

运行说明:

直接在命令行下运行。

运行 main.py, 其中包括了 KNN 算法和朴素贝叶斯分类算法
SVM 没有写。

运行结果:

```
PS F:\作业2020\人工智能作业\FBI7111609_lab2\supervise\src> python main.py
size=, split= 395 0.7
KNN result:
k=, true_case=, total_case= 3 106 118
k=, true_case=, total_case= 4 102 118
k=, true_case=, total_case= 5 104 118
k=, true_case=, total_case= 6 99 118
k=, true_case=, total_case= 7 96 118
k=, true_case=, total_case= 8 94 118
k=, true_case=, total_case= 9 98 118
k=, true_case=, total_case= 10 104 118
k=, true_case=, total_case= 11 97 118
k=, true_case=, total_case= 12 96 118
k=, true_case=, total_case= 13 94 118
k=, true_case=, total_case= 14 92 118
k=, true_case=, total_case= 15 95 118
k=, true_case=, total_case= 16 96 118
k=, true_case=, total_case= 17 94 118
k=, true_case=, total_case= 18 94 118
k=, true_case=, total_case= 19 92 118
finally we choose k=3, its accuracy rate 0.8983050847457628
Bayes result:
true_case=, total_case= 104 118
PS F:\作业2020\人工智能作业\FBI7111609_lab2\supervise\src>
```

(此次运行结果测出了 knn 算法中 k 的最佳选择是 3)

(以下结果都是使用了 G1,G2 来预测的)

knn.py:

先在 main.py 中预处理数据, 并将数据归一化(下限是 0 上限是 1)。

然后对于每个测试集中的点 通过计算与训练集每个点的欧式距离(平方和相加再开根号)
找到距离最小的点, 计算哪类 G3 更多来确定预测 G3 到底及格还是不及格。

TP=76 FP=7 TN=5 FN=30

$P = \frac{TP}{TP+FP} = 0.9157$

$R = \frac{TP}{TP+FN} = 0.7170$

$F1score = \frac{2PR}{P+R} = 0.8043$

other.py:

朴素贝叶斯分类公式:

$$p(\text{及格} | \text{某些因素}) = \frac{(p(\text{因素 1} | \text{及格}) * \dots * p(\text{因素 n} | \text{及格}) * p(\text{及格}))}{(p(\text{因素 1}) * \dots * p(\text{因素 n}))}$$

运行得出的预测结果不是特别理想：最开始 118 个测试样例预测对了 104 个，本身朴素贝叶斯模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。

而在属性相关性较小时，朴素贝叶斯性能最为良好。

所以原因之一可能是由于属性之间相关性较大。

其二就是在进行分类的时候并没有细分，只是粗略的按大于 0.5 的当作一类, 其他的另作一类。