

异常检测算法:KNN检测算法

相关的文献链接（请按住Ctrl键不放，然后鼠标单击下面这段文字来查看对应的文档内容）

(49条消息)【机器学习】一文详解异常检测算法: KNN_knn异常检测_Python数据挖掘的博客-CSDN博客

请先掌握本文前面的第一大点——**图解KNN异常检测算法**，后面的可看也可不看，毕竟本人也没看懂，哈哈.....

根据该文中所提到的KNN检测算法思路可知我们总体思路就是先要计算每个样本点与其它样本点之间的距离，然后再将这些距离值进行从小到大升序操作，取前面K个样本距离的均值作为异常值再将其与阈值进行比较从而判断出这个数据是否为我们检测出的那些异常点（异常数据）。

假如这里的k=3，可以先用欧式距离计算出每个点与其它点之间的距离，然后将这些距离从小到大排序（升序）后，取前3个距离值的均值作为最后的异常值，这里就以上面链接中文档中的数据为例

```
7 7 9 3
5 4 5 6
8 6 9 3
9 9 7 7
5 5 5 5
9 9 9 1
5 2 5 5
8 8 7 6
10 12 10 13
```

咳咳，由于本人对python的学习不是了解的很多，所以还是用C++来实现这个过程较妥当。参考代码在后面，由本人亲自从头写到尾，利用C++面向对象的特性完成，过程简单易懂，如果有什么不对的地方，编者水平有限，还请谅解。我的大体思路：首先用创建的Point类来存储每个数据点，然后再用Distance类来计算该点与其它点之间的距离，然后再配合vector容器将其存储并进行升序，最后取前面三个距离的均值作为异常值存入对应的对象中，之后再根据每个对象的异常值与阈值比较加上另外两个条件来判断它是否为异常数据，至少这里说的另外两个条件后面有提到，由于本人的数学知识有限，不知道如何去确定这个阈值，网上的资料本人太笨也没看懂，但最后我还是根据它们最后的异常值规律来设置阈值。首先我们先验证我的这个程序对那个KNN文档中的示例是否正确，如图所示：

```
C:\Windows\system32\cmd.exe
请输入下面9条测试用例，数据之间用空格隔开！

确诊病例 疑似病例 死亡累计 治愈出院累计

7 7 9 3
5 4 5 6
8 6 9 3
9 9 7 7
5 5 5 5
9 9 9 1
5 2 5 5
8 8 7 6
10 12 10 13

-----*数据与其对应的异常值*-----

第【1】条数据：( 7 7 9 3) 对应的异常值：2.92
第【2】条数据：( 5 4 5 6) 对应的异常值：3.01
第【3】条数据：( 8 6 9 3) 对应的异常值：3.09
第【4】条数据：( 9 9 7 7) 对应的异常值：4.17
第【5】条数据：( 5 5 5 5) 对应的异常值：3.07
第【6】条数据：( 9 9 9 1) 对应的异常值：4.26
第【7】条数据：( 5 2 5 5) 对应的异常值：3.98
第【8】条数据：( 8 8 7 6) 对应的异常值：3.24
第【9】条数据：( 10 12 10 13) 对应的异常值：9.29
```

2. 查看计算结果

输入数据并计算结果，可以查看计算结果和异常值。输入数据后，点击计算按钮，可以查看计算结果和异常值。

序号	确诊	疑似	死亡	治愈	异常值
6号	5	2	5	5	3.01
1号	5	4	5	6	3.09
4号	5	5	5	5	3.07
5号	9	9	9	1	4.26
2号	8	6	9	3	3.24
0号	7	7	9	3	4.17
7号	8	8	7	6	3.98
3号	9	9	7	7	2.92
9号	10	12	10	13	9.29

3. 计算结果

输入数据并计算结果，可以查看计算结果和异常值。输入数据后，点击计算按钮，可以查看计算结果和异常值。

序号	确诊	疑似	死亡	治愈	异常值
9号	10	12	10	13	9.29
5号	9	9	9	1	4.26
3号	9	9	7	7	6.32
2号	8	6	9	3	5.48
0号	7	7	9	3	5.29
7号	8	8	7	6	4.00
6号	5	2	5	5	3.00
1号	5	4	5	6	1.41
4号	5	5	5	5	3.07

可以看出于是与文档中的结果是一致的，在这里我们可以看出假如设定阈值为5，再用异常值与阈值比较，异常值若比阈值要大则是异常数据，否则就不是异常数据，于是就可以得出异常数据是第9条数据。

于是乎进行对原来那个流感数据进行分析，发现可以找出如下几种是异常数据的情况：

第一：如果死亡累计和治愈出院累计的数据比前一些数中最大的那个数要小，说明它是异常数据，因为累计的话字面意思就是累加，也就是只会增加，不会减少，于是这两个地方可以用这种

方法来检测它是否为异常的数据。

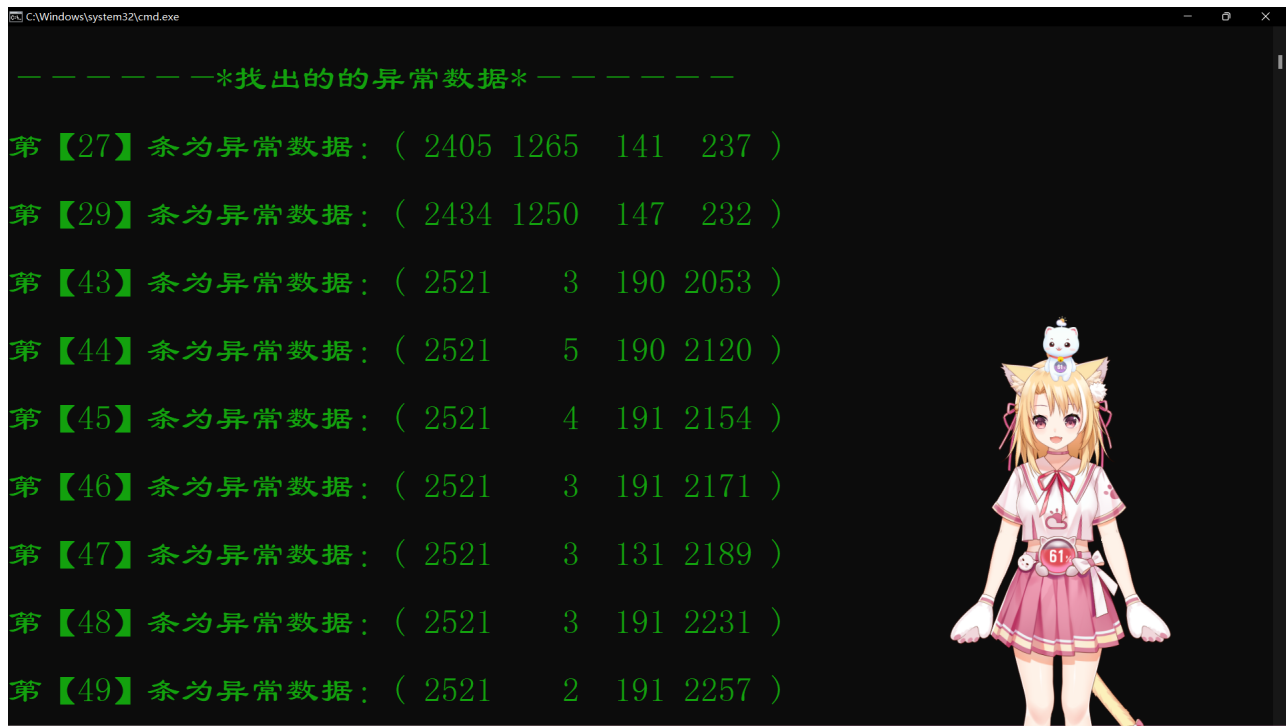
例如：这里面的第28行和第30行明显治愈出院累计出现倒退，所以是属于异常的数据。

	A	B	C	D	E
1	日期	确诊病例	疑似病例	死亡累计	治愈出院累计
23	5月11日	2265	1411	120	186
24	5月12日	2304	1378	129	208
25	5月13日	2347	1338	134	244
26	5月14日	2370	1308	139	252
27	5月15日	2388	1317	140	257
28	5月16日	2405	1265	141	237
29	5月17日	2420	1250	145	307
30	5月18日	2434	1250	147	232
31	5月19日	2437	1249	150	349
32	5月20日	2444	1225	154	395
33	5月21日	2444	1221	156	447
34	5月22日	2456	1205	158	528
35	5月23日	2465	1179	160	582
36	5月24日	2490	1134	163	667
37	5月25日	2499	1105	167	704
38	5月26日	2504	1069	168	747

第二：由于确诊病例会受到死亡人数和治愈出院人数的增多而减少，通过观察文档中的确诊数据还是比较正常的，后面的值可能是由于到了疫情稳定的情况数据变得平稳，但后面的几天肯定会有所下滑。而最有异常的情况就是疑似病例这列数据了，此时我利用KNN算法单独对它来计算它的异常值，发现结果如图所示：

```
C:\Windows\system32\cmd.exe
第【38】条数据：(2512 1005 172 828) 对应的异常值：60.00
第【39】条数据：(2514 941 175 866) 对应的异常值：51.33
第【40】条数据：(2517 803 176 928) 对应的异常值：40.00
第【41】条数据：(2520 760 177 1006) 对应的异常值：18.67
第【42】条数据：(2521 747 181 1087) 对应的异常值：11.33
第【43】条数据：(2521 3 190 2053) 对应的异常值：0.00
第【44】条数据：(2521 5 190 2120) 对应的异常值：1.33
第【45】条数据：(2521 4 191 2154) 对应的异常值：0.67
第【46】条数据：(2521 3 191 2171) 对应的异常值：0.00
第【47】条数据：(2521 3 131 2189) 对应的异常值：0.00
第【48】条数据：(2521 3 191 2231) 对应的异常值：0.00
第【49】条数据：(2521 2 191 2257) 对应的异常值：0.67
第【50】条数据：(2521 2 191 2277) 对应的异常值：0.67
第【51】条数据：(2522 739 181 1124) 对应的异常值：9.33
第【52】条数据：(2522 734 181 1157) 对应的异常值：9.33
```

可以看到异常数据所对应的异常值很小，几乎在 $[0, 2)$ 之间，于是我将阈值设置为2，假设异常值在 $[0, \text{阈值})$ 之间的就是异常数据，尽管与文中说的有点相反，但是头笨请谅解，最后通过这两个条件进行数据的检测，最后跑出来的代码如图所示：



—————*找出的异常数据*—————

第【27】条为异常数据: (2405 1265 141 237)

第【29】条为异常数据: (2434 1250 147 232)

第【43】条为异常数据: (2521 3 190 2053)

第【44】条为异常数据: (2521 5 190 2120)

第【45】条为异常数据: (2521 4 191 2154)

第【46】条为异常数据: (2521 3 191 2171)

第【47】条为异常数据: (2521 3 131 2189)

第【48】条为异常数据: (2521 3 191 2231)

第【49】条为异常数据: (2521 2 191 2257)

第【49】条为异常数据：(2521 2 191 2257)

第【50】条为异常数据：(2521 2 191 2277)

第【51】条为异常数据：(2522 739 181 1124)

第【52】条为异常数据：(2522 734 181 1157)

第【53】条为异常数据：(2522 724 181 1189)

第【54】条为异常数据：(2522 718 181 1263)

第【55】条为异常数据：(2522 716 181 1321)

第【56】条为异常数据：(2522 713 183 1403)

第【57】条为异常数据：(2522 550 184 1543)

第【58】条为异常数据：(2522 451 184 1653)

第【59】条为异常数据：(2522 351 186 1747)

第【60】条为异常数据：(2522 71 187 1944)

第【61】条为异常数据：(2522 4 189 1994)

第【62】条为异常数据：(2522 3 189 2015)

第【63】条为异常数据：(2533 668 183 1446)

第【64】条为异常数据：(2533 257 186 1821)

第【65】条为异常数据：(2533 155 187 1876)

请按任意键继续. . .

即最后的这些点就是我们要排除掉的异常数据了。

取注就是把前面的“//”给去掉，注释就是在前面加上“//”，即单行注释：

使用后面的代码前请注意参数的调整，如果是用KNN文档中的数据的话，有四个地方需要保证：

```
using namespace std;

class Distance; //前向引用声明

const long S=9; //测试用例数量 即S组数据
//感数据的话请将其设置为65，默认为65.

class Point
{
private:
    double a, b, c, d; //用于存储四个对应
    static long n; //用于记录创建了多少
    double Exception; //用于存储异常值
```

开头的S赋值为9

```
double Distance::distance(Point &x, Point &y)
{
    return sqrt(pow(fabs(x.a-y.a), 2)+pow(fabs(x.b-y.b), 2)+pow(fabs(x.c-
y.c), 2)+pow(fabs(x.d-y.d), 2)); //如果用的是那个KNN文档中的数据用例请取注这
条语句，如果已经被注释的话，然后将下面那条语句注释

    //return sqrt(pow(fabs(x.b-y.b), 2)); //如果是用流感数据的话请取注这条
语句，如果已经被注释的话，然后再将上面那条语句注释
}
```

使用上面的return语句，下面的注释掉



```
double gz=5, num(0); //设置阈值, 由程序所运行出的结果可知异常值当在[0, 2)之间的
```

```
cout<<"\n\n-----*找出的异常数据*-----"<<endl<<endl;
```

把这里的阈值属性赋值为5

```
double c_max=p[0].getC(), d_max=p[0].getD();
```

```
for(int i=1;i<S;i++)//后一个数跟前一个数比较, 如果累计的数突然变少那肯定是异常数据。  
{
```

```
if(p[i].getException()>gz)
```

是异常点

```
//if(p[i].getC()<c_max||p[i].getD()<d_max||p[i].getException()<gz) //如果异常值小于阈值,  
是异常点
```

```
{
```

```
num++;
```

```
cout<<"第【"<<p[i].getNum()<<"】条为异常数据: ( "<<p[i]<<" )"<<endl;
```

```
cout<<endl;
```

```
}
```

使用这条判断, 下面的那个判断注释

```
c_max=max(p[i].getC(), c_max);
```

```
d_max=max(p[i].getD(), d_max);
```

运行效果:

确诊病例 疑似病例 死亡累计 治愈出院累计

7 7 9 3
5 4 5 6
8 6 9 3
9 9 7 7
5 5 5 5
9 9 9 1
5 2 5 5
8 8 7 6
10 12 10 13

———*数据与其对应的异常值*———

第【1】条数据：(7 7 9 3) 对应的异常值：2.92
第【2】条数据：(5 4 5 6) 对应的异常值：3.01
第【3】条数据：(8 6 9 3) 对应的异常值：3.09
第【4】条数据：(9 9 7 7) 对应的异常值：4.17
第【5】条数据：(5 5 5 5) 对应的异常值：3.07
第【6】条数据：(9 9 9 1) 对应的异常值：4.26
第【7】条数据：(5 2 5 5) 对应的异常值：3.98
第【8】条数据：(8 8 7 6) 对应的异常值：3.24
第【9】条数据：(10 12 10 13) 对应的异常值：9.29

———*找出的异常数据*———

第【9】条为异常数据：(10 12 10 13)

请按任意键继续. . .

同理，如果是使用的流感数据的话同样有四个要注意的地方：


```

9
0  class Distance; //前向引用声明
1
2  const long S=65; //测试用例数量 即S组数据用例
   请将其设置为65，默认为65.
3
4
5  class Point
6  {
7      private:

```

这个地方赋值为65

```

double Distance::distance(Point &x, Point &y)
{
    //return sqrt(pow(fabs(x.a-y.a), 2)+pow(fabs(x.b-y.b), 2)+
    +pow(fabs(x.c-y.c), 2)+pow(fabs(x.d-y.d), 2)); //如果用的是那个
    KNN文档中的数据用例请取注这条语句，如果已经被注释的话，然后
    将下面那条语句注释

    return sqrt(pow(fabs(x.b-y.b), 2)); //如果是用流感
    话请取注这条语句，如果已经被注释的话，然后再将上面那
    注释
}

```

使用下面的这个return 语句

```

double gz=2, num(0); //设置阈值, 由程序所运行出的结果可知异常值当在[0, 2)
cout<<"\n\n-----*找出的异常数据*-----"<<endl<<endl;

double c_max=p[0].getC(), d_max=p[0].getD();

for(int i=1; i<S; i++) //后一个数跟前一个数比较，如果累计的数突然变少那肯
{

```


这里的阈值设置为2

```

for(int i=1;i<S;i++)//后一个数跟前一个数比较，如果累计的数突然变少那肯定是异常数据。
{
    //if(p[i].getException()>gz)
    if(p[i].getC()<c_max||p[i].getD()<d_max||p[i].getException()<gz) //如果异常值小于阈
    值，则说明是异常点
    {
        num++;
        cout<<"第【"<<p[i].getNum()<<"】条为异常数据：( "<<p[i]<<" )"<<endl;
        cout<<endl;
    }
}

```

使用下面这个判断语句，上面的注释掉



运行效果：

```

选择 C:\Windows\system32\cmd.exe
第【62】条数据：(2522 3 189 2015) 对应的异常值：0.00
第【63】条数据：(2533 668 183 1446) 对应的异常值：31.33
第【64】条数据：(2533 257 186 1821) 对应的异常值：113.67
第【65】条数据：(2533 155 187 1876) 对应的异常值：112.00

```

```

-----*找出的异常数据*-----
第【27】条为异常数据：( 2405 1265 141 237 )
第【29】条为异常数据：( 2434 1250 147 232 )
第【43】条为异常数据：( 2521 3 190 2053 )
第【44】条为异常数据：( 2521 5 190 2120 )
第【45】条为异常数据：( 2521 4 191 2154 )
第【46】条为异常数据：( 2521 3 191 2171 )
第【47】条为异常数据：( 2521 3 131 2189 )
第【48】条为异常数据：( 2521 3 191 2231 )
第【49】条为异常数据：( 2521 2 191 2257 )
第【50】条为异常数据：( 2521 2 191 2277 )
第【51】条为异常数据：( 2522 739 181 1124 )
第【52】条为异常数据：( 2522 734 181 1157 )
第【53】条为异常数据：( 2522 724 181 1189 )
第【54】条为异常数据：( 2522 718 181 1263 )
第【55】条为异常数据：( 2522 716 181 1321 )

```

由于数据太多，一次截不全
具体的值可以跑一下代码就知道了

代码实现如下：

```

1 #include<iostream>

```

```

2  #include<math.h>
3  #include<algorithm>
4  #include<vector>
5  #include<stdlib.h>
6  #include<iomanip>
7
8  using namespace std;
9
10 class Distance;//前向引用声明
11
12 const long S=65; //测试用例数量 即S组数据用例 如果是KNN文档中的数据请将其设置为 9 ，如果是流
    感数据的话请将其设置为65，默认为65。
13
14 class Point
15 {
16     private:
17         double a,b,c,d;//用于存储四个对应的数据
18         static long n;//用于记录创建了多少个对象
19         double Exception;//用于存储异常值 Exception
20         long num;//记录数
21     public:
22
23         friend class Distance;//设置为友元类
24
25         Point(){}
26         Point(double x,double y,double s,double n):a(x),b(y),c(s),d(n){n++;}
27
28         friend istream& operator>> (istream &in ,Point &x);
29         friend ostream& operator<< (ostream &in ,Point &x);
30         static double size(){return n;}
31         double getC(){return c;}
32         double getD(){return d;}
33         double getException() {return Exception;}
34         double setException(double x) { Exception=x; return x;}
35         long getNum(){return num;}
36         long setNum(long n){num=n; return num;}
37 };
38
39 long Point::n=0;

```

```

40
41 istream& operator>> (istream &in ,Point & x)
42 {
43     Point::n++;
44
45     in>>x.a>>x.b>>x.c>>x.d;
46
47     return in;
48 }
49
50 ostream& operator<< (ostream &ou ,Point & x)
51 {
52     ou<<setiosflags(ios::right); //设置输出格式
53
54     ou<<setw(4)<<x.a<<" "<<setw(4)<<x.b<<" "<<setw(4)<<x.c<<" "<<setw(4)<<x.d;
55
56     return ou;
57 }
58
59
60 class Distance
61 {
62     private:
63         Point x,y;
64     public:
65         Distance(){}
66         Distance(Point &a,Point &b):x(a),y(b){}
67
68         double distance(Point&,Point &); //计算两点之间的距离
69 };
70
71 double Distance::distance(Point &x,Point &y)
72 {
73     //return sqrt(pow(fabs(x.a-y.a),2)+pow(fabs(x.b-y.b),2)+pow(fabs(x.c-
74     y.c),2)+pow(fabs(x.d-y.d),2)); //如果用的是那个KNN文档中的数据用例请取注这条语句， 如果已经被
75     注释的话， 然后将下面那条语句注释
76
77     return sqrt(pow(fabs(x.b-y.b),2)); //如果是用流感数据的话请取注这条语句， 如果已经被注释的
78     话， 然后再将上面那条语句注释
79

```

```

77 }
78
79 void start()
80 {
81     cout<<"请输入下面"<<S<<"条测试用例，数据之间用空格隔开!\n\n";
82     cout<<"确诊病例   疑似病例   死亡累计   治愈出院累计\n\n";
83 }
84
85 void findException()
86 {
87     Point p[S];
88
89     long n=0; //用于配合goto语句进行计算每个点的异常值
90
91     start();
92
93     for(int i=0;i<S;i++)
94         cin>>p[i];//已进行输入符的重载
95
96     vector<double> mean;//用于计算平均值 mean
97
98     Distance ds[9];//用于存每个点与p[n]之间的距离
99
100 go_on:
101
102 //计算每个点与p[n]之间的距离存入容器mean中，自身与自身不计算
103 for(int i=0;i<S;i++)
104     if(i==n) continue;//自身与自身不计算
105     else mean.push_back(ds[i].distance(p[i],p[n]));
106
107 sort(mean.begin(),mean.end());//将所有的距离值进行升序
108
109 p[n++].setException((mean[0]+mean[1]+mean[2])/3.0); //记录异常值，假设knn算法用k=3的
    情况，存取到对应的对象中去
110
111     mean.clear(); //清除所有平均值，重新计算下一个点的异常值
112
113     if(n<=S-1) goto go_on; //如果每个点还没遍历完成则继续循环。
114
115     cout<<endl<<endl;

```

```

116
117     cout<<"————*数据与其对应的异常值*————\n"<<endl;
118
119     for(int i=0;i<S;i++)//遍历所有的异常值
120     {
121         printf("第【%2d】条数据：",p[i].setNum(i+1));
122         cout<<"("<<p[i]<<")";
123         printf("    对应的异常值：%.2lf\n\n",p[i].getException());
124     }
125
126     double gz=2,num(0); //设置阈值,由程序所运行出的结果可知异常值当在[0,2)之间的即为异常数据
127
128     cout<<"\n\n————*找出的异常数据*————"<<endl<<endl;
129
130     double c_max=p[0].getC(),d_max=p[0].getD();
131
132     for(int i=1;i<S;i++)//后一个数跟前一个数比较，如果累计的数突然变少那肯定是异常数据。
133     {
134
135         //if(p[i].getException()>gz)
136         if(p[i].getC()<c_max||p[i].getD()<d_max||p[i].getException()<gz) //如果异常值小
于阈值，则说明是异常点
137         {
138             num++;
139
140             cout<<"第【"<<p[i].getNum()<<"]条为异常数据：( "<<p[i]<<")"<<endl;
141             cout<<endl;
142
143         }
144
145         c_max=max(p[i].getC(),c_max);
146         d_max=max(p[i].getD(),d_max);
147
148     }
149
150     if(!num) cout<<"未找到异常点！\n";
151
152     cout<<endl<<endl;
153
154     system("pause");

```

```
155
156 }
157
158 int main()
159 {
160
161     findException();
162
163     return 0;
164 }
165
166 /*测试用例
167
168 表中的流感数据:
169     339 402 18 13
170     482 610 25 23
171     588 669 28 32
172     693 782 35 55
173     774 863 39 65
174     877 953 42 73
175     988 1093 48 76
176     1114 1255 56 78
177     1199 1275 59 78
178     1347 1358 66 83
179     1440 1408 75 90
180     1553 1415 82 100
181     1636 1468 91 109
182     1741 1493 96 115
183     1803 1537 100 118
184     1897 1510 103 121
185     1960 1523 107 134
186     2049 1514 110 141
187     2136 1486 112 152
188     2177 1425 114 168
189     2227 1397 116 175
190     2265 1411 120 186
191     2304 1378 129 208
192     2347 1338 134 244
193     2370 1308 139 252
194     2388 1317 140 257
```

195	2405	1265	141	237
196	2420	1250	145	307
197	2434	1250	147	232
198	2437	1249	150	349
199	2444	1225	154	395
200	2444	1221	156	447
201	2456	1205	158	528
202	2465	1179	160	582
203	2490	1134	163	667
204	2499	1105	167	704
205	2504	1069	168	747
206	2512	1005	172	828
207	2514	941	175	866
208	2517	803	176	928
209	2520	760	177	1006
210	2521	747	181	1087
211	2521	3	190	2053
212	2521	5	190	2120
213	2521	4	191	2154
214	2521	3	191	2171
215	2521	3	131	2189
216	2521	3	191	2231
217	2521	2	191	2257
218	2521	2	191	2277
219	2522	739	181	1124
220	2522	734	181	1157
221	2522	724	181	1189
222	2522	718	181	1263
223	2522	716	181	1321
224	2522	713	183	1403
225	2522	550	184	1543
226	2522	451	184	1653
227	2522	351	186	1747
228	2522	71	187	1944
229	2522	4	189	1994
230	2522	3	189	2015
231	2533	668	183	1446
232	2533	257	186	1821
233	2533	155	187	1876


```
234
235 KNN文档中的测试数据:
236 7 7 9 3
237 5 4 5 6
238 8 6 9 3
239 9 9 7 7
240 5 5 5 5
241 9 9 9 1
242 5 2 5 5
243 8 8 7 6
244 10 12 10 13
245
246 */
```

好的，大一小白的思路就写到这了，感谢您的观看……………