Main文件下：

文件Instance.java:

样例类，包含样例的属性和类别，每从数据文件中读取一行，就创建一个该类的对象。同时为了能够让其作为MapReduce的Key或Value，实现了hadoop的序列化（实现WritableComparable接口）。

其中重载了3个构造方法，一个无参构造，2个有参构造。

readFields方法和write方法实现了hadoop的序列化。

compareTo方法实现了两个对象之间的大小比较。

toString方法实现其对象的格式化输出。

文件DistanceAndLabel.java:

可看为一种数据结构，仅仅用来保存类别和距离，同样也实现了hadoop的序列化。

文件KNNMain.java：

程序入口。负责创建一个MapReduce任务，然后设置任务的一些配置和所需参数。

具体的：

1. 设置任务的名字。
2. 设置任务的入口。
3. 设置任务所需参数，如K的取值，训练集、测试集路径，输出路径。
4. 设置任务运行所需的Mapper实现类和Reducer实现类，以及其对应的输入输出的Key、Value类型
5. 如果有Combiner则设置Combiner对应的实现类。

文件KNNMapper.java:

KNN算法的Mapper实现类，实现了setup方法和map方法。

其中setup方法就是获取测试集，将其加载到集合testSet中，这个方法在Map阶段map方法执行前运行，且只运行一次。

map方法按行读取训练集，计算每一个训练样例与每一个测试样例的距离，输出（测试样例， 与训练样例的距离和其类别）

文件KNNCombiner.java:

Combiner类的实现、实现了setup方法和reduce方法，获取map的输出，将相同的测试样例，和其与每个训练样例的距离和类别，合并到一起作为key，value输入。。

其中：

setup方法获取K值

reduce方法将找到测试样例目前对应的k个最近邻。输出（测试样例， 目前计算出的K个最近邻）

文件KNNReducer.java:

Reducer类的实现，实现了setup方法、reduce方法、valueOfMostFrequent方法。

输入为（测试样例， 所有combiner输出的K\*n个最近邻）其中n为combiner的个数，根据输入文件的大小来确定。

其中valueOfMostFrequent方法为统计K个最近邻中类别最多的，用来得到测试样例的最终类别。

setup方法获取K值

reduce方法找到测试样例对应的k个近邻，并通过valueOfMostFrequent方法确定测试样例的类别，输出包含类别的测试样例。

util文件下：

文件 Distance.java:

只包含一个EuclideanDistance静态方法，用来计算两个样例之间的欧式距离。

如果需要使用其它距离的定义，可以在这里添加计算距离的方法。

文件Sort.java:

包含getNearest和indexOfMax两个静态方法。

getNearest用来在一个集合中找到距离最小的K个对象，并返回一个新集合。

indexOfMax获取集合最大值的索引。