监督学习实验报告

王嵘晟
PB17111614

1. 运行方法

环境： Python 3.7.6
main.py 完成了对数据集的读取和预处理，同时在主函数中调用了 KNN, SVM, ID3 三个类。读取数据集的路径以及相关参数可以直接在主函数中作出修改。此程序为命令行程序，在命令行中运行输出结果

2. 数据集处理

在 main.py 中，我对数据集进行了读取并处理操作。对于 absences 属性，由于缺课次数分布非常分散，所以在这里进行了归类处理，将缺课次数归为4类。对于成绩 G1,G2,G3 同样由于分布非常分散，在这里将成绩分为合格与不合格两类
随后将数据集中的字符串类型转化为整型，这里调用了 sklearn 中 preprocessing.LabelEncoder() 来实现
最后将数据集划分，随机7:3分为训练数据， 训练结果， 测试数据， 测试结果

3. KNN

算法描述：

用 KNN 算法，不需要对数据进行单独的训练，在预测时，计算每个测试点与训练点的欧几里得距离，然后排序，选择距离测试点最近的 K 个训练点，根据这些训练集合中的邻居的标签来确定该输入样例的标签。

运行结果：

当 K=30 时：

student-mat.csv	**P**	**R**	**F1**
使用所有属性	0.783	1.0	0.879
使用除去G1和G2的所有属性	0.748	0.988	0.851
student-por.csv	**P**	**R**	**F1**
使用所有属性	0.897	1.0	0.946
使用除去G1和G2的所有属性	0.877	1.0	0.934

经过多次测试发现：当逐渐增大K值时，各项指标的值先增加后减少。且使用所有属性的评分比使用除去G1, G2外所有属性的评分会高一些

4. SVM

算法描述：

SVM 算法解决凸二次规划时先用 Lagrange 乘数法得到对偶问题：
即原问题为

$$\min_{\omega,b} \frac{1}{2}||\omega^2||$$

$$s.t.\ y_i(\omega \cdot x_i + b) \geq 1,\ i = 1 \cdots N$$

转化成

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\mathbf{x_i} \cdot \mathbf{x_j^T}) - \sum_{i=1}^{N}\alpha_i\ s.t.\ \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geq 0,\ i = 1, \cdots, N$$

这里可以使用 SMO 算法，只需要求

$$\omega^* = \sum_{i=1}^{N}\alpha_i^* y_i \mathbf{x_i}$$

$$b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_i(\mathbf{x_i} \cdot \mathbf{x_j^T})$$

实现：（具体可参考代码注释）

- 每次迭代中，先找到一个违背 KKT 的变量
- 随机选取另一个与上述变量不同的变量
- 固定其他变量，通过SMO算法中的关系式得到alpha[i]与alpha[j]的迭代值
- 通过上述计算的结果求解新的bias
- 输入的数据不会被完美的线性分割，这时候我们可允许一部分数据点不满足分割超平面，并对他们施加惩罚系数C，这就是软间隔SVM。
- 不断迭代直到没有大改变为止（设定了一个小常量，小于这个值的变化将被忽略，用于加速迭代过程）
- 这里实现了线性核和高斯核，参数可以在主函数里修改

运行结果：

这里固定 C=0.5
线性核：

student-mat.csv	**P**	**R**	**F1**
使用所有属性	0.765	1.0	0.867
使用除去G1和G2的所有属性	0.739	1.0	0.85
student-por.csv	**P**	**R**	**F1**
使用所有属性	0.887	1.0	0.940
使用除去G1和G2的所有属性	0.882	1.0	0.937

高斯核：
这里令 sigma = 0.5

student-mat.csv	**P**	**R**	**F1**
使用所有属性	0.748	1.0	0.856
使用除去G1和G2的所有属性	0.714	1.0	0.833
student-por.csv	**P**	**R**	**F1**
使用所有属性	0.898	1.0	0.946
使用除去G1和G2的所有属性	0.897	1.0	0.945

经过多次测试，可以发现对于葡语成绩，高斯核的跑分明显高于线性核，但对于数学成绩却不是这样。原因可能是高斯核实现对于数学成绩数据集不够敏感。不过总体来说使用高斯核是最优的。

5. ID3（自己实现的算法）

算法描述：

计算每个属性的信息熵和信息增益，然后选择信息增益最大的属性作为决策树的分支属性，遍历这个过程来构建决策树。在预测时，遍历决策树，若遍历到的结点是非叶结点则继续，直到找到符合要求的叶子结点，作为预测结果返回。

运行结果：

student-mat.csv	**P**	**R**	**F1**
使用所有属性	0.901	0.948	0.924
使用除去G1和G2的所有属性	0.843	0.678	0.752
student-por.csv	**P**	**R**	**F1**
使用所有属性	0.940	0.981	0.960
使用除去G1和G2的所有属性	0.916	0.905	0.911

ID3 程序运行有时会报错，报错概率大概6%左右，并未解决但不影响测试。通过测试可以发现总体来说 ID3 算法的评分是最好的