电子科技大学

实验报告

课程名称： 机器学习

学　　院： 电子科技大学（深圳）高等研究院

专　　业：电子信息

指导教师：张栗粽

学生姓名： 刘文晨

学　　号：202222280328

电 子 科 技 大 学

实 验 报 告

实验四

一、实验项目名称

利用 SVM 模型实现判断病人是否属于糖尿病

二、实验学时：4 学时

三、实验目的

1. 掌握支持向量机的基本原理；
2. 了解机器学习库 sklearn。

四、实验原理

1. 支持向量机

支持向量机（Support Vector Machine, SVM）是一类按监督学习方式对数据进行二元分类的广义线性分类器，其决策边界是对学习样本求解的最大边距超平面，可以将问题化为一个求解凸二次规划的问题。具体来说就是在线性可分时，在原空间寻找两类样本的最优分类超平面。在线性不可分时，加入松弛变量并通过使用非线性映射将低维度输入空间的样本映射到高维度空间使其变为线性可分，这样就可以在该特征空间中寻找最优分类超平面。与逻辑回归和神经网络相比，SVM 可以通过核方法（Kernel Method）进行非线性分类，是常见的核学习（Kernel Learning）方法之一，在学习复杂的非线性方程时提供了一种更为清晰，更加强大的方式。

在分类问题中给定输入数据和学习目标：

$$X = \{X_1, X_2, \dots, X_n\}$$

$$y = \{y_1, y_2, \dots, y_n\}$$

其中，输入数据的每个样本都包含多个特征并由此构成特征空间：$X_i = [x_1, x_2, \dots, x_n] \in \chi$，而学习目标为二元变量$y \in \{-1,1\}$，表示负类和正类。

若输入数据所在的特征空间存在作为决策边界的超平面：

$$w^T X + b = 0$$

并将学习目标按正类和负类分开，并使任意样本的点到平面距离大于等于 1：

$$y_i(w^T X_i + b) \geq 1$$

则称该分类问题具有线性可分性，参数w和b分别为超平面的法向量和截距。

如图 1 所示，满足该条件的决策边界实际上构造了 2 个平行的超平面作为间隔边界以判别样本的分类：

$$w^T X_i + b \geq +1 \Longrightarrow y_i = +1$$

$$w^T X_i + b \leq -1 \Longrightarrow y_i = -1$$

所有在上间隔边界上方的样本属于正类，在下间隔边界下方的样本属于负类。两个间隔边界的距离$d = \frac{2}{\|w\|}$被定义为边距，位于间隔边界上的正类和负类样本为支持向量。

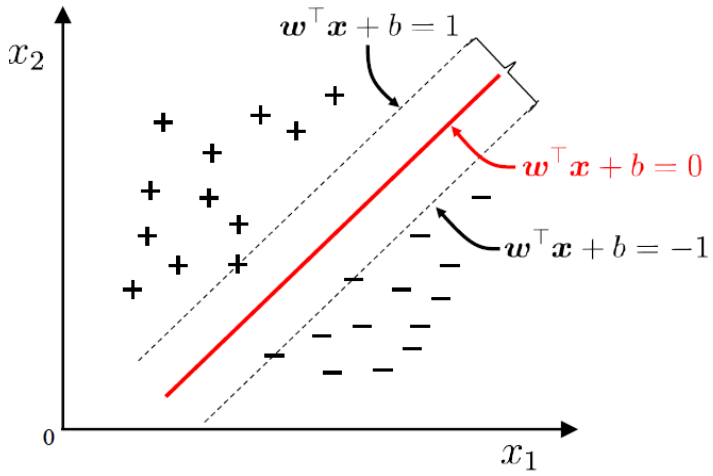

图 1 支持向量机线性分类

2. 机器学习库 sklearn

Scikit-learn（以前称为 scikits.learn，也称为 sklearn）是针对 Python 编程语言的免费软件机器学习库。它具有各种分类，回归和聚类算法，包括支持向量机，随机森林，梯度提升，k 均值和 DBSCAN，并且旨在与 Python 数值科学库 NumPy 和 SciPy 联合使用。

五、实验内容与要求

1. 数据预处理；

2. 数据标准化处理；
3. 训练SVM模型；
4. 绘制图像。

六、实验器材（设备、元器件）

处理器：Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz

Python 3.9.0

matplotlib 3.4.0

sklearn 0.0

xlrd 1.2.0

七、实验步骤

1. 数据预处理

给定的训练集和测试集是 data 文件，先把它们改为 xlsx 文件。设计一个 LoadData 函数，输入为 diabetes_train.xlsx 和 diabetes_test.xlsx 的文件路径，再调用 Python 第三方库——xlrd，将当中的数据转换为列表 x_train、y_train、x_test、y_test，最后返回这四个列表。代码如下：

```
    def LoadData(trainpath, testpath):
        file_train_path = trainpath
        file_train_xlsx = xd.open_workbook(file_train_path)
        file_train_sheet = file_train_xlsx.sheet_by_name('Sheet1')
        x_train = []
        y_train = []
        for row in range(file_train_sheet.nrows):
            x_data = []
            for col in range(file_train_sheet.ncols):
                if col < file_train_sheet.ncols - 1:
                    x_data.append(file_train_sheet.cell_value(row, col))
                else:
                    if file_train_sheet.cell_value(row, col) == 'tested_negative':
                        y_train.append(0)
                    else:
                        y_train.append(1)
            x_train.append(list(x_data))

```

```
        file_test_path = testpath
        file_test_xlsx = xd.open_workbook(file_test_path)
        file_test_sheet = file_test_xlsx.sheet_by_name('Sheet1')
        x_test = []
        y_test = []
        for row in range(file_test_sheet.nrows):
            x_data = []
            for col in range(file_test_sheet.ncols):
                if col < file_test_sheet.ncols - 1:
                    x_data.append(file_test_sheet.cell_value(row, col))
                else:
                    if file_test_sheet.cell_value(row, col) == 'tested_negative':
                        y_test.append(0)
                    else:
                        y_test.append(1)

            x_test.append(list(x_data))

        # print(x_train)
        # print(y_train)
        # print(x_test)
        # print(y_test)
        return x_train, y_train, x_test, y_test
```

2. 数据标准化处理

在将训练集和测试集导入 SVM 模型之前，要对训练集和测试集的数据进行归一化处理。代码如下：

```
    # 分别初始化对特征值和目标值的标准化器
    ss_x = StandardScaler()
    ss_y = StandardScaler()
    # 训练数据都是数值型，所以要标准化处理
    x_train = ss_x.fit_transform(x_train)
    x_test = ss_x.fit_transform(x_test)
    # 目标数据也是数值型，所以也要标准化处理
    y_train = ss_y.fit_transform(np.array(y_train).reshape(-1, 1))
    y_test = ss_y.fit_transform(np.array(y_test).reshape(-1, 1))
```

3. 训练 SVM 模型

使用 LinearSVC 分类训练，参数均未设置，默认为初始值。代码如下：

```
    # 使用 LinearSVC 分类训练
    linear_svc = LinearSVC()
    linear_svc.fit(x_train, y_train.astype('int'))
    linear_svc_predict = linear_svc.predict(x_test)
```

4. 绘制图像

在 run.py 中绘制支持向量机回归预测图像。代码如下：

```
    # 绘图
    l1, = plt.plot(y_test, color='b', linewidth=2)
    l2, = plt.plot(linear_svc_predict, color='r', linewidth=2)
    plt.legend([l1, l2], ['y_test', 'linear_svc_predict'], loc=2)
    plt.savefig('支持向量机回归预测.jpg')
    plt.show()
```

5. 实验结果

编写好代码后，运行 run.py。

运行结果如图 2 所示。

```
C:\Users\81228\AppData\Local\Programs\Python\Python39\python.exe C:/Users/81228/Desktop/资料/机器学习/实验/第四次实验/proj4/run.py
C:\Users\81228\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarni
  y = column_or_1d(y, warn=True)
C:\Users\81228\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\svm\_base.py:1225: ConvergenceWarning: Liblin
  warnings.warn(
The Accuracy of Linear SVC is 0.788659793814433

进程已结束,退出代码0
```

图 2 运行结果

支持向量机回归预测如图 3 所示。其中，蓝色折线是测试集的标签，红色折线是预测标签。

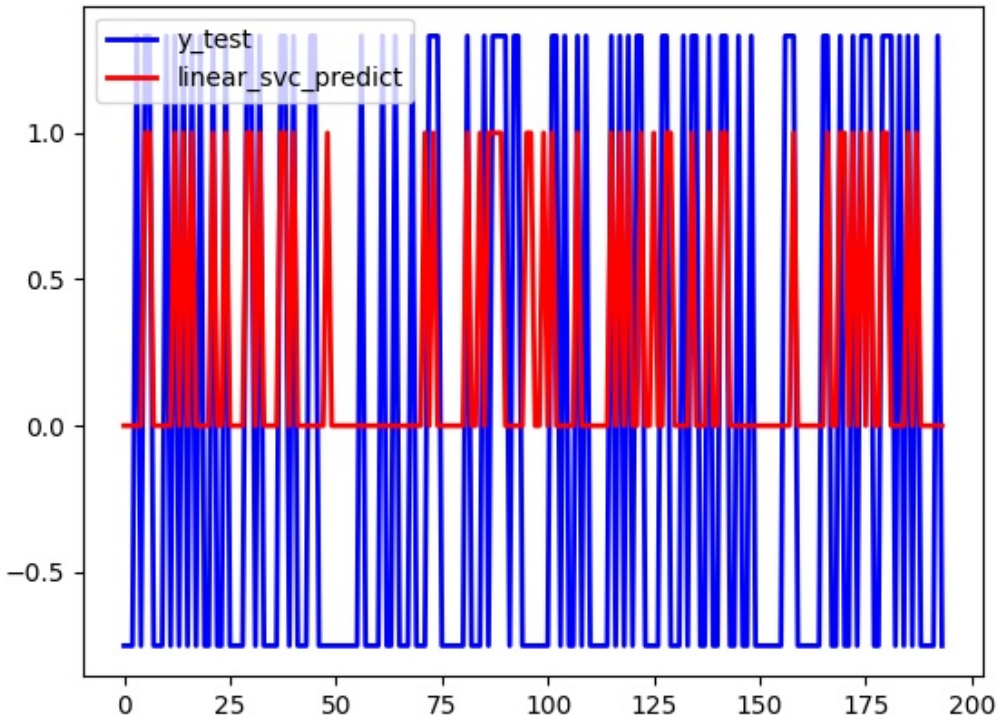

图 3 支持向量机回归预测

在本次实验中，使用线性 SVM 分类器 LinearSVC 进行分类训练，测试集的准确率为 0.788659793814433，预测成功率在 50%以上，实验成功。

八、心得体会

本实验利用 SVM 模型实现判断病人是否属于糖尿病，使用了机器学习库 sklearn 中的线性 SVM 分类器 LinearSVC 进行分类训练，测试集的准确率为 0.788659793814433，达到预期目标。通过此次实验，很好地掌握了支持向量机的基本原理，熟悉了 Python 第三方库——sklearn、matplotlib 和 xlrd 的使用。