数据挖掘：分类与聚类

学院：计算机学院
专业：计算科学与技术
姓名：王学博
学号：2120171073

一. 实验环境

电脑：64 位 CPU：Intel5 Memory：8G
系统：Ubuntu14.04
语言：python

二. 数据集

由于使用的计算机是单机，没有安装集群，在实验的时候选择了较小的数据集 https://www.kaggle.com/c/titanic/data（泰坦尼克号全体人员信息）。

数据集的结构：

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

三. 实验步骤及结果

实验分别采用了两种方法进行分类和聚类操作，方法是：

3.1 数据预处理

泰坦尼克号的人员信息有缺失值，特别是在年龄这一列的数据内容缺失是非常影响实验结果的，所以要对数据进行补充。对数据进行分类和聚类的的时候，有一些数据的信息可能对实验结果没有关系，采取了删掉的方法，减少计算的复杂度，也增强了数据预测的准确性。处理后的数据如下：

3.2 分类模型——决策树分类

1. 方法简介

决策树是一种用于对实例进行分类的树形结构。决策树由节点（node）和有向边（directed edge）组成。节点的类型有两种：内部节点和叶子节点。其中，内部节点表示一个特征或属性的测试条件（用于分开具有不同特性的记录），叶子节点表示一个分类。

构造了一个决策树模型，以它为基础来进行分类将是非常容易的。具体做法是，从根节点开始，地实例的某一特征进行测试，根据测试结构将实例分配到其子节点（也就是选择适当的分支）；沿着该分支可能达到叶子节点或者到达另一个内部节点时，那么就使用新的测试条件递归执行下去，直到抵达一个叶子节点。当到达叶子节点时，我们便得到了最终的分类结果。

2. 实验结果

采用决策树分类，在测试集上的准确率为 77%

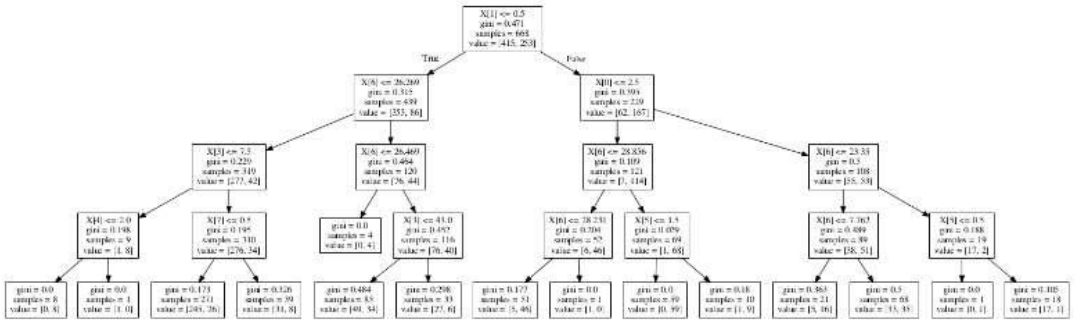
```
('The accuracy of decision tree is', 0.7727272727272727)
             precision    recall  f1-score   support

          0       0.83      0.82      0.82       269
          1       0.68      0.69      0.68       149

avg / total       0.77      0.77      0.77       418

('The cross_val_score of decision tree is', array([0.78114478, 0.79461279, 0.76094276]))
```

可视化结果如图所示：



3.3 分类模型——随机森林分类

1. 方法简介

随机森林顾名思义，是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。随机森林可以用于分类和回归。

在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类，然后看看哪一类被选择最多，就预测这个样本为那一类。

2. 实验结果

```
('The accuracy of random forest classifier is', 0.7894736842105263)
             precision    recall  f1-score   support

          0       0.85      0.82      0.84       276
          1       0.68      0.73      0.70       142

avg / total       0.79      0.79      0.79       418

('The cross_val_score of decision tree is', array([0.77104377, 0.78787879, 0.76430976]))
```

3.4 聚类方法——K 均值聚类

1. 方法简介

K 均值聚类算法是先随机选取 K 个对象作为初始的聚类中心。然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。一旦全部对象都被分配了，每个聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有（或最小数目）对象被重新分配给不同的聚类，没有（或最小数目）聚类中心再发生变化，误差平方和局部最小。

先随机选取 K 个对象作为初始的聚类中心。然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。一旦全部对象都被分配了，每个聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是以下任何一个：

1)没有（或最小数目）对象被重新分配给不同的聚类。

2)没有（或最小数目）聚类中心再发生变化。

3)误差平方和局部最小。

2. 实验结果

聚类的评价指标使用 Calinski-Harabaz Index，Calinski-Harabasz 分数值 ss 的数学计算公式是：

$$s(k) = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1}$$

其中m为训练集样本数，k为类别数。B_k为类别之间的协方差矩阵，W_k为类别内部数据的协方差矩阵。tr为矩阵的迹。

也就是说，类别内部数据的协方差越小越好，类别之间的协方差越大越好，这样的 Calinski-Harabasz 分数会高，对于我们的实验，得到的 Calinski-Harabasz 的分数结果如下图所示

```
**********************************************************************
1201.13285936
```

3.5 聚类方法——DBSCAN 聚类

1. 方法简介

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的聚类算法。与划分和层次聚类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。

具体算法步骤如下：

1) 解析样本数据文件

2) 计算每个点与其他所有点之间的欧几里德距离

3) 计算每个点的 k-距离值，并对所有点的 k-距离集合进行升序排序，输出的排序后的 k-距离值

4) 将所有点的 k-距离值，在 Excel 中用散点图显示 k-距离变化趋势

5) 根据散点图确定半径 Eps 的值

6) 根据给定 MinPts=4，以及半径 Eps 的值，计算所有核心点，并建立核心点与到核心点距离小于半径 Eps 的点的映射

7) 根据得到的核心点集合，以及半径 Eps 的值，计算能够连通的核心点，

并得到离群点

8) 将能够连通的每一组核心点，以及到核心点距离小于半径 Eps 的点，都放到一起，形成一个簇

9) 选择不同的半径 Eps，使用 DBSCAN 算法聚类得到的一组簇及其离群点，使用散点图对比聚类效果

3. 实验结果

评价指标与 K-means 一样，结果如下图所示