山东大学 计算机科学与技术 学院

机器学习与模式识别 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202300130183 | 姓名： 宋浩宇 | | 班级：23级人工智能班 |
| 实验题目：Random Forest | | | |
| 实验学时：2 | | 实验日期：2025/4/15 | |
| 实验环境：  软件环境：  系统：Windows 11 家庭中文版23H2 22631.4317  计算软件：MATLAB 版本: 9.8.0.1323502 (R2020a)  Java 版本: Java 1.8.0\_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode  Python解释器版本：Python 3.12.6  硬件环境：  CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz  内存：32.0 GB (31.6 GB 可用)  磁盘驱动器：NVMe WD\_BLACKSN850X2000GB  显示适配器：NVIDIA GeForce RTX 4080 Laptop GPU | | | |
| 1. 实验内容   In this exercise, we’ll take a look at motivating another powerful algorithm—a  non-parametric algorithm called random forests.   1. 实验步骤 2. 配置实验环境   安装基本数学库    安装机器学习框架     1. 获取测试数据并编写简单决策树 2. 加入决策器并编写简单随机森林，训练模型 3. 结果可视化 4. 测试结果     决策树分类结果如上    随机森林分类结果如上   1. 附录：实现源代码  |  | | --- | | import numpy as np  import matplotlib.pyplot as plt  from sklearn.datasets import make\_blobs  import seaborn as sns  from sklearn.tree import DecisionTreeClassifier  from sklearn.tree import DecisionTreeClassifier  from sklearn.ensemble import BaggingClassifier  from sklearn.ensemble import RandomForestClassifier  sns.set()  X, y = make\_blobs(n\_samples=300, centers=4, random\_state=0, cluster\_std=1.0)  plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap="rainbow")  tree = DecisionTreeClassifier().fit(X, y)  plt.show()  plt.cla()  def visualize\_classifier(model, X, y, ax=None, cmap="rainbow"):      ax = ax or plt.gca()      # 绘制训练点      ax.scatter(          X[:, 0], X[:, 1], c=y, s=30, cmap=cmap, clim=(y.min(), y.max()), zorder=3      )      ax.axis("tight")      ax.axis("off")      xlim = ax.get\_xlim()      ylim = ax.get\_ylim()      # 拟合估计器      model.fit(X, y)      xx, yy = np.meshgrid(np.linspace(\*xlim, num=200), np.linspace(\*ylim, num=200))      Z = model.predict(np.c\_[xx.ravel(), yy.ravel()]).reshape(xx.shape)      # 创建颜色图      n\_classes = len(np.unique(y))      contours = ax.contourf(          xx,          yy,          Z,          alpha=0.3,          levels=np.arange(n\_classes + 1) - 0.5,          cmap=cmap,          # clim=(y.min(), y.max()),          zorder=1,      )      ax.set(xlim=xlim, ylim=ylim)  visualize\_classifier(DecisionTreeClassifier(), X, y)  tree = DecisionTreeClassifier()  bag = BaggingClassifier(tree, n\_estimators=100, max\_samples=0.8, random\_state=1)  bag.fit(X, y)  visualize\_classifier(bag, X, y)  model = RandomForestClassifier(n\_estimators=100, random\_state=0)  visualize\_classifier(model, X, y)  plt.show() | | | | |