北京邮电大学实践课程实验报告

课程名称	**创新创业实践**	**学院**	**计算机学院（国家示范性软件学院）**	**指导老师**	
知识模块	**大数据**	**完成时间**		**2024 年 11 月 11 日**	
班级	**序号**	**学号**	**学生姓名**	**成绩**	
2023211XXX		**2023XXXXXXX**	**Yokumi**		

实验一 熟悉大数据实验平台

实验目的

- 熟悉 BDAP 平台的基本操作，包括数据加载、工作流的创建与运行，以及常见组件的使用。旨在了解工作流的整体流程。

实验内容

第一步：加载数据

- 使用 BDAP 平台的数据源组件加载台风数据集 (Asia_storm.csv)。
- 数据路径：/public/Experiment_1/Asia_storm.csv。
- 分隔符设置为逗号，文件格式为 CSV。

第二步：列投影

- 使用投影组件，选择 ADV_DATE（台风日期）和 SPEED（风速）列，提取时间与风速的对应关系。

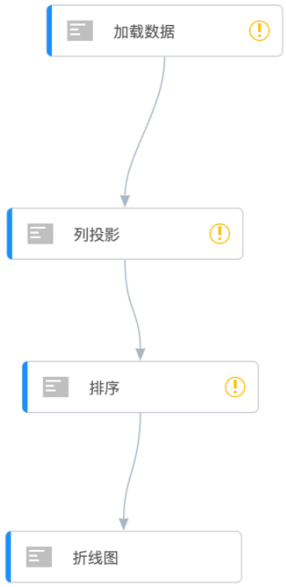
第三步：数据排序

- 使用排序组件，对 ADV_DATE 列进行升序排列（从早到晚）。

第四步：数据可视化

- 使用可视化组件绘制折线图，展示台风风速随时间变化的趋势。

实验步骤



实验结果

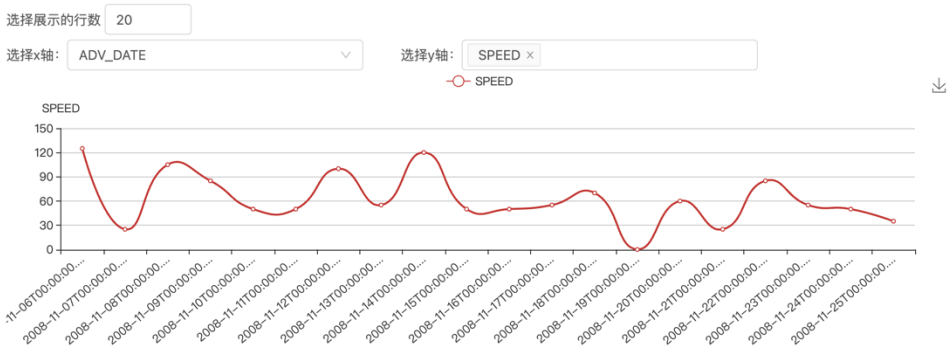


图 1-1 台风风速随时间变化的趋势

实验总结

- 通过本次实验，熟悉了 BDAP 平台的操作流程，理解了工作流的概念和数据可视化的重要性。

实验二 常见的数据预处理实验

实验目的

- 学习并掌握常见的数据预处理操作，包括分组计算、排序、数据可视化等方法，理解数据预处理在数据分析中的重要性。

实验内容

第一部分：2024 年全国空气质量数据集 (air_quality.csv)

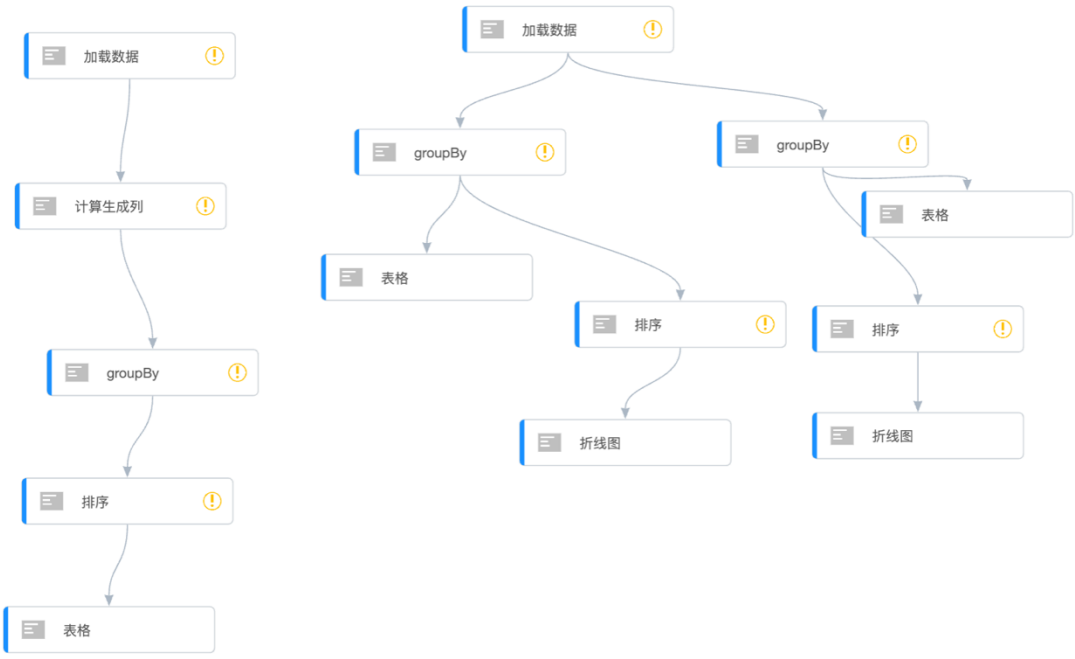
- **步骤一**：加载数据
- 使用加载数据组件，选择文件路径/public/Experiment_2/air_quality.csv，文件格式为 CSV，分隔符为逗号。
- **步骤二**：数据分组 (GroupBy)
- 使用 GroupBy 组件，以 hour 字段为聚集字段，计算 AQI 的平均值，命名为 avg_h。
- 再以 date 字段为聚集字段，计算每天的 AQI 平均值，命名为 avg_d。
- **步骤三**：数据排序
- 对 hour 列进行升序排序，按时间点展示 AQI 的变化趋势。
- 对 date 列进行升序排序，按日期展示 AQI 的日均值变化。
- **步骤四**：数据可视化
- 使用折线图组件，分别展示每小时和每日的 AQI 变化趋势。

第二部分：杭州亚运会运动员名单数据集 (HZ_game.csv)

- **步骤一**：加载数据
- 使用加载数据组件，选择文件路径/public/Experiment_2/HZ_game.csv，文件格式为 CSV。
- **步骤二**：字符串切割
- 使用计算生成列组件，提取每个运动员姓名的第一个字符（姓氏），命名为 familyName。
- **步骤三**：分组统计 (GroupBy)
- 以 familyName 为聚集字段，使用 GroupBy 组件，统计各个姓氏的出现次数，命名为 familyCount。

- **步骤四**：排序
- 对 familyCount 进行降序排序，找出最常见的姓氏。
- **步骤五**：数据可视化
- 使用柱状图展示不同姓氏的出现次数

实验步骤



实验结果

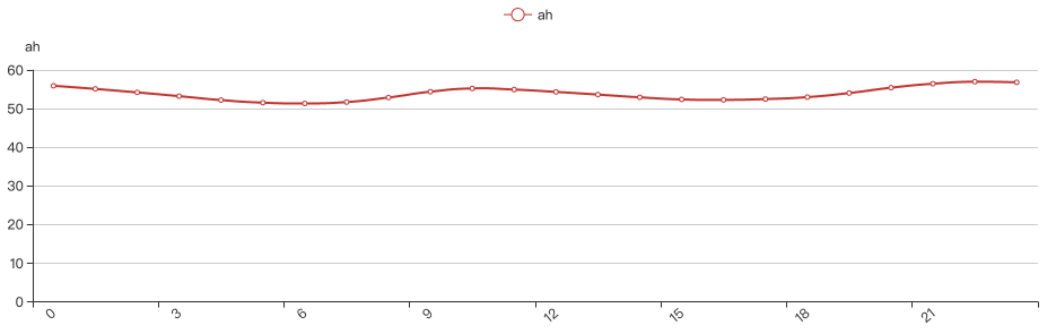


图 2-1 每小时 AQI 变化趋势

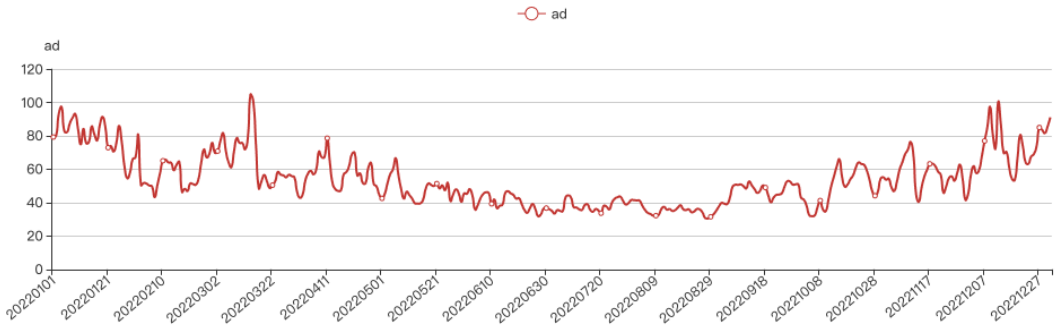


图 2-2 每日的 AQI 变化趋势

选择展示的行数 20

下载结果

family_name	family_name_count
王	84
张	58
李	56
陈	47
刘	45
杨	24
黄	24

图 2-3 不同姓氏的出现次数

实验总结

- 本实验通过数据预处理操作，掌握了分组计算和排序的常见方法，并理解了数据可视化在数据分析中的重要性。
- 数据预处理是分析的关键环节，可以显著提高数据的利用价值，有助于发现数据中的规律和异常。

实验三 决策树分类实验

实验目的

- 学习使用决策树算法进行分类任务的基本原理和方法，掌握决策树的训练、验证和测试过程，了解如何利用机器学习算法进行数据分类。
- 通过对鸢尾花数据集的分类，理解决策树的构造和分类效果。

实验内容

步骤一：加载数据

- 使用数据源组件加载鸢尾花数据集 (Iris.csv)。
- 数据路径为/public/Experiment_3/Iris.csv，文件格式为 CSV，不包含表头，分隔符为逗号。

步骤二：训练决策树模型

- 拖动决策树组件到工作流中，设置相关参数：
    - 训练字段：选择鸢尾花的四个特征（花萼长度、花萼宽度、花瓣长度、花瓣宽度）。
    - 标签字段：选择鸢尾花的种类。
    - 信息增益的选择标准为基尼系数。
    - 设置最大树深度和最小信息增益等参数，以避免过拟合。
- 将模型保存到私有文件目录下的新建文件夹中，用于后续验证和测试。

步骤三：模型验证

- 使用验证数据集进行模型验证，将训练好的模型加载到验证组件中，查看预测结果。
- 验证结果包括每个样本的预测类别和真实类别的比较。

步骤四：测试模型

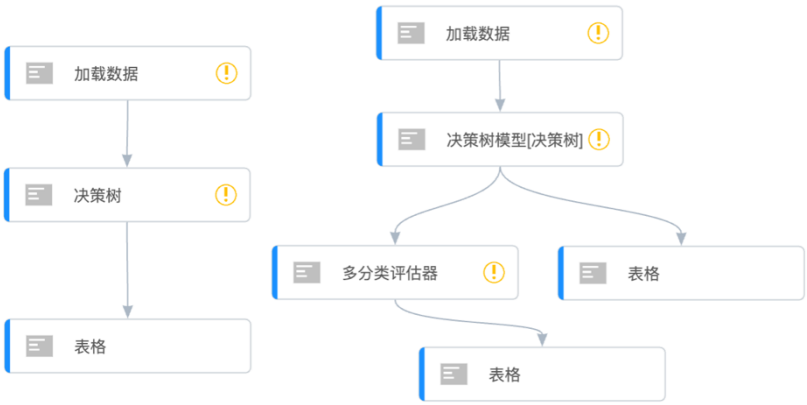
- 对测试数据进行预测，将结果与真实类别对比，计算预测准确率。
- 使用多分类评估器组件评估模型的性能，计算准确率（Accuracy）。

- 准确率计算公式：Accuracy = 分类正确数 / 样本总数。

步骤五：可视化决策树

- 使用决策树的可视化工具展示生成的树结构，解释每个节点的含义（如划分依据、条件等）。

实验步骤



实验结果

选择展示的行数 20

下载结果

_c0	_c1	_c2	_c3	_c4	indexed_c4	prediction	predictedLabel
5.1	3.5	1.4	0.2	Iris-setosa	0	0	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa	0	0	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa	0	0	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa	0	0	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa	0	0	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa	0	0	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa	0	0	Iris-setosa

图 3-1 部分预测结果

metric
1

图 3-2 分类模型准确率

实验总结

- 通过本次实验，掌握了决策树算法的基本原理和实现过程，理解了如何通过选择最佳特征进行分类。
- 学会了在 BDAP 平台上训练和评估机器学习模型的具体操作步骤，了解了验证集在防止模型过拟合中的作用。
- 决策树的可解释性使其在分类任务中具有优势，但在处理复杂数据时容易产生过拟合，需要进行剪枝等处理。

实验四 K-Means 聚类实验

实验目的

- 掌握 K-Means 聚类算法的原理和应用，了解无监督学习方法如何将数据分组。
- 使用 K-Means 算法对新能源汽车数据进行聚类分析，根据座椅数量和价格对汽车进行分组。

实验内容

步骤一：加载数据

- 使用 BDAP 平台的数据源组件加载新能源汽车数据集 (NewEnergyCar.csv)。
- 数据路径为/public/Experiment_4/NewEnergyCar.csv，文件格式为 CSV，分隔符为逗号。

步骤二：数据预处理

- 使用“填充/删除空值”组件处理缺失数据，选择“填充”模式，空白区域用列的平均值填充。
- 使用列投影组件选择所需的特征列：Seats（座椅数量）和 PriceEuro（价格）。

步骤三：K-Means 聚类

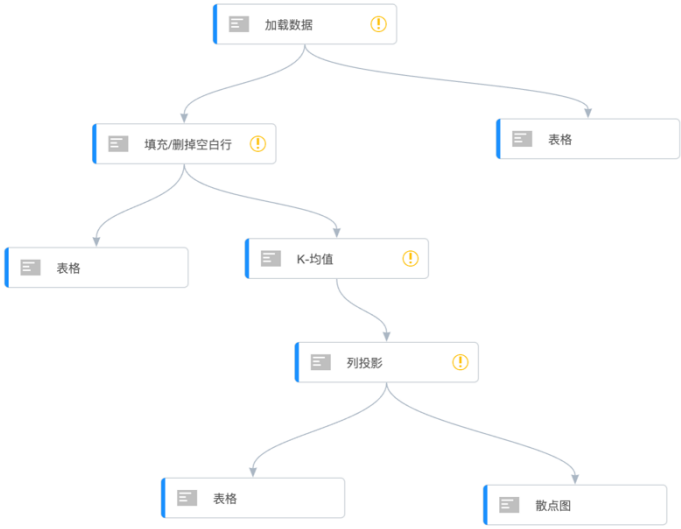
- 拖动 K-Means 组件到工作流中，设置聚类参数：
    - 聚类簇数（K）设定为 5，即将汽车分为 5 类。
    - 最大迭代次数设定为 300，收敛阈值设定为 0.001，表示在中心点变化小于 0.001 时停止迭代。
    - 训练字段选择 Seats 和 PriceEuro，标签列选择 Model（汽车型号）。
- 运行 K-Means 算法对数据进行聚类。

步骤四：数据可视化

- 使用散点图组件展示聚类结果，以 Seats 为 X 轴，PriceEuro 为 Y 轴，不同颜色表示不同的聚类类别。

- 可视化结果用于分析各类新能源汽车在座椅数量和价格方面的特征差异。

实验步骤



实验结果

选择展示的行数 20

下载结果

Model	cluster	PriceEuro	Seats
e-Up!	2	21421	4
CITIGOe iV	2	24534	4
Mii Electric	2	20129	4
e	2	32997	4
UX 300e	0	50000	5
e Advance	2	35921	4
e-NV200 Evalia	2	33246	7

图 4-1 聚类结果

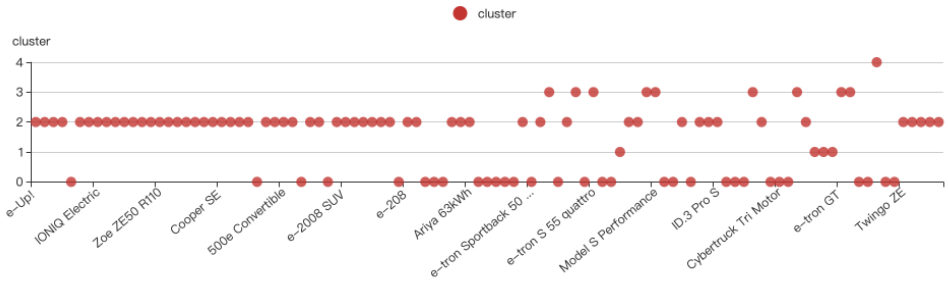


图 4-2 散点图可视化结果

实验总结

- 本次实验学习了 K-Means 聚类的原理和在实际数据中的应用，通过聚类分析能够发现数据中的内在模式。
- 通过对新能源汽车数据的聚类探索，不仅理解了聚类算法的操作步骤，还掌握了无监督学习中常见的数据分析方法。
- 数据的预处理和参数选择对聚类结果有较大影响，调节聚类簇数（K）可以更好地适应不同的分析需求。

实验五 关联规则实验

实验目的

- 学习关联规则算法的基本原理及其应用，掌握如何利用关联规则挖掘数据中的潜在关联关系。
- 通过对电影参演人员数据集的分析，发现电影演员之间的关联规则。

实验内容

步骤一：加载数据

- 使用 BDAP 平台的数据源组件加载电影参演人员数据集 (MovieCast.csv)。
- 数据路径为/public/Experiment_5/MovieCast.csv，文件格式为 CSV，分隔符为逗号。

步骤二：数据预处理

- 检查数据格式，确保数据集中包含电影名称和参演演员信息的字段。
- 使用计算生成列或投影组件，选择与关联分析相关的字段，如演员名单。

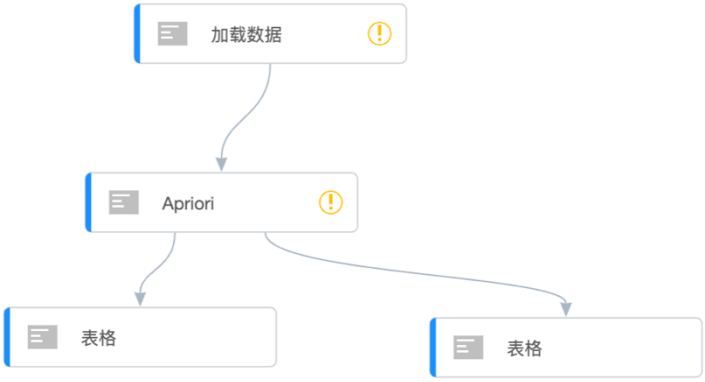
步骤三：频繁项集挖掘

- 使用关联规则组件进行频繁项集挖掘：
    - 设置最小支持度阈值（例如 0.02），表示频繁项集需要至少出现在 2%的事务中。
    - 设置最小置信度阈值（例如 0.5），用于筛选满足条件的关联规则。
- 运行算法，生成频繁项集，并基于频繁项集生成关联规则。

步骤四：关联规则的可视化与分析

- 使用关联规则可视化组件，展示生成的关联规则，包括项集的支持度和置信度。
- 对挖掘结果进行分析，找出具有较高置信度的关联规则，如哪些演员在同一部电影中出现的概率较高。

实验步骤



实验结果

选择展示的行数 20

下载结果

Item	Support
王学兵	0.016359918200409
杜江	0.016359918200409
陆毅	0.012269938650306749
张国立	0.010224948875255624
舒淇	0.028629856850715747
张一白	0.012269938650306749
杨祐宁	0.022494887525562373

图 5-1 生成的关联规则

选择展示的行数 20

下载结果

KnownItem	RelatedItem	Lift	Confidence	Support
杨幂,谢依霖	王琳	97.8	1	0.0102249488752556...
王琳	杨幂,谢依霖	97.8	1	0.0102249488752556...
郭采洁,杨幂	陈学冬	69.85714285714286	1	0.0102249488752556...
陈学冬	郭采洁,杨幂	69.85714285714286	0.7142857142857143	0.0102249488752556...
谢依霖	王琳	61.125	0.625	0.0102249488752556...
谢依霖	杨幂,王琳	61.125	0.625	0.0102249488752556...
王琳	谢依霖	61.12499999999999	1	0.0102249488752556...

图 5-2 挖掘结果分析

实验总结

- 通过本次实验，理解了关联规则算法的核心概念（支持度和置信度），并学会在 BDAP 平台上实现关联规则挖掘。
- 关联规则能够帮助揭示数据中隐藏的关联关系，在市场分析、推荐系统等领域有广泛应用。
- 数据的选择和参数的设置对关联规则挖掘的效果有显著影响，适当的支持度和置信度阈值能够提高规则的质量。

实验六 SVM 分类算法实验

实验目的

- 学习支持向量机（SVM）分类算法的基本原理及其应用，掌握 SVM 模型的训练和预测方法。
- 使用 SVM 算法对冠状动脉疾病分类数据集进行分析，实现对疾病分类的预测。

实验内容

步骤一：加载数据

- 使用 BDAP 平台的数据源组件加载冠状动脉疾病数据集 (HeartDisease.csv)。
- 数据路径为/public/Experiment_6/HeartDisease.csv，文件格式为 CSV，分隔符为逗号。

步骤二：数据预处理

- 检查数据集的完整性，处理缺失值。
- 选择合适的特征字段用于分类，如年龄、血压、胆固醇水平等。
- 使用列投影组件选择分类任务所需的字段，去除不相关的特征。

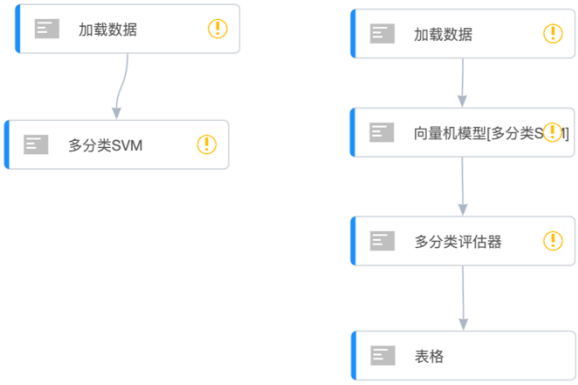
步骤三：训练 SVM 模型

- 拖动 SVM 分类组件到工作流中，设置训练参数：
    - 选择核函数类型（如线性核或径向基核）。
    - 设置惩罚参数 C，控制对误分类的惩罚强度。
    - 训练字段选择健康数据的特征，标签字段选择疾病状态。
- 运行 SVM 模型训练，将模型保存到私有文件目录中，以备验证和测试使用。

步骤四：模型验证和测试

- 使用验证集或测试集对训练好的模型进行预测，比较模型的预测结果与真实标签。
- 通过多分类评估器组件计算预测准确率和混淆矩阵，评估模型的分类性能。
- 准确率计算公式：Accuracy = 分类正确数 / 样本总数。
- 混淆矩阵可用于分析分类结果的详细情况，如真阳、假阳、真阴和假阴数量。

实验步骤



实验结果

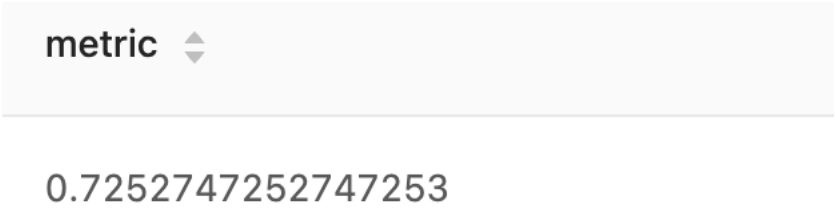


图 6-2 模型准确率

实验总结

- 通过本次实验，学会了 SVM 分类算法的基本原理，理解了超平面、支持向量和核函数在分类中的作用。
- 熟悉了 BDAP 平台上 SVM 模型的训练和测试流程，并掌握了如何评估分类模型的性能。

数据的选择和特征工程对 SVM 模型的分类效果有显著影响，调节核函数和参数 C 可以优化模型性能。

实验七 手写数字预测实验

实验目的

- 深入应用支持向量机（SVM）分类算法的基本原理及其应用，进一步掌握 SVM 模型的训练和预测方法，了解 BDAP 在线编程平台使用。
- 实现并评估一个基于支持向量机（SVM）的手写数字分类器。通过使用 sklearn 提供的数字数据集，构建分类器，并通过模型对测试集进行预测，最终评估分类器的性能，了解其在手写数字识别中的表现。

实验内容

步骤一：数据集获取

- 数据集来源为 sklearn 中的 Digits 数据集。Digits 数据集由 8x8 像素的数字图像组成。数据集的 images 属性存储每个图像的 8x8 灰度值数组。数据集的 target 属性存储每个图像代表的数字。

步骤二：必要库的导入

- sklearn 用于训练和评估模型，matplotlib 用于分类结果的可视化。

步骤三：数据加载与预处理

- 使用 sklearn.datasets.load_digits() 加载手写数字数据集。
- 通过 digits.images.reshape() 将图像数据展平，为后续的分类任务准备输入数据。

步骤四：训练集和测试集划分

- 使用 train_test_split() 函数，将数据集按 1:1 的比例划分为训练集和测试集。训练集用于训练模型，测试集用于评估模型的分类效果。

步骤五：模型训练

- 使用支持向量机分类器（svm.SVC）来训练模型。通过设置合适的 gamma 参数，优化分类器的性能，训练集上的标签（Y_train）用于训练，图像数据（X_train）作为输入特征。

步骤六：可视化结果

- 使用 matplotlib.pyplot.imshow() 函数查看前四个预测结果，帮助直观地查看分类器的表现。

步骤六：查看混淆矩阵

- 通过 metrics.classification_report() 输出分类报告，评估模型的精度、召回率、F1-score 等指标。

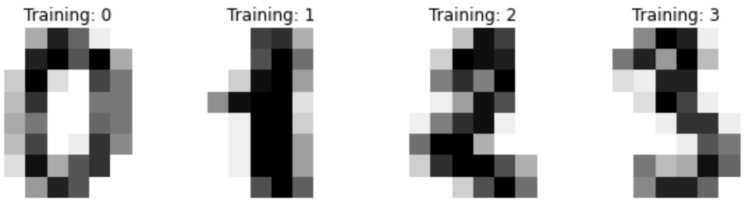
实验步骤

[1]:
```
# 导入必要的库
import matplotlib.pyplot as plt  # 用于可视化
from sklearn import datasets, svm, metrics  # 用于加载数据集、训练SVM模型、评估模型
from sklearn.model_selection import train_test_split  # 用于划分训练集和测试集
```

[2]:
```
# 加载手写数字数据集（包含图像和标签）
digits = datasets.load_digits()

# 可视化训练集中的前4个数字样本
_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))  # 创建1行4列的子图
# 将前4个数字图像展示出来
for ax, image, label in zip(axes, digits.images, digits.target):
    ax.set_axis_off()  # 关闭坐标轴显示
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")  # 显示数字图像，使用灰度色图
    ax.set_title("Training: %i" % label)  # 设置每张图像的标题为对应的标签（数字）

# 将手写数字图像数据从8x8的二维矩阵展平成一维数组，准备训练
n_samples = len(digits.images)  # 计算样本数
data = digits.images.reshape((n_samples, -1))  # 将每张8x8图像展平为64维向量
```



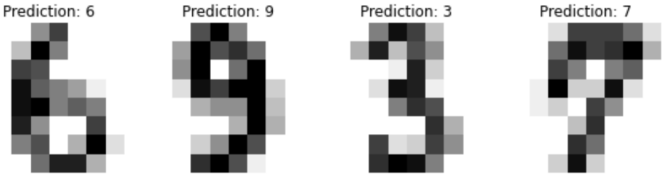
[3]:
```
# 创建SVM分类器，指定gamma参数
clf = svm.SVC(gamma=0.001)  # SVM分类器使用径向基函数(RBF)核，gamma值较小

# 按照1:1的比例划分数据集为训练集和测试集，测试集大小为50%
X_train, X_test, Y_train, Y_test = train_test_split(data, digits.target, test_size=0.5, random_state=42)
# 其中X_train, X_test分别为图像数据，Y_train, Y_test为对应的标签数据

# 用训练集对分类器进行训练
clf.fit(X_train, Y_train)  # 训练SVM分类器，使用训练数据X_train和对应标签Y_train

# 在测试集上进行预测
predicted = clf.predict(X_test)  # 用训练好的模型预测测试集数据X_test
```

[4]:
```
# 可视化测试集中的预测结果
_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))  # 创建新的子图用于显示预测结果
for ax, image, prediction in zip(axes, X_test, predicted):
    ax.set_axis_off()  # 关闭坐标轴显示
    image = image.reshape(8, 8)  # 将展平后的测试样本重新reshape为8x8的图像格式
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")  # 显示预测结果的图像
    ax.set_title(f"Prediction: {prediction}")  # 设置标题为预测的数字标签
```



[5]:
```
# 输出分类报告，评估模型在测试集上的表现
print(
    f"Classification report for classifier {clf}:\n"  # 输出模型类型
    f"{metrics.classification_report(Y_test, predicted)}\n"  # 输出分类报告，包括准确率、召回率、F1得分等指标
)
```

实验结果

```
Classification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       0.99      1.00      0.99        82
           1       1.00      1.00      1.00        89
           2       1.00      1.00      1.00        83
           3       0.99      0.97      0.98        93
           4       1.00      1.00      1.00        93
           5       0.98      0.98      0.98        99
           6       1.00      0.98      0.99        98
           7       0.97      0.99      0.98        87
           8       0.98      1.00      0.99        83
           9       0.97      0.96      0.96        92

    accuracy                           0.99       899
   macro avg       0.99      0.99      0.99       899
weighted avg       0.99      0.99      0.99       899
```

图 7-1 模型混淆矩阵

对于每个类别（数字 0 到 9），精确度、召回率和 F1 得分都接近 1.00，表明 SVM 模型在识别各个类别上都表现出色，分类器的整体准确率也达到 99%。这表明我们的模型在该数据集上有非常高的准确性。

SVM 分类器对手写数字的识别准确率极高，说明模型能很好地处理该数据集。该 SVM 模型在手写数字数据集上表现出极高的鲁棒性，对大多数样本进行了准确的分类，为了减少相似数字间的误分类，可以尝试在数据增强、特征提取或模型调参上进一步优化。

实验总结

- 通过本次实验，掌握了使用 SVM 进行图像分类的基本流程。
- 进一步理解了如何通过评估指标来全面分析模型的性能。
- 加深了对机器学习模型的理解，尤其是在分类任务中的评估方法。SVM 不仅在性能上表现出色，而且通过合适的参数调优，它在实际应用中具有广泛的适应性。
- 熟悉了 BDAP 平台上在线编程模块的使用。

考虑到模型误分类的情况，我计划在后续的实验中尝试数据增强方法，如旋转、平移、缩放图像等，以增加训练集的多样性，从而提高模型对不同手写体的鲁棒性。