
| 四、实验结果及分析  (1)对于西瓜数据集  学习率为0.01，迭代数为1000次，准确度结果为0.6470588235294118，进行调参发现，学习率为0.03，迭代数为1000次的时候，准确度为0.7647058823529411，  学习率为0.02，迭代数为2000次，准确度为0.7058823529411765，并且对于这三个不同的参数，我作图如下：      图一：西瓜数据三种参数预测结果图  由预测结果和图可知，当学习率和迭代次数不同，预测结果会有所不同，且通过不断调试容易知道，当迭代次数越大，一般预测准确度会好，而学习率的选择也对模型训练有重要作用  (2)对于鸢尾花数据集  由于数据较多，预测结果比西瓜数据集好不少，当学习率为0.01，迭代次数为1000时，预测准确率便为1，将迭代次数减少到100，可以发现预测准确率降到0.7666666666666667，将学习率设为0.02，迭代次数为100时，可以发现预测准确率为1，可以发现学习率的设置，对于模型训练的效果有显著的影响，故在模型训练中，学习率的选择由为重要。  (3)总结  在逻辑回归模型中，迭代次数在机器学习模型的训练过程中非常重要。它指的是模型在训练数据集上的遍历次数。每次迭代，模型都会尝试调整其参数（如权重）以最小化损失函数，这是学习过程的核心。  同样，学习率的选择对模型的训练效果有重要影响。学习率决定了在梯度下降过程中参数更新的步长。如果学习率设置得太大，可能会导致模型在最优解附近震荡，甚至无法收敛；如果学习率设置得太小，模型可能需要更多的迭代次数才能收敛，训练过程会变得非常缓慢。  通常，学习率的设置需要根据具体问题和数据集进行调整。一些常用的初始学习率值包括 0.01、0.001 或 0.0001。 | | | | | | |

**说明：学生应按照如下要求正确地撰写实验报告：**

* 1. 在实验报告上正确地填写“实验时间”、“实验地点”等栏目。
  2. 将实验所涉及的源程序文件内容（实验操作步骤或者算法）填写在“实验过程或算法（源程序）”栏目中。
  3. 将实验所涉及源程序调试过程（输入数据和输出结果）或者实验的分析内容填写在“实验结果及分析”栏目中。
  4. 在实验报告页脚的“报告创建时间：”处插入完成实验报告时的日期和时间。
  5. 学生将每个实验完成后，按实验要求的文件名通过PTA提交（上载）到指定处。每个实验一个电子文档，如果实验中有多个电子文档（如源程序或图形等），则用WinRAR压缩成一个压缩包文档提交，压缩包文件名同实验报告文件名（见下条）。
  6. 提交的实验报告电子文档命名为：“年级（两位数字不要“级”字）专业（缩写：计算机科学与技术专业（计科）、网络工程专业（网络）、信息安全专业（信息）、物联网工程（物联网））班级（两位数字）学号（八位数字）姓名实验序号（一位数字）．doc。如学号为20185676、年级为2018级、专业为“计算机科学与技术”专业、班级为“02班”、姓名为“王宇”的学生，完成的第一次实验命名为： **18计科02班20185676王宇-实验1.Doc**，以后几次实验的报告名称以此类推。