**中山大学数据科学与计算机学院**

**移动信息工程专业-人工智能**

**本科生实验报告**

**（2016学年秋季学期）**

课程名称：Artificial Intelligence

|  |  |  |  |
| --- | --- | --- | --- |
| 教学班级 | **14M2** | 专业（方向） | **移动互联网** |
| 学号 | **14353205** | 姓名 | **刘万里** |

# 实验题目

# 分类：必须实现1-NN，使用欧氏距离，在246个训练集文本上进行训练，再在1000个测试文本上进行预测，得到所有测试文本的情感预测结果，与标准答案（test文本第二列）进行对比，将准确率记录在实验报告上

回归：使用k-NN处理回归问题，得出所有500个测试文本属于每个情感（一共6种）的概率，计算出在验证集上的相关系数（使用validation.xlsx文件进行计算），并记录在实验报告中

# 实验内容

1. 算法原理

分类：在实验1的基础上做出适当修改，用一个vector<vector<int>>的数据结构存储每一行的字符串在OneHot矩阵中出现的下标，最后在进行测试的时候把当前测试行和每一个样本行计算距离，用这个数据结构便可以计算而不用计算OneHot矩阵上每一列的距离，以降低算法复杂度。最后找到的最小的距离就可以决定分类结果。

回归：在第一个任务的基础上，无需做出很大改动，在读取训练集的时候用一个二维矩阵保存下每一句话的6个概率。在读取到验证集的字符串时，对训练集的每一行保存下相互间的距离，用一个二维矩阵存储。最后读取完成后对验证集的每一行根据已得到的矩阵按公式计算出对应的概率，最后对6个概率进行归一化。

1. 伪代码

分类：

int MinLine = 0, MinDis = 999999;

for(int r = 0; r < 246; r++){ //与训练文本的每一行求欧式距离

if(这行的字符串个数 > 3)

//这里是自己选择的优化，因为2个字母的样本容易使所有测试样本都和它距离最小

{

得到当前行和第r行的dis;

if(dis < MinDis){

MinDis = dis;

MinLine = r;

}

}

}

输出这行的感情预测值到文本；

if(预测值和正确答案相等) RightNums++;

}

最后分类正确率 = RightNums / 测试样本个数；

回归：

for(int r = 0; r < 246; r++){ //与训练文本的每一行求欧式距离

得到距离最小值dis;

DIS[i-246][r] = dis; //得到距离矩阵

}

for(int i = 0 ; i < 500; i++) //实际上代表第 i + 246 行

{

double SUM[6], all = 0;

for(int j = 0 ; j < 6 ; j++)

{

double sum = 0;

sum = 该行对训练文本的每一行的第J种感情的概率的和;

SUM[j] = sum; //得到这种感情的概率

all += sum;

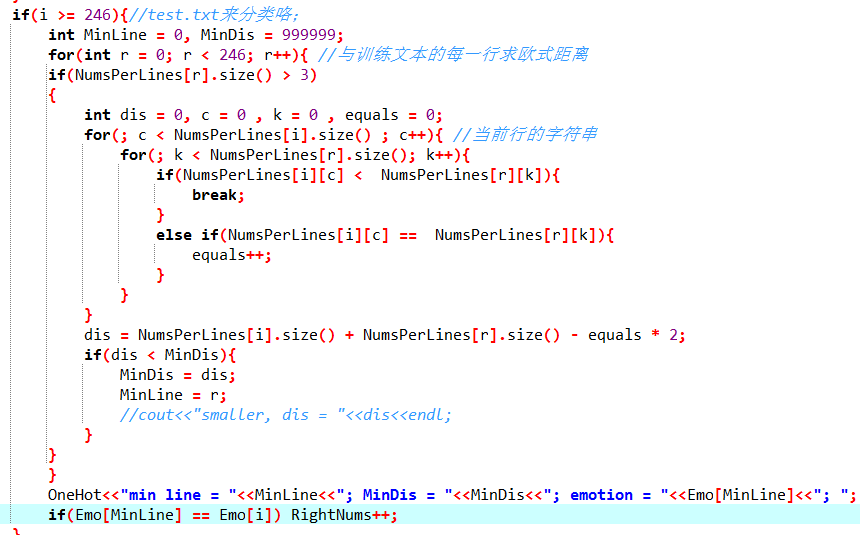
}

归一化并得到6个概率的值；

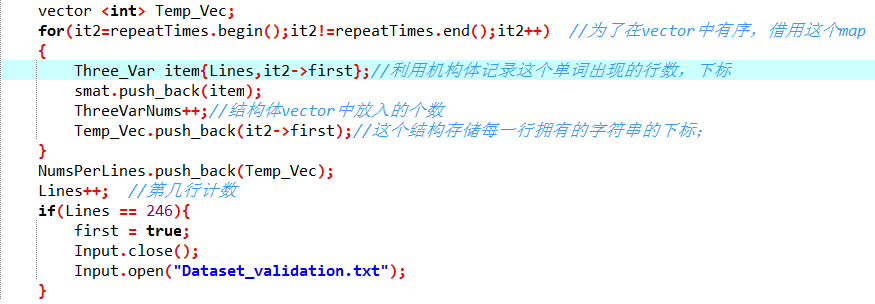
}

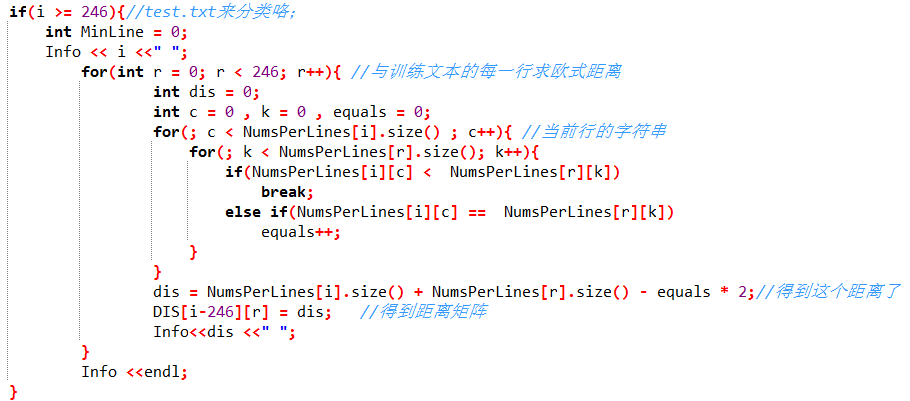
1. 关键代码截图（带注释）

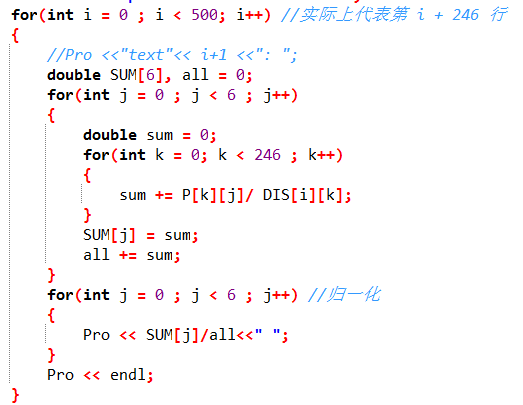
分类：



回归：







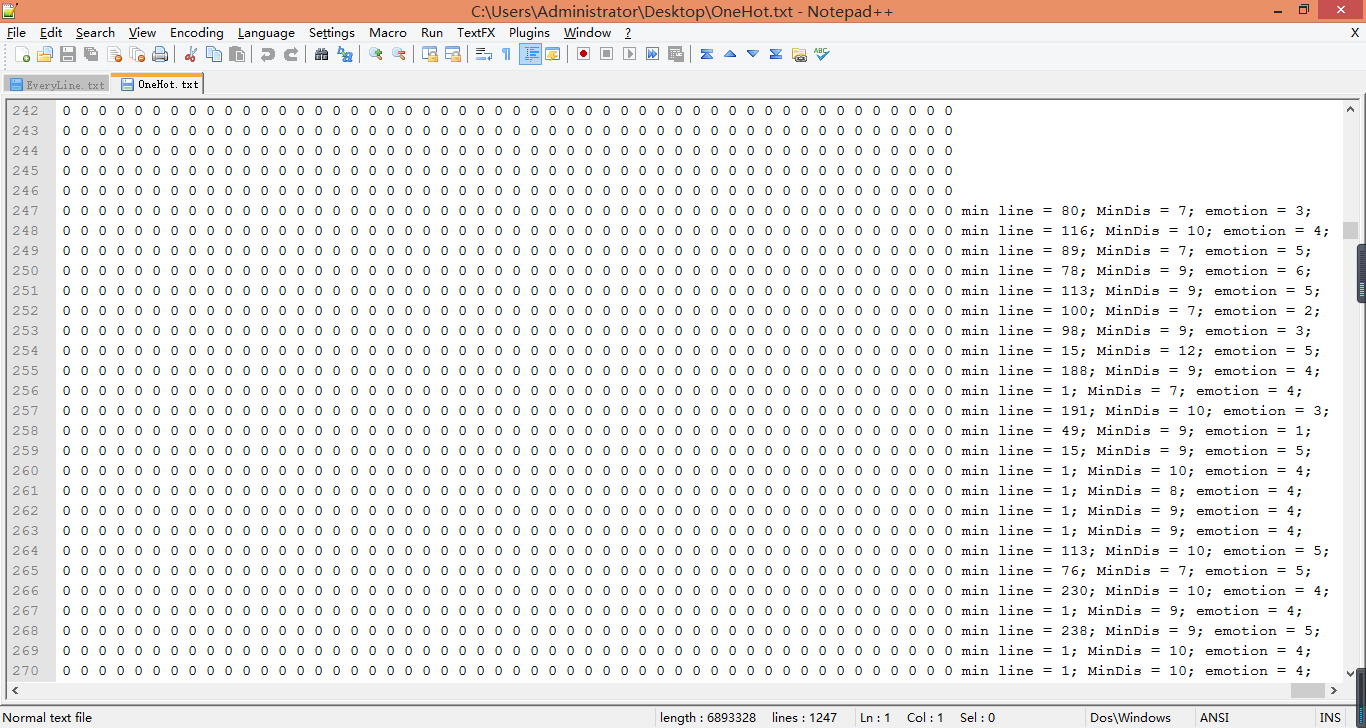
1. 创新点&优化

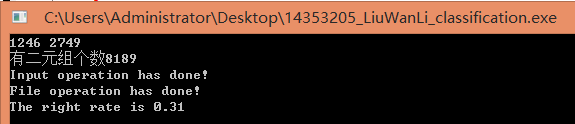
无

# 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

分类：





回归：

