

BIOF3 FIGCODE · SCI 绘图代码集

Python 二分类评估 (sklearn ROC + CM)

导出日期: 2026年6月27日



扫码关注微信公众号【**生信F3**】
获取文章完整内容，分享生物信息学最新知识。

分类: 生存分析 | 依赖包: scikit-learn、numpy、matplotlib

解决的生物学问题

Python 训练的二分类模型表现如何? 在 precision、recall 上的具体数值?

应用场景

- scikit-learn 模型评估
- PyTorch / TensorFlow 模型评估
- AutoML 性能对比
- Kaggle 比赛标准评估

输入数据格式

numpy 数组: y_true (0/1), y_score (0-1), y_pred (0/1)

关键参数

- threshold (默认 0.5, 可调)
- normalize = "true" / "pred" / "all" (混淆矩阵归一化)

快速开始

1. 直接在线运行

打开 [FigCode 在线绘图](#)，点击本工具卡片的"在线绘图"按钮即可用内置示例数据出图，零环境配置。

2. 本地复现

下载脚本和示例数据，本地 RStudio 运行：

```
# 下载
curl -O https://<your-site>/figcode/scripts/py-roc-confusion.R
curl -O https://<your-site>/figcode/data/py-roc-confusion.csv
```

3. 安装依赖

```
# CRAN 包
install.packages(c("scikit-learn", "numpy", "matplotlib"))

# Bioconductor 包 (如需)
# BiocManager::install(c())
```

完整代码

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import (
    roc_curve, auc, confusion_matrix, classification_report,
    ConfusionMatrixDisplay
)

# --- Demo data ---
np.random.seed(42)
n = 300
y_true = np.random.binomial(1, 0.4, n)
y_score = np.clip(y_true * 0.7 + np.random.normal(0, 0.4, n), 0, 1)
y_pred = (y_score > 0.5).astype(int)

# --- ROC ---
fpr, tpr, _ = roc_curve(y_true, y_score)
roc_auc = auc(fpr, tpr)

fig, axes = plt.subplots(1, 2, figsize=(11, 4.5))

axes[0].plot(fpr, tpr, color="#dc2626", lw=2,
             label=f"AUC = {roc_auc:.3f}")
axes[0].plot([0, 1], [0, 1], "--", color="grey")
axes[0].set_xlabel("False Positive Rate")
axes[0].set_ylabel("True Positive Rate")
axes[0].set_title("ROC Curve")
axes[0].legend(loc="lower right")
axes[0].grid(alpha=0.3)
```

```
# --- Confusion matrix ---
cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(cm, display_labels=["Negative", "Positive"])
disp.plot(ax=axes[1], cmap="Blues", colorbar=False)
axes[1].set_title("Confusion Matrix")

plt.suptitle(f"Classifier Evaluation (n = {n})", y=1.02, fontsize=12)
plt.tight_layout()
plt.savefig("plot_001.png", dpi=150, bbox_inches="tight")

# --- Print classification report ---
print(classification_report(y_true, y_pred,
                           target_names=["Negative", "Positive"]))
```

替换为自己的数据

脚本中以 `# --- Demo data ---` 标注的段落是示例数据生成代码。替换为自己的数据时，**保持列名一致**即可：

- 输入格式：numpy 数组：y_true (0/1), y_score (0-1), y_pred (0/1)
- 使用 `read.csv()` / `readRDS()` 读取本地文件

延伸阅读

- 相关教程：[Python 二分类评估 完整流程](#)
- 出现 bug？欢迎在 [FigCode 页面](#) 点击对应卡片，在评论区留言。