机器学习-2023春 作业2

***Steven***

1. 逻辑回归求导：

对于逻辑回归模型，其损失函数为，其中为样本总数，表示第个样本的标签，表示第个样本的特征向量，为模型参数，与均为维向量。对于损失函数梯度，我们有。请计算并给出详细推导过程。

设中间变量，则。

1. 计算：
2. 计算：
3. 计算：
4. 对于线性回归模型，试解析在什么情况下可以消去线性回归的偏置项。

根据最小二乘法可知，，当时，。此时对于所有样本，设作为新的目标值，则此时回归模型函数可用代替，消去偏置项以简化运算。

1. 试用队列实现非递归的决策树算法。

下例以Python风格伪代码表示非递归决策树生成算法。

|  |
| --- |
| D={(X1,y1),(X2,y2),...,(Xn,yn)} # 训练集  A={a1,a2,...,am} # 属性集  def TreeGenerate(D, A):      queue = [Node(D, A)] # 生成队列并将结点加入队列      while queue:          root = queue.pop(0) # 从队列中取出结点          if D中样本全属于同一类别C:              root.type = C              root.is\_leaf = True              return root          if A is None or D中样本在A上取值相同:              root.type = max(count(D.type)) # 选择D中样本数最多的类别              root.is\_leaf = True              return root          attr\_best = A.best\_attr() # 选择最优划分属性          for attr\_v in attr\_best.values: # 遍历最优划分属性的每个取值              Dv = D[attr\_v] # 生成Dv              node\_branch = Node(Dv, A-{attr\_best}) # 生成新结点              if Dv is None:                  node\_branch.type = max(count(D.type)) # 标记为D中样本数最多的类别                  node\_branch.is\_leaf = True                  root.children.append(node\_branch) # 构建树              else:                  root.children.append(node\_branch)                  queue.append(node\_branch) # 将新结点加入队列，下次while将会从该结点开始继续构建树        return root |