**Python程序设计——使用Matplotlib绘图**

**专业 软件工程 姓名 刘嘉琪 学号 1120241378**

以下是使用 matplotlib 和 scikit-learn 来展示支持向量机（SVM）分类器的绘图示例。这个示例将生成一个简单的二分类数据集，使用 SVM 进行分类后，利用matplotlib绘图包绘制可视化分类结果。

代码将生成一组线性可分的数据，训练 SVM 模型，并绘制决策边界，最终将所有数据点和决策边界使用matplotlib展示出来，具体示例如下：

（1）代码部分：

*import numpy as np*

*import matplotlib.pyplot as plt*

*from sklearn import datasets*

*from sklearn import svm*

*# 生成线性可分的数据*

*X, y = datasets.make\_blobs(n\_samples=100, centers=2, random\_state=6)*

*# 创建 SVM 分类器*

*clf = svm.SVC(kernel='rbf', C=1.0)*

*clf.fit(X, y)*

*# 创建一个网格来绘制决策边界*

*xx, yy = np.meshgrid(np.linspace(X[:, 0].min() - 1, X[:, 0].max() + 1, 100),*

*np.linspace(X[:, 1].min() - 1, X[:, 1].max() + 1, 100))*

*# 预测每个点的类*

*Z = clf.predict(np.c\_[xx.ravel(), yy.ravel()])*

*Z = Z.reshape(xx.shape)*

*# 绘制决策边界和样本点*

*plt.figure(figsize=(10, 6))*

*plt.contourf(xx, yy, Z, alpha=0.8, cmap='coolwarm')*

*plt.scatter(X[:, 0], X[:, 1], c=y, edgecolors='k', marker='o', s=100)*

*plt.title('SVM Decision Boundary', fontsize=16)*

*plt.xlabel('Feature 1', fontsize=14)*

*plt.ylabel('Feature 2', fontsize=14)*

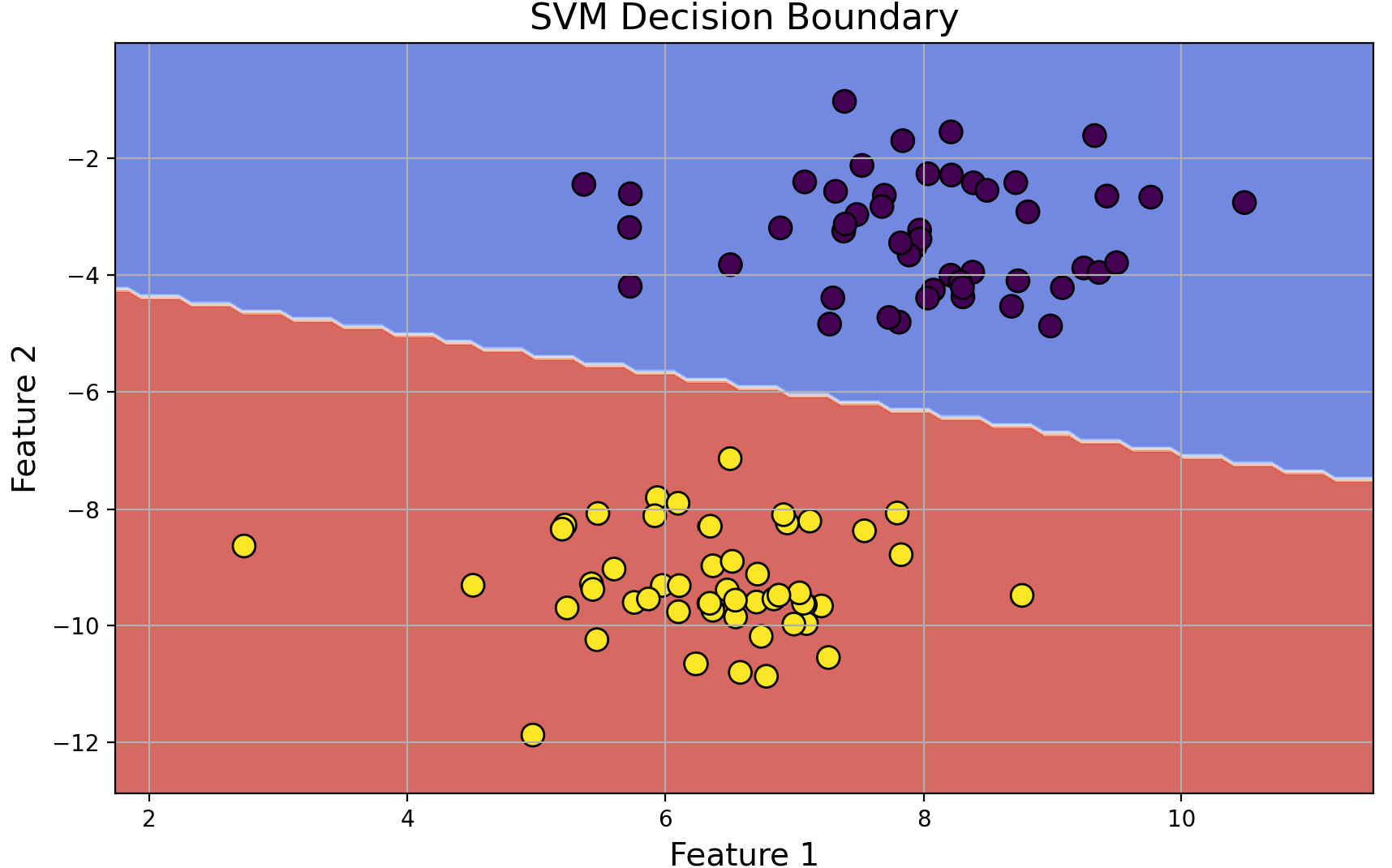
*plt.xlim(X[:, 0].min() - 1, X[:, 0].max() + 1)*

*plt.ylim(X[:, 1].min() - 1, X[:, 1].max() + 1)*

*plt.grid()*

*plt.show()*

（2）绘图展示：



**代码解释**

1. **导入必要的库**：

*import numpy as np*

*import matplotlib.pyplot as plt*

*from sklearn import datasets*

*from sklearn import svm*

* + 导入 numpy、matplotlib 和 scikit-learn 的相关模块，以便生成数据、训练模型和绘图。

1. **生成线性可分的数据集**：

*X, y = datasets.make\_blobs(n\_samples=100, centers=2, random\_state=6)*

* + 使用 make\_blobs 生成一个包含两个中心的二分类数据集，共100个样本。

1. **创建和训练 SVM 分类器**：

*clf = svm.SVC(kernel='linear', C=1.0)*

*clf.fit(X, y)*

* + 创建一个线性核的 SVM 分类器，并用生成的数据进行训练。

1. **创建决策边界的网格**：

*xx, yy = np.meshgrid(np.linspace(X[:, 0].min() - 1, X[:, 0].max() + 1, 100),*

*np.linspace(X[:, 1].min() - 1, X[:, 1].max() + 1, 100))*

* + 使用 np.meshgrid 创建网格，以便在决策边界上进行预测，同时更清晰地在图片上展示分类效果。

1. **预测每个网格点的类**：

*Z = clf.predict(np.c\_[xx.ravel(), yy.ravel()])*

*Z = Z.reshape(xx.shape)*

* + 对网格中的每个点进行预测，并将结果重塑为与网格相同的形状。

1. **绘制决策边界和样本点**：

*plt.figure(figsize=(10, 6))*

*plt.contourf(xx, yy, Z, alpha=0.8, cmap='coolwarm')*

*plt.scatter(X[:, 0], X[:, 1], c=y, edgecolors='k', marker='o', s=100)*

*plt.title('SVM Decision Boundary', fontsize=16)*

*plt.xlabel('Feature 1', fontsize=14)*

*plt.ylabel('Feature 2', fontsize=14)*

*plt.xlim(X[:, 0].min() - 1, X[:, 0].max() + 1)*

*plt.ylim(X[:, 1].min() - 1, X[:, 1].max() + 1)*

*plt.grid()*

*plt.show()*

* + 使用 plt.contourf 绘制决策边界，并用 plt.scatter 绘制样本点。设置标题、坐标轴标签和图形范围，最后显示图形。

**绘图解释：**

**在支持向量机（SVM）的可视化图中，包含以下几个关键元素：**

**1. 决策边界**

* 决策边界是 SVM 模型学习到的，用于区分不同类别样本的线或曲线。在图中，最终显示为一个明显的分割线。决策边界将特征空间划分为两个区域，每个区域对应于一个类别。样本点位于某个区域内，意味着它被分为对应的类别。

**2. 样本点**

* 点的横坐标（Feature 1）和纵坐标（Feature 2）分别表示该样本在这两个特征上的数值。
* 样本点的颜色与其所属类别一致，便于区分不同的类别。
* 样本点使用圆形标记，并且有边框颜色，以突出显示效果。

**3. 阴影区域**

* 阴影区域的颜色对应于不同的类别。在决策边界的两侧有不同颜色的阴影区域，表示不同的类别。每个阴影区域中的样本点代表被分类为该类别的样本。颜色的变化帮助用户直观地理解分类的效果。

**4. 坐标轴**

* **横轴和纵轴**：横纵坐标分别表示特征 1 和特征 2。它们的范围根据数据集的特征值而定，具体数值范围由数据集特征值的最小值和最大值决定。

**5. 标题**

* **定义**：图表的标题为“支持向量机决策边界”，和代码功能相关，以帮助快速理解图的主题和目的。

**6. 网格线**

* **定义**：图中绘制了网格线，在 x 轴和 y 轴的刻度上呈现，帮助用户更容易地读取坐标值，网格线提供参考，使得观察数据点位置和决策边界更加直观。