

```

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
import pandas as pd

X, y = make_classification(n_samples=300, n_features=2,
                           n_redundant=0, n_clusters_per_class=1)
df = pd.DataFrame(X, columns=["Feature1", "Feature2"])
df['Target'] = y

sns.set(style="white", context="talk")
g = sns.jointplot(data=df, x="Feature1", y="Feature2", hue="Target",
                  kind="kde", fill=True, palette="cubehelix",
                  alpha=0.7, height=8)
g.fig.suptitle("双变量密度图", fontsize=16, y=1.03)
plt.show()

```

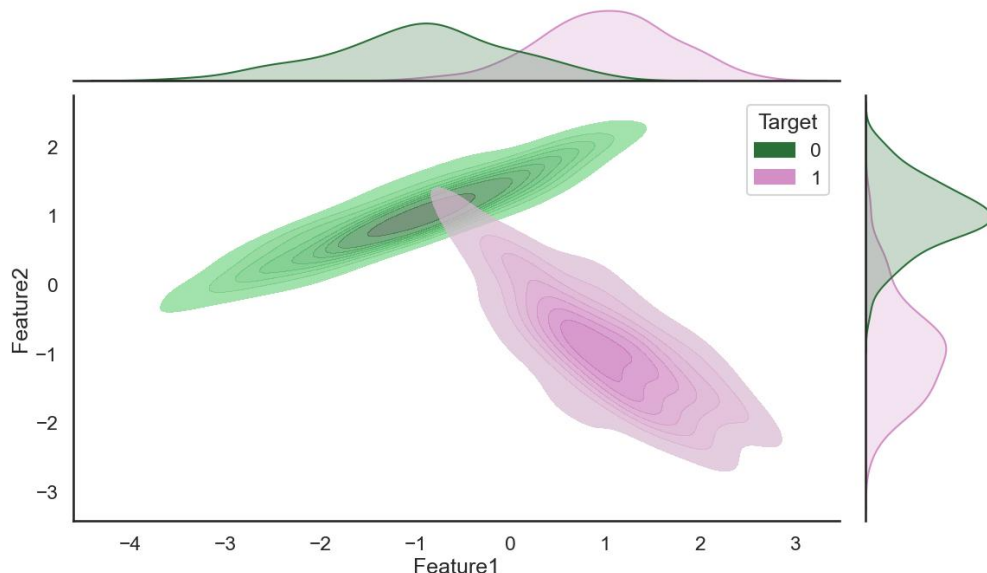


图 1 双变量密度图

```

import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import rcParams

# 设置中文字体，防止乱码（如果你有其他中文字体也可替换）
plt.rcParams['font.family'] = 'SimHei' # 黑体
plt.rcParams['axes.unicode_minus'] = False # 正常显示负号
G = nx.erdos_renyi_graph(30, 0.15)
pos = nx.spring_layout(G, seed=42)
degrees = dict(G.degree)

```

```
plt.figure(figsize=(10, 8))
nodes = nx.draw_networkx_nodes(G, pos, node_size=[v * 80 for v in
degrees.values()],
                                node_color=list(degrees.values()),
                                cmap=plt.cm.plasma, alpha=0.85)
nx.draw_networkx_edges(G, pos, alpha=0.2, edge_color='gray')
plt.title("复杂网络结构图", fontsize=16)
plt.axis("off")
plt.colorbar(nodes)
plt.show()
```

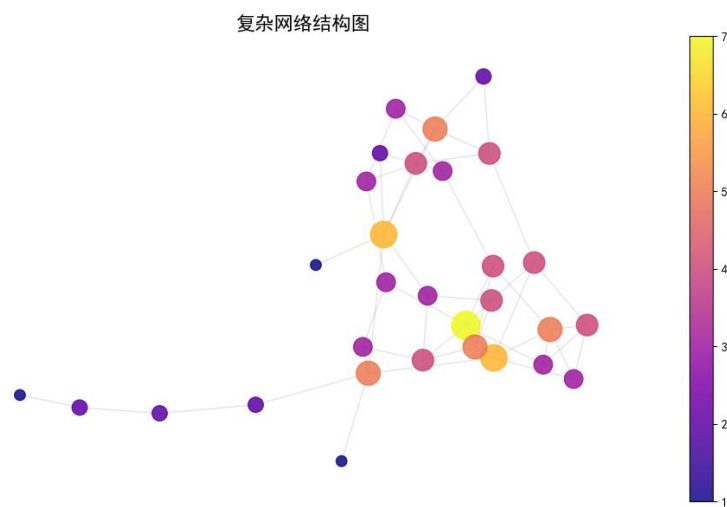


图 2 复杂网络结构图

```
import plotly.express as px

df = px.data.gapminder().query("year == 2007")
fig = px.scatter(df, x="gdpPercap", y="lifeExp", size="pop",
color="continent",
                hover_name="country", log_x=True, size_max=60,

color_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_layout(title="2007 年全球国家发展水平气泡图",
title_font_size=20,
                plot_bgcolor='white', paper_bgcolor='white')
fig.show()
```

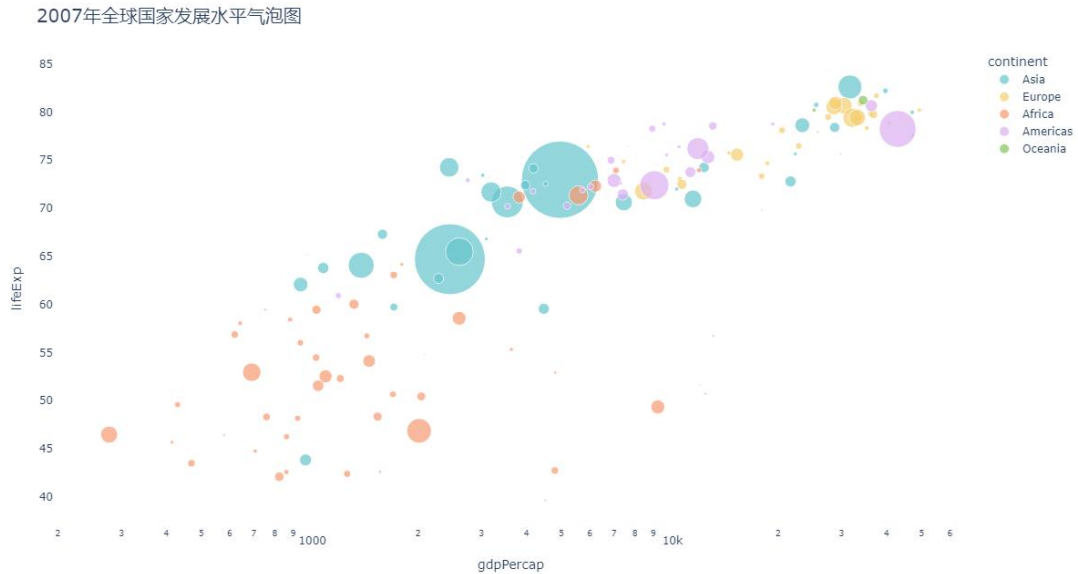


图 3 2007 年全球国家发展水平气泡图

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from matplotlib.font_manager import FontProperties

# 设置中文字体路径（SimHei 是黑体，可替换为其他中文字体）
# Windows 示例路径
font = FontProperties(fname=r"C:\Windows\Fonts\simhei.ttf", size=12)

# 创建带中文标签的 DataFrame
data = pd.DataFrame(np.random.rand(10, 12),
                    index=[f"样本{i}" for i in range(1, 11)],
                    columns=[f"特征{j}" for j in range(1, 13)])

# 设置 seaborn 风格
sns.set(context='talk', style='white')

# 创建聚类热图
g = sns.clustermap(data, cmap="coolwarm", figsize=(12, 10),
                  linewidths=.5, cbar_kws={"shrink": .5})

# 设置中文标题
g.ax_heatmap.set_title("热力聚类图", fontproperties=font,
                      fontsize=16)

# 设置行列标签字体
for label in g.ax_heatmap.get_xticklabels():
    label.set_fontproperties(font)
```

```
for label in g.ax_heatmap.get_yticklabels():
    label.set_fontproperties(font)
```

```
# 显示图像
plt.show()
```

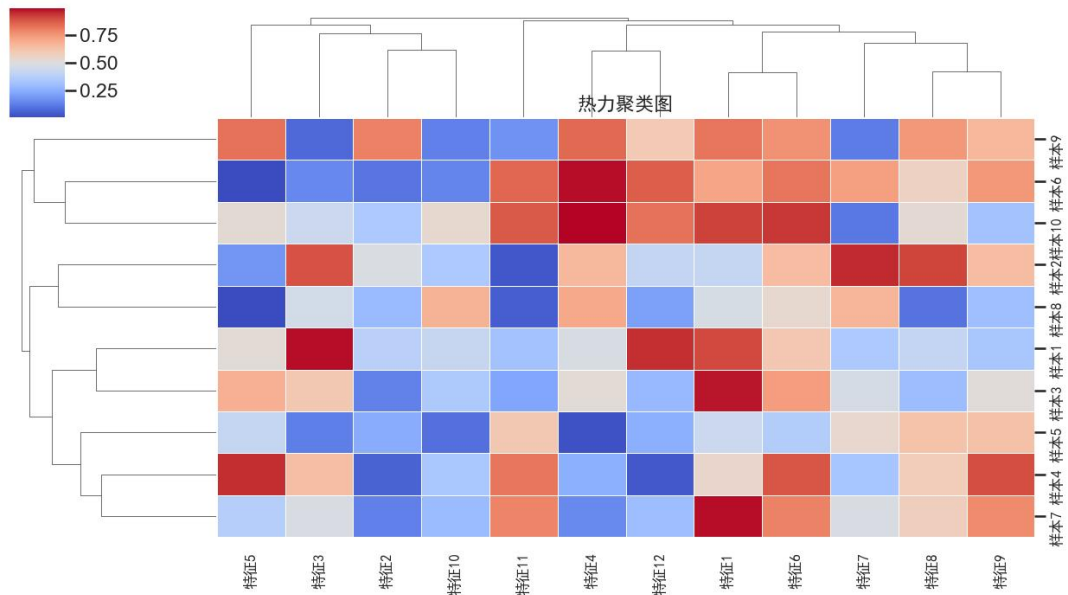


图 4 热力聚类图

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.family'] = 'SimHei' # 中文字体支持

labels = ['准确率', '召回率', 'F1', 'AUC', '鲁棒性', '泛化']
data = [0.88, 0.75, 0.81, 0.92, 0.67, 0.78]
angles = np.linspace(0, 2 * np.pi, len(labels),
                    endpoint=False).tolist()
data += data[:1]
angles += angles[:1]

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
ax.plot(angles, data, color='#1f77b4', linewidth=3)
ax.fill(angles, data, color='#1f77b4', alpha=0.25)
ax.set_thetagrids(np.degrees(angles[:-1]), labels, fontsize=13)
ax.set_title("模型性能雷达图", size=15, y=1.1)
ax.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

模型性能雷达图

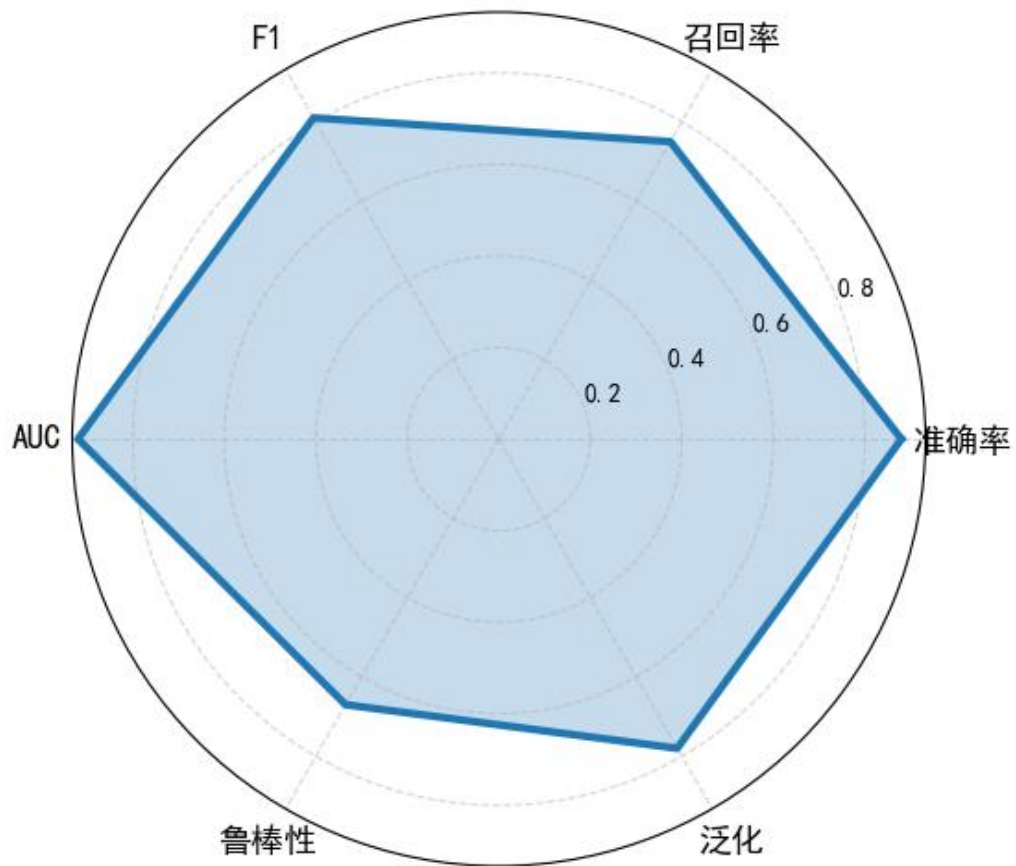


图 5 模型性能雷达图

```
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib

sns.set(style="white", palette="muted", font_scale=1.1)
df = sns.load_dataset("penguins")
g = sns.lmplot(data=df, x="flipper_length_mm", y="body_mass_g",
               col="species", hue="sex", palette="coolwarm", height=5,
               aspect=1)

g.set_titles("{col_name}")
g.fig.subplots_adjust(top=0.85)
g.fig.suptitle("Penguin Flipper Length vs Body Mass by Species",
               fontsize=18)
```

```

for ax in g.axes.flat:
    ax.set_xlabel("Flipper Length (mm)")
    ax.set_ylabel("Body Mass (g)")

plt.show()

```

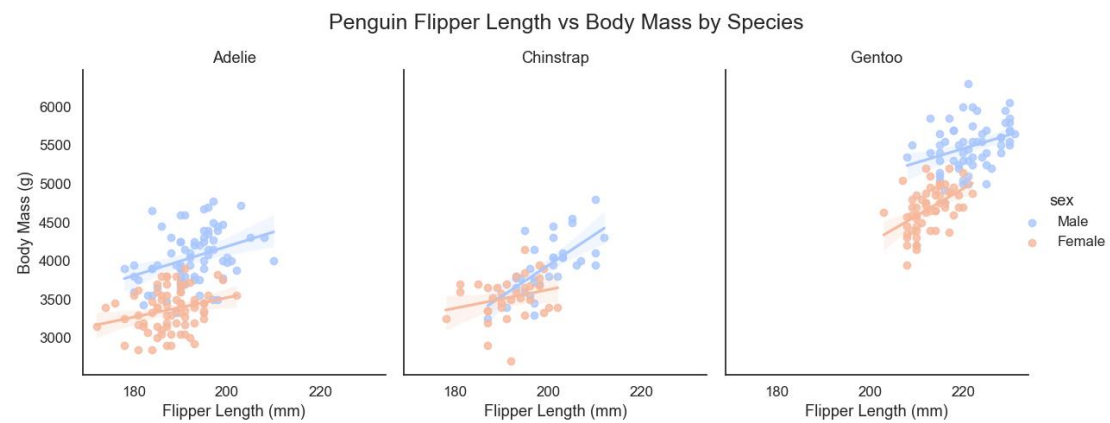


图 6 企鹅物种鳍长与体重回归关系