

**实 验 报 告**

**（ 2024 / 2025 学年 第 一 学期 ）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | Python程序设计（混合式）（JS170101S） | | | | | |
| 实验名称 | 实验二：线性回归与逻辑斯蒂分类实验 | | | | | |
| 实验时间 | 2024 | 年 | 11 | 月 | 22 | 日 |
| 指导单位 | 计算机学院、软件学院、网络空间安全学院 | | | | | |
| 指导教师 | 杨尚东 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 袁浩宽 | 班级学号 | B23042424 |
| 学院(系) | 计算机学院、软件学院、网络空间安全学院 | 专 业 | 数据科学与大数据技术 |

| **实 验 报 告** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | **线性回归与逻辑斯蒂分类实验** | | | **指导教师** | 杨尚东 |
| **实验类型** | **验证** | **实验学时** | **2** | **实验时间** | 2学时 |
| 1. **实验目的和要求**    1. 掌握线性回归的基本原理。    2. 掌握逻辑斯蒂分类的基本原理。    3. 学会使用Python语言实现线性回归算法。    4. 学会使用Python语言实现逻辑斯蒂分类。 | | | | | |
| 二、**实验环境(实验设备)**  硬件: 微型计算机  软件: Windows / MacOS / Linux 操作系统、Python3.7或3.8 | | | | | |
| 1. **实验原理及内容** 2. 使用线性回归对房屋单价（数据集:题目1数据.txt）进行预测，需要使用前350行作为训练，351-414行作为测试数据，并对测试数据输出评价指标R方和残差平方和。    1. 先用sklearn实现线性回归；    2. 再用最小二乘或梯度下降法实现（参考书上代码）。   选用随机种子42：np.random.seed(42)  提示：用np.loadtxt()函数进行读取，示例代码如下  import numpy as np  a = np.loadtxt("题目1数据.txt")  注意题目1数据.txt中数据的最后一列是y值，即房屋单价值，其它列是属性。  import numpy as np  import matplotlib.pyplot as plt  from sklearn.preprocessing import StandardScaler  from sklearn.linear\_model import LinearRegression  np.random.seed(42)  data = np.loadtxt("data/experiment\_2/题目1数据.txt")  X, y = data[:, :-1], data[:, -1]  scaler = StandardScaler()  X = scaler.fit\_transform(X)  X\_train, X\_test = X[:350], X[350:414]  y\_train, y\_test = y[:350], y[350:414]  model = LinearRegression()  model.fit(X\_train, y\_train)  y\_pred = model.predict(X\_test)  num\_features = X.shape[1]  fig, axes = plt.subplots(num\_features, 1, figsize=(10, 5 \* num\_features))  rss = np.sum((y\_test - y\_pred) \*\* 2)  tss = np.sum((y\_test - np.mean(y\_test)) \*\* 2)  r2 = 1 - rss / tss  print("Manual R^2 and analysis results:")  print(**f**"R^2: {r2**:.4f**}")  print(**f**"RSS: {rss**:.4f**}")  for i in range(num\_features):      ax = axes[i] if num\_features > 1 else axes      ax.scatter(X\_train[:, i], y\_train, color='blue', label='Train', alpha=0.7,s=5)      ax.scatter(X\_test[:, i], y\_test, color='red', label='Test', alpha=0.7,s=5)        x\_vals = np.linspace(X[:, i].min(), X[:, i].max(), 100).reshape(-1, 1)      x\_vals\_all = np.zeros((100, num\_features))      x\_vals\_all[:, i] = x\_vals[:, 0]      y\_vals = model.predict(x\_vals\_all)        ax.plot(x\_vals, y\_vals, color='green', label='Fit line')      ax.set\_title(**f**"Feature {i + 1} vs Price")      ax.set\_xlabel(**f**"Feature {i + 1}")      ax.set\_ylabel("Price")      y\_min = min(y\_train.min(), y\_test.min(), y\_vals.min())      y\_max = max(y\_train.max(), y\_test.max(), y\_vals.max())      ax.set\_ylim(y\_min, y\_max)      ax.legend()  plt.savefig("img/features.png")  plt.tight\_layout()  plt.show()  ’’’  拟合效果可视化（在不同特征维度上）:    ’’’   1. 编写代码，使用Logistic Regression对训练数据（题目2训练数据.txt）进行模型训练，并针对测试数据（题目2测试数据.txt）输出二分类结果，结果用0-1之间的概率表示，用空格进行分割，分别计算在训练集和验证集（训练数据后20%）的AUC值。  * 数据集说明   该数据集需要通过特征数据分析，来预测二分类的结果。训练数据集中的二分类结果已经给出，为最后一列   * 数据集信息   本数据集用于估计信用卡的违约概率。其中训练数据包括27997个样本，测试数据包括2001个样本。需要根据训练数据，建立Logistic Regression模型，对测试数据集中的数据预测结果。  结果文件为一个txt文件，其中包含2001个0-1之间的实数值，保留两位小数，用空格分割。最后会在真实的结果上比较每个同学的预测结果。例如：  0.01 0.24 0.55 0.67   * 数据属性   本数据集供包括23个特征，具体说明如下：  - 第1列：信用卡的额度  - 第2列：性别（1 = male; 2 = female）  - 第3列：教育（1 = graduate school; 2 = university; 3 = high school; 4 = others）  - 第4列：婚姻状态（1 = married; 2 = single; 3 = others）  - 第5列：年龄.  - 第6 - 11列：过去几个月的还款状态（4月到9月），-1，-2代表按时还款，其他代表延迟还款的月份数  - 第12 - 17列：过去几个月的账单记录  - 第 18 – 23 列：过去几个月的支付记录  - 最后一列为类别y值，取值为0或1  要求：  （1）使用Sklearn实现  （2）自定义编码使用梯度下降法实现（选做）  import numpy as np  import pandas as pd  from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import roc\_auc\_score  from sklearn.model\_selection import train\_test\_split  import matplotlib.pyplot as plt  train\_data = pd.read\_csv('data/experiment\_2/题目2训练数据.txt', header=None, delim\_whitespace=True)  test\_data = pd.read\_csv('data/experiment\_2/题目2测试数据.txt', header=None, delim\_whitespace=True)  X\_train = train\_data.iloc[:, :-1]  y\_train = train\_data.iloc[:, -1]  X\_test = test\_data.iloc[:,:]  X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_train, y\_train, test\_size=0.2, random\_state=42)  model = LogisticRegression(      penalty='l2',      C=1.0,      solver='saga',      max\_iter=500,      class\_weight='balanced',      random\_state=42  )  model.fit(X\_train, y\_train)  y\_train\_prob = model.predict\_proba(X\_train)[:, 1]  y\_val\_prob = model.predict\_proba(X\_val)[:, 1]  train\_auc = roc\_auc\_score(y\_train, y\_train\_prob)  val\_auc = roc\_auc\_score(y\_val, y\_val\_prob)  print(**f**'Training AUC: {train\_auc**:.4f**}')  print(**f**'Validation AUC: {val\_auc**:.4f**}')  y\_test\_prob = model.predict\_proba(X\_test)[:, 1]  np.savetxt('prediction.txt', y\_test\_prob, fmt='%.2f')  plt.figure(figsize=(8, 6))  plt.scatter(range(len(y\_train\_prob)), y\_train\_prob,label='Training Prediction Probabilities', alpha=0.5,s = 0.5)  plt.scatter(range(len(y\_val\_prob)), y\_val\_prob,label='Validation Prediction Probabilities', alpha=0.5, s = 0.5)  plt.title(**f**'Prediction Probabilities (Train AUC: {train\_auc**:.4f**}, Val AUC: {val\_auc**:.4f**})')  plt.xlabel('Samples')  plt.ylabel('Prediction Probability')  plt.legend()  plt.savefig('img/problem\_2\_visual.png')  plt.show()  ’’’  Plot result:    ’’’ | | | | | |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议、实验出错信息及解决方案等）  (一)实验中遇到的主要问题及解决方法  在逻辑斯蒂回归模型的训练中，我遇到了计算效率的问题。由于数据集较大，模型训练时间较长，且计算资源的限制使得模型训练速度受到一定影响。为了解决这个问题，我调整了模型的超参数，如增加了迭代次数并选择了合适的求解器。通过这些优化，训练效率得到了显著提升。此外，我还在模型训练时加入了早停机制，通过提前停止训练避免了无意义的计算浪费，提高了训练的整体效率。  （二）实验心得  通过这次实验，我对回归和分类问题的建模过程有了更深入的理解。尤其是在逻辑斯蒂回归模型中，我学会了如何利用AUC这一指标来评估分类模型的性能。AUC的计算不仅让我掌握了模型的评估技巧，还帮助我理解了分类模型对于不同类型数据的适应性。  在处理线性回归时，我深刻认识到标准化对于模型性能的重要性，尤其是在特征之间存在量纲差异时，标准化是不可或缺的。通过模型拟合和评估的过程，我对回归分析的应用有了更加清晰的理解。  （三）意见与建议（没有可省略）  从实验的过程中，我发现实验代码涉及的内容较为基础，适合入门级的学习和理解。然而，若想进一步提升模型的精度和泛化能力，可以在此基础上增加一些进阶内容。例如，在线性回归中加入正则化项（如岭回归和套索回归），可以有效防止模型过拟合；在逻辑斯蒂回归中，利用交叉验证来选择最优参数，能够进一步提升模型的泛化能力。 | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **支撑毕业要求指标点**   （一）思想、素质教育目标  目标1.1 在教学过程中，激发学生自豪感与爱国情怀，鼓励学生通过努力学习掌握先进科学技术，服务国家，回馈社会。  目标1.2 通过Python编程、人工智能主流算法等知识的学习，增强学生的科技强国意识，**帮助其树立正确的价值观。**  （二）知识教学目标  目标2.1 使学生掌握Python语言编程。  目标2.2 使学生掌握人工智能基本算法及Python语言实现方法。  （三）能力教学目标  目标3.1 使学生掌握Python语言编程方法，具有使用Python语言进行编程的能力。  目标3.2 使学生掌握线性回归、逻辑斯蒂分类、朴素贝叶斯分类、神经网络等人工智能基本算法，使学生具有使用Python语言实现人工智能基本算法的能力，对实际应用中的问题进行设计研究方案的能力。 | | | | | | |
| **六、指导教师评语** | | | | | | |
| **评**  **分**  **细**  **则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **回答问题准确度** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  | **批阅人** |  | **日期** |  |