**机器学习作业二**

**20201060287 李昂**

**1.解：  
 文本, 信件

描述已自动生成**

**2.对线性回归模型，试解析在什么情况下可以消去线性回归的偏置项b**

当所有样本的自变量的均值为0时，可以消去线性回归的偏置项b。因为当所有自变量的均值为0时，模型中的偏置项b对因变量的期望值没有影响，可以将其消去。

在一些特殊情况下，也可以消去线性回归的偏置项b。如，在一些需要通过线性回归来拟合原点通过的模型中，因为原点通过，偏置项为0，所以可以将其消去。

**3. 试用队列实现非递归的决策树算法**

|  |
| --- |
| *# -\*- coding: utf-8 -\*-  # @Time : 1/4/23 11:55  # @Author : ANG* class TreeNode:  def \_\_init\_\_(self, value):  self.val = value  self.left = None  self.right = None  def build\_decision\_tree(data):  root = TreeNode(None) *# 初始化根节点* queue = [(root, data)] *# 将根节点和数据集放入队列中* while queue:  node, subset = queue.pop(0) *# 取出队首元素* if len(subset) == 0: *# 如果数据集为空，则返回* continue  labels = [item[-1] for item in subset]  if labels.count(labels[0]) == len(labels): *# 如果数据集中所有实例都属于同一类别，则将该节点标记为叶子节点* node.val = labels[0]  continue  best\_feature = choose\_best\_feature(subset) *# 选择最优特征* node.val = best\_feature  left\_subset, right\_subset = split\_data\_set(subset, best\_feature) *# 划分数据集* node.left = TreeNode(None) *# 创建左子树* node.right = TreeNode(None) *# 创建右子树* queue.append((node.left, left\_subset)) *# 将左子树和数据集放入队列中* queue.append((node.right, right\_subset)) *# 将右子树和数据集放入队列中* return root |