4. Programming: Relief

本项目通过 python + numpy 实现。程序主要流程如下：

1.通过 pandas.read_csv 读取 watermelon.csv 中的数据

```
dataset = pd.read_csv('./watermelon_3.csv')
```

2.构建 python 字典类 Map，将 watermelon 数据库中数据转换为 int 和 float 型数据。映射关系如下表所示：

```
Map['浅白'],Map['青绿'],Map['乌黑']=0, 0.5, 1
Map['蜷缩'],Map['稍蜷'],Map['硬挺']=0, 0.5, 1
Map['沉闷'],Map['浊响'],Map['清脆']=0, 0.5, 1
Map['模糊'],Map['稍糊'],Map['清晰']=0, 0.5, 1
Map['凹陷'],Map['稍凹'],Map['平坦']=0, 0.5, 1
Map['硬滑'],Map['软粘']=0, 1
Map['否'],Map['是']=0, 1
```

3.构建函数实现 relief 算法，这里主要参考的是经典的 RelieF 算法：

Require: for each training instance a vector of feature values and the class value
$n \leftarrow$ number of training instances
$a \leftarrow$ number of features (i.e. attributes)
Parameter: $m \leftarrow$ number of random training instances out of n used to update W

initialize all feature weights $W[A] := 0.0$
for i:=1 **to** m **do**
  randomly select a 'target' instance R_i
  find a nearest hit 'H' and nearest miss 'M' (instances)
  for A:= 1 **to** a **do**
    $W[A] := W[A] - diff(A, R_i, H)/m + diff(A, R_i, M)/m$
  end for
end for
return the vector W of feature scores that estimate the quality of features

这里 Diff 类似度量函数的构造过程按照特征数据类型分情况处理：对于离散特征(int)型，只要不相同全部置 1，相同为 0；而对于连续性特征(float)，采用范数计算即可，这里选择负无穷范数。之后将计算的 diff 带入公式迭代计算相应特征的 Relief 值

$$\delta^j = \sum_i -diff(x_i^j, x_{i,nh}^j)^2 + diff(x_i^j, x_{i,nm}^j)^2$$

```
data, features = getdata()
relief = Relief(data)
```

4. 计算每一个特征的 Relief 值，并按照从小到大排序，结果如下：

色泽 < 触感 < 含糖量 < 密度 < 根蒂 < 敲声 < 纹理 < 脐部

```
特征排序: ['色泽' '触感' '含糖率' '密度' '根蒂' '敲声' '纹理' '脐部']
[-1, 2, 2, 2, 4, 0, 0.06548499999999997, 0.022073999999999996]
```