|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名： 张晨龙 | | 学号：1925050069 | 专业年级：2019 | 班级：软件工程1班 |
| 分组： 无 | | 实验室： 曾宪梓楼503 | 指导教师：李霞 | 实验日期：2022.3.14 |
| **实验的准备阶段** | **课程名称** | 机器学习及与智能数据处理 | | |
| **实验名称** | 一元线性回归-最小二乘法 | | |
| **实验目的** | 1. 掌握最小二乘法的数学原理及推导过程 2. 编程实现最小二乘法，并应用于具体案例 | | |
| **实验内容** | 1. 使用最小二乘法进行房价预测：   给定训练样本集合如下：    求解：当房屋面积为55平方时，租赁价格是多少？给出代码与运行结果图。   1. 使用最小二乘法进行贷款额度预测：   给定训练样本集合如下：    求解：当工资18000、年龄30时，额度是多少？给出代码与运行结果图 | | |
| **实验类型**  （打☑） | □验证性 □演示性 🗹设计性 □综合性 | | |
| **实验的重点、难点** | 最小二乘法的线性回归矩阵计算 | | |
| **实验环境** | Python 3.6  Numpy | | |
| **实验的实施阶段** | **实验步骤及实验结果** | **1.使用最小二乘法进行房价预测：**  **import** matplotlib.pyplot **as** plt **import** numpy **as** np  **def** build\_dataset(): *# 创建用于训练的数据集 create dataset to be trained* X = np.asmatrix([10,15,20,30,50,60,60,70])  y = np.asmatrix([0.8,1,1.8,2,3.2,3,3.1,3.5])  **return** X,y  **def** main():  X,Y = build\_dataset()  *# 为训练集添加一列1* X\_temp = np.ones((X.shape[1],2))  X\_temp[:,0] = X  X = np.asmatrix(X\_temp)  *# 使用线性回归得到模型的权重 params -> [w,b] np.ndarray* params = (((X.T \* X).I) \* X.T) \* Y.T  *# 打印线性回归得到的模型* print(**"The trained LinearRegression got function : Y = %f \* x + %f "** % (params[0], params[1]))  *# 定义预测集* X\_predict = np.asarray([55,1])  *# 用模型进行预测* Y\_predict = X\_predict \* params  *# 打印预测结果* print(**"The input square %s m^2 got predicted price %s !"** % (X\_predict[0],float(Y\_predict))) **if** \_\_name\_\_ == **'\_\_main\_\_'**:  main()  **2. 使用最小二乘法进行贷款额度预测：**  **import** matplotlib.pyplot **as** plt **import** numpy **as** np  **def** build\_dataset():  *# 创建用于训练的数据集 create dataset to be trained* X = np.asmatrix([[4000, 8000, 5000, 7500, 12000],  [25, 30, 28, 33, 40]])  y = np.asmatrix([20000, 70000, 35000, 50000, 85000])  **return** X, y   **def** main():  X, Y = build\_dataset()  *# 为训练集添加一列1* X\_temp = np.ones((X.shape[1], 3))  X\_temp[:, :2] = X.transpose()  X = np.asmatrix(X\_temp)  *# 使用线性回归得到模型的权重 params -> [w1, w2,b] np.ndarray* params = (((X.T \* X).I) \* X.T) \* Y.T  *# 打印线性回归得到的模型* print(**"The trained LinearRegression got function : Y = %f \* x1 + %f \* x2 + %f "** % (params[0], params[1], params[2]))  *# 定义预测集* X\_predict = np.asarray([18000, 30, 1])  *# 用模型进行预测* Y\_predict = X\_predict \* params  *# 打印预测结果* print(**"The input salary %s and age %s got predicted loan %s !"** % (X\_predict[0], X\_predict[1], float(Y\_predict)))   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  main() | | |
| **实验结果的处理阶段** | **实验结果的分析与总结** | 1. **使用最小二乘法进行房价预测结果：**     **2. 使用最小二乘法进行贷款额度预测结果：** | | |