Assignment 5 详细解析

第一题

(1) Bagging 与随机森林的计算成本

问题： Bagging 和随机森林使用决策树作为基学习器，哪一个的计算成本更低？并说明原因。（假设它们的训练迭代次数相同）

答案： 随机森林的计算成本更低。

原因：

- **Bagging**：在Bagging中，多个决策树是通过对原始数据集进行自助采样（Bootstrap sampling）生成的。每棵树都使用了所有特征，因此在训练每棵树时都需要遍历所有的特征进行最佳分裂点的选择。
- **随机森林**：随机森林在每棵树的每个节点分裂时引入了特征随机性。具体来说，在分裂节点时，随机选择部分特征进行分裂。这种随机性减少了每棵树在每个节点的计算复杂度，因为只需要在部分特征上进行分裂点选择，而不是全部特征。

由于随机森林在每个节点分裂时只使用部分特征，因此减少了每棵树训练时的计算成本，尤其是在高维数据情况下，特征子集的选择显著降低了计算开销。

(2) Bagging 和随机森林如何引入随机性以生成多样的个体学习器？

答案：

- **Bagging**：通过对原始数据集进行自助采样（Bootstrap sampling）引入随机性。每棵树都在不同的自助样本上训练，从而引入了数据的随机性。
- **随机森林**：除了使用自助采样外，还在每棵树的每个节点引入了特征随机性。在分裂节点时，随机选择部分特征进行分裂，从而增加了模型的多样性。

(3) 从偏差-方差分解的角度，Bagging 和随机森林分别减少了什么？

答案：

- **Bagging**：主要减少了模型的方差。通过在不同的自助样本上训练多个模型，并将它们的预测结果进行平均，Bagging能够降低模型对训练数据的敏感性，从而减少方差。
- **随机森林**：不仅减少了模型的方差，还通过引入特征随机性降低了偏差。特征随机性使得每棵树不会过度拟合训练数据中的某些特征，因此在一定程度上也减少了偏差。

(4) 在回归问题中应用 LASSO 的潜在目的是什么？

答案：

- **特征选择**：LASSO（Least Absolute Shrinkage and Selection Operator）通过在损失函数中加入 L1 范数的正则化项，能够将一些回归系数缩小到零，从而实现特征选择。
- **防止过拟合**：通过引入正则化项，LASSO 可以限制模型的复杂度，防止模型对训练数据过拟合，从而提高模型的泛化能力。

(5) 在线性回归模型中使用 L1 范数和 L2 范数作为正则化项的区别是什么？

答案：

- **L1 范数（LASSO）**：
  - **特点**：L1 范数的正则化项是回归系数的绝对值之和。它会产生稀疏解，即一些回归系数会被缩小到零，从而实现特征选择。
  - **应用**：适用于希望进行特征选择的场景，特别是当存在许多不相关或冗余特征时。
- **L2 范数（Ridge 回归）**：
  - **特点**：L2 范数的正则化项是回归系数的平方和。它会将回归系数均匀地缩小，但不会将它们缩小到零。
  - **应用**：适用于当所有特征都对模型有一定贡献时，不希望完全忽略任何特征。

总之，L1 范数正则化会产生稀疏模型，有助于特征选择，而 L2 范数正则化则主要用于防止过拟合，不会导致特征的完全去除。

第二题

初始距离矩阵

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
A & 0 & 12 & 6 & 2 & 3 & 1 \\
B & 12 & 0 & 8 & 7 & 6 & 8 \\
C & 6 & 8 & 0 & 9 & 2 & 20 \\
D & 2 & 7 & 9 & 0 & 7 & 6 \\
E & 3 & 6 & 2 & 7 & 0 & 2 \\
F & 1 & 8 & 20 & 6 & 2 & 0
\end{array}
$$

Step 1: 合并 A 和 F

计算新簇 (A,F) 与其他簇之间的距离：

$$
\begin{aligned}
d((A,F),B) &= \frac{d(A,B)+d(F,B)}{2} = \frac{12+8}{2} = \frac{20}{2} = 10, \\
d((A,F),C) &= \frac{d(A,C)+d(F,C)}{2} = \frac{6+20}{2} = \frac{26}{2} = 13, \\
d((A,F),D) &= \frac{d(A,D)+d(F,D)}{2} = \frac{2+6}{2} = \frac{8}{2} = 4, \\
d((A,F),E) &= \frac{d(A,E)+d(F,E)}{2} = \frac{3+2}{2} = \frac{5}{2} = \frac{5}{2}.
\end{aligned}
$$

更新后的距离矩阵：

$$
\begin{array}{c|ccccc}
 & (A,F) & B & C & D & E \\
\hline
(A,F) & 0 & 10 & 13 & 4 & \frac{5}{2} \\
B & 10 & 0 & 8 & 7 & 6 \\
C & 13 & 8 & 0 & 9 & 2 \\
D & 4 & 7 & 9 & 0 & 7 \\
E & \frac{5}{2} & 6 & 2 & 7 & 0
\end{array}
$$

Step 2: 合并 C 和 E

计算新簇 (C,E) 与其他簇之间的距离：

$$
\begin{aligned}
d((C,E),(A,F)) &= \frac{d(C,(A,F)) + d(E,(A,F))}{2} = \frac{13 + \frac{5}{2}}{2} = \frac{\frac{26}{2} + \frac{5}{2}}{2} = \frac{\frac{31}{2}}{2} = \frac{31}{4}, \\
d((C,E),B) &= \frac{d(C,B) + d(E,B)}{2} = \frac{8+6}{2} = \frac{14}{2} = 7, \\
d((C,E),D) &= \frac{d(C,D) + d(E,D)}{2} = \frac{9+7}{2} = \frac{16}{2} = 8.
\end{aligned}
$$

更新后的距离矩阵：

	(A,F)	(C,E)	B	D
(A,F)	0	$\frac{31}{4}$	10	4
(C,E)	$\frac{31}{4}$	0	7	8
B	10	7	0	7
D	4	8	7	0

Step 3: 合并 (A,F) 和 D

计算新簇 (A,F,D) 与其他簇之间的距离：

$$
\begin{aligned}
d((A,F,D),(C,E)) &= \frac{d((A,F),(C,E)) + d(D,(C,E))}{2} = \frac{\frac{31}{4} + 8}{2} = \frac{\frac{31}{4} + \frac{32}{4}}{2} = \frac{\frac{63}{4}}{2} = \frac{63}{8}, \\
d((A,F,D),B) &= \frac{d((A,F),B) + d(D,B)}{2} = \frac{10+7}{2} = \frac{17}{2} = 8.5.
\end{aligned}
$$

更新后的距离矩阵：

	(A,F,D)	(C,E)	B
(A,F,D)	0	$\frac{63}{8}$	$\frac{17}{2}$
(C,E)	$\frac{63}{8}$	0	7
B	$\frac{17}{2}$	7	0

Step 4: 合并 (C,E) 和 B

计算新簇 (C,E,B) 与其他簇之间的距离：

$$
d((C,E,B),(A,F,D)) = \frac{d((C,E),(A,F,D)) + d(B,(A,F,D))}{2} = \frac{\frac{63}{8} + \frac{17}{2}}{2} = \frac{\frac{63}{8} + \frac{68}{8}}{2} = \frac{\frac{131}{8}}{2} = \frac{131}{16}.
$$

更新后的距离矩阵：

	(A,F,D)	(C,E,B)
(A,F,D)	0	$\frac{131}{16}$
(C,E,B)	$\frac{131}{16}$	0

Step 5: 合并所有簇

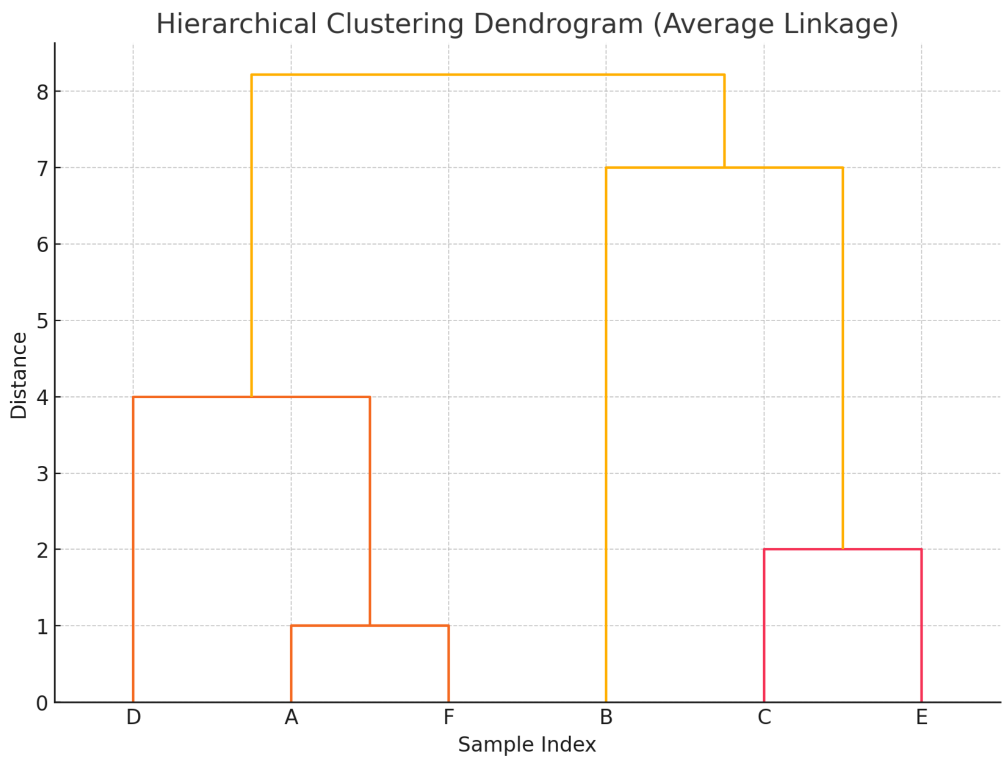
最终合并剩余的簇 (A,F,D)和 (C,E,B):

更新后的距离矩阵：

$$
\begin{array}{c|c}
 & (A,F,D,C,E,B) \\
\hline
(A,F,D,C,E,B) & 0
\end{array}
$$

树状图 (Dendrogram)

根据这些步骤，我们可以绘制树状图如下：



这是按照平均链接法生成的层次聚类树状图。图中的每个步骤都表示了簇的合并过程：

1. 首先合并了 A 和 F。
2. 接着合并了 C 和 E。
3. 然后合并了 (A, F) 和 D。
4. 接着合并了 (C, E) 和 B。
5. 最后合并了 (A, F, D)和 (C, E, B)。

每个合并步骤对应于树状图中的一个节点，节点的高度表示合并时的距离。通过这个树状图，可以直观地看到每一步的合并过程。

第三题

带有 L2 正则化项的逻辑回归目标函数

目标函数为：

$$J(\beta)=\sum_{i=1}^{m}\left[y_i(\beta^T\hat{x}_i)-\log\left(1+e^{\beta^T\hat{x}_i}\right)\right]+\lambda\|\beta\|_2^2$$

梯度推导

首先，计算不带正则化项的部分的梯度：

$$\frac{\partial}{\partial\beta}\left(y_i(\beta^T\hat{x}_i)-\log\left(1+e^{\beta^T\hat{x}_i}\right)\right)$$

对于第一个项 $y_i(\beta^T\hat{x}_i)$：

$$\frac{\partial}{\partial\beta}y_i(\beta^T\hat{x}_i)=y_i\hat{x}_i$$

对于第二个项 $-\log\left(1+e^{\beta^T\hat{x}_i}\right)$：

$$\frac{\partial}{\partial\beta}\left(-\log\left(1+e^{\beta^T\hat{x}_i}\right)\right)=-\frac{e^{\beta^T\hat{x}_i}}{1+e^{\beta^T\hat{x}_i}}\hat{x}_i=-\left(\frac{1}{1+e^{-\beta^T\hat{x}_i}}\right)\hat{x}i=-h\beta(\hat{x}_i)\hat{x}_i$$

其中 $h_\beta(\hat{x}_i)=\frac{1}{1+e^{-\beta^T\hat{x}_i}}$ 是逻辑回归的假设函数。

因此，不带正则化项的梯度为：

$$\frac{\partial}{\partial\beta}\left(y_i(\beta^T\hat{x}_i)-\log\left(1+e^{\beta^T\hat{x}_i}\right)\right)=y_i\hat{x}i-h\beta(\hat{x}_i)\hat{x}_i$$

将所有样本的梯度求和：

$$\frac{\partial}{\partial\beta}\left(\sum_{i=1}^{m}\left[y_i(\beta^T\hat{x}_i)-\log\left(1+e^{\beta^T\hat{x}i}\right)\right]\right)=\sum i=1^m\left[y_i\hat{x}i-h\beta(\hat{x}_i)\hat{x}_i\right]$$

接下来，计算正则化项的梯度：

$$\frac{\partial}{\partial\beta}\left(\lambda\|\beta\|_2^2\right)=2\lambda\beta$$

因此，带有 L2 正则化项的总梯度为：

$$\frac{\partial J(\beta)}{\partial\beta}=\sum_{i=1}^{m}\left[y_i\hat{x}i-h\beta(\hat{x}_i)\hat{x}_i\right]+2\lambda\beta$$

参数更新公式

使用梯度下降法更新参数：

$$\beta:=\beta-\alpha\frac{\partial J(\beta)}{\partial\beta}$$

其中 α 是学习率。将梯度带入参数更新公式中：

$$\beta := \beta - \alpha \left(\sum_{i=1}^{m} [y_i \hat{x} i - h\beta(\hat{x}_i)\hat{x}_i] + 2\lambda\beta \right)$$

总结

带有 L2 正则化项的逻辑回归目标函数的梯度为：

$$\frac{\partial J(\beta)}{\partial \beta} = \sum_{i=1}^{m} [y_i \hat{x} i - h\beta(\hat{x}_i)\hat{x}_i] + 2\lambda\beta$$

对应的参数更新公式为：

$$\beta := \beta - \alpha \left(\sum_{i=1}^{m} [y_i \hat{x} i - h\beta(\hat{x}_i)\hat{x}_i] + 2\lambda\beta \right)$$