作业三：分类与聚类分析

学号：2620170056 姓名：相铮

一、数据源：

选择第三个数据集：[https://www.kaggle.com/c/titanic/data]

二、实验要求：

1、使用分类模型（至少2个）对数据集进行挖掘；

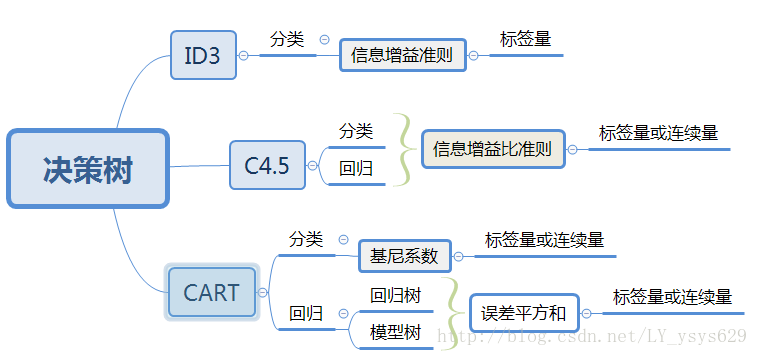
2、对挖掘结果进行可视化，并解释其意义；

3、使用聚类方法（至少2种）对数据集进行分析；

4、对挖掘结果进行可视化，并解释其意义。

1. 分类分析
2. 决策树

CART 是在给定输入X条件下输出随机变量Y的条件概率分布的学习方法。CART二分每个特征（包括标签特征以及连续特征），经过最优二分特征及其最优二分特征值的选择、切分，二叉树生成，剪枝来实现CART算法。对于回归CART树选择误差平方和准则、对于分类CART树选择基尼系数准则进行特征选择，并递归调用构建二叉树过程生成CART树。 决策树的经典算法包括ID3、C4.5、CART算法，其应用领域及所使用的准则，如下图所示。

本实验提取了提取每个人的“Sex”,“Embarked”，“PcClass”，“Sibsp”，“Fare”和“Age”六维特征。

实验代码：

#预处理

attr= ['Pclass','Sex','Age','SibSp','Parch','Embarked']

train = train\_data[attr]

train\_label = train\_data['Survived']

test = test\_data[attr]

test\_r = test\_label['Survived']

train['Age'].fillna(train['Age'].mean(),inplace=True)

train['Embarked'].fillna(train['Embarked'].value\_counts().index[0],inplace = True)

test['Age'].fillna(test['Age'].mean(),inplace=True)

test['Embarked'].fillna(test['Embarked'].value\_counts().index[0],inplace = True)

#对类别型特征进行转化,成为特征向量

vec=DictVectorizer(sparse=False)

\_train=vec.fit\_transform(train.to\_dict(orient='record'))

\_test=vec.fit\_transform(test.to\_dict(orient='record'))

#单一决策树进行模型训练以及预测分析

param\_dtc = {'min\_samples\_leaf':range(5,31,5)}

dtc=DecisionTreeClassifier(criterion='gini',splitter='best')

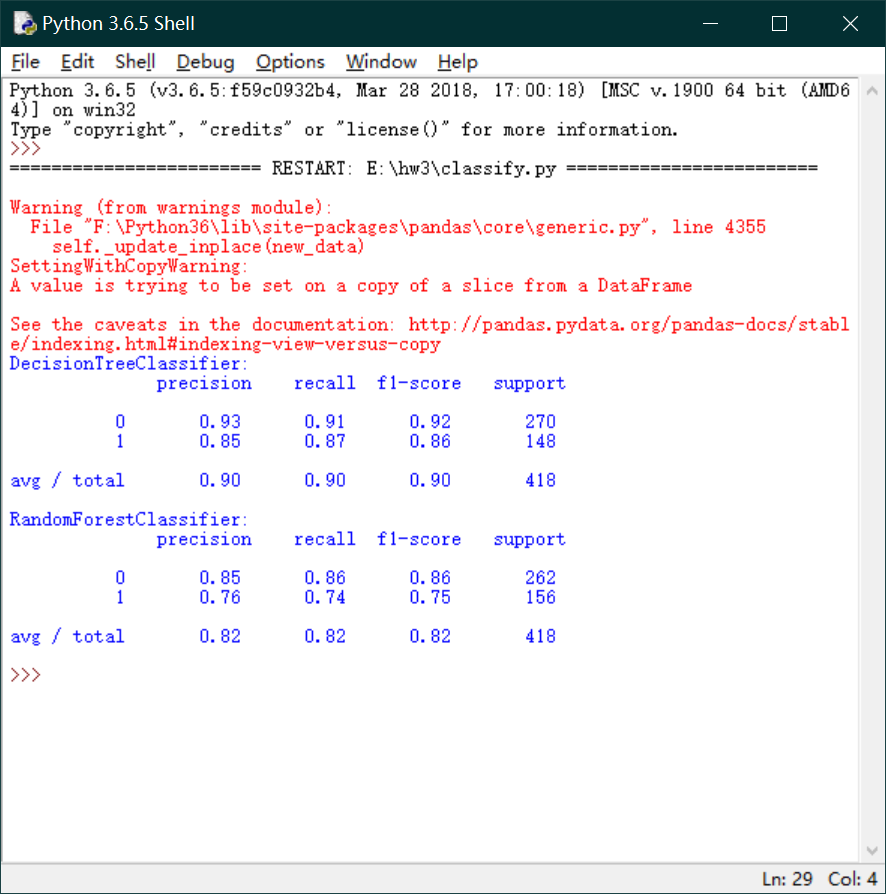
grid\_dtc = GridSearchCV(estimator=dtc,param\_grid=param\_dtc,cv=5)

grid\_dtc.fit(\_train,train\_label)

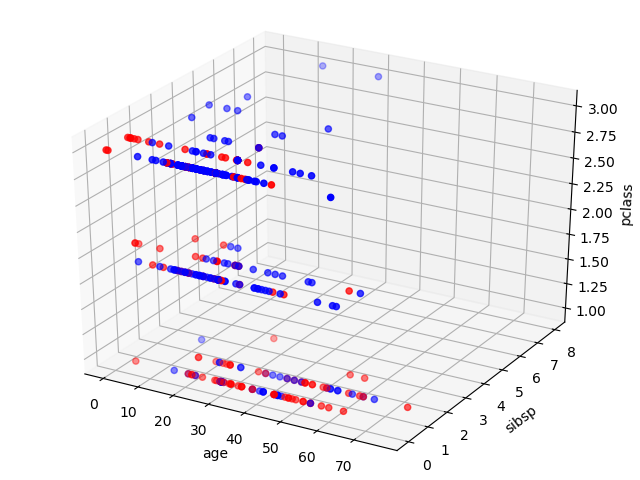
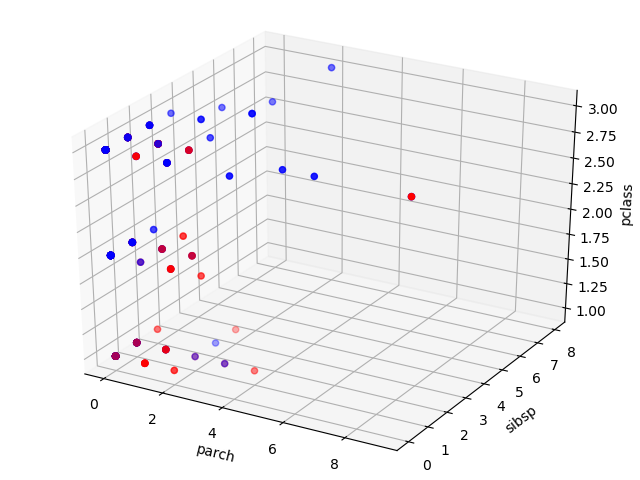
dtc\_pred=grid\_dtc.predict(\_test)

print('DecisionTreeClassifier:\n',classification\_report(dtc\_pred,test\_r))

print('DecisionTreeClassifier:\n',classification\_report(dtc\_pred,test\_r),file=result)

实验结果：预测结果的精确度、召回率、F1指数和支持率输出如下

parch,sibsp,pclass散点图 age,sibsp,pclass散点图



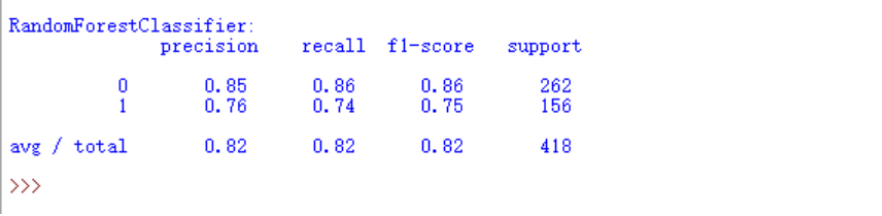
1. 决策森林

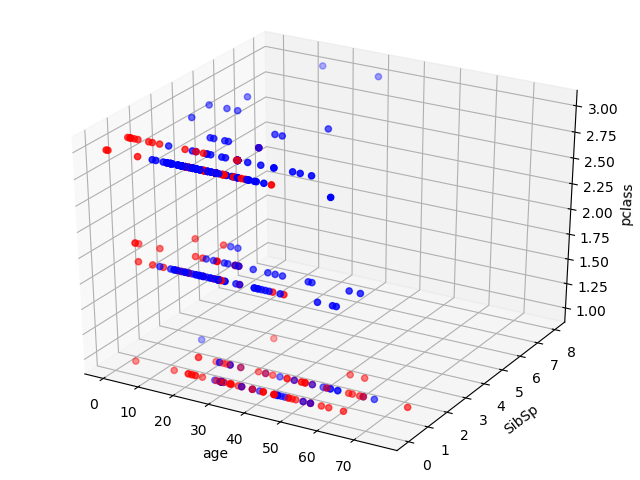
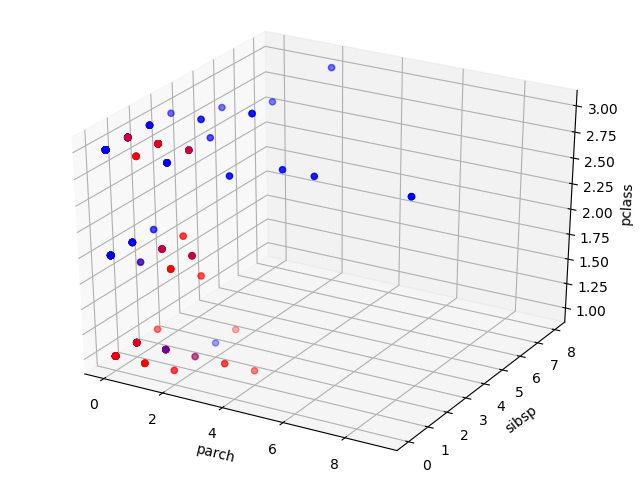
作为新兴起的、高度灵活的一种机器学习算法，随机森林（Random Forest，简称RF）拥有广泛的应用前景。随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树，而它的本质属于机器学习的一大分支——集成学习（Ensemble Learning）方法。随机森林的名称中有两个关键词，一个是“随机”，一个就是“森林”。“森林”我们很好理解，一棵叫做树，那么成百上千棵就可以叫做森林了，这样的比喻还是很贴切的，其实这也是随机森林的主要思想--集成思想的体现。

决策森林的特点：

* 在当前所有算法中，具有极好的准确率
* 能够有效地运行在大数据集上
* 能够处理具有高维特征的输入样本，而且不需要降维
* 能够评估各个特征在分类问题上的重要性
* 在生成过程中，能够获取到内部生成误差的一种无偏估计
* 对于缺省值问题也能够获得很好得结果

实验结果：预测结果的精确度、召回率、F1指数和支持率输出如下





parch,sibsp,pclass散点图 age,sibsp,pclass散点图

1. 聚类分析

聚类分析采用了k均值聚类（K-means）与均值漂移（Mean Shift）两种模型对数据集进行训练与预测。

1. k均值聚类（K-means）
2. means算法是硬[聚类算法](https://baike.baidu.com/item/%E8%81%9A%E7%B1%BB%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/K-means/_blank)，是典型的基于原型的[目标函数](https://baike.baidu.com/item/%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0" \t "https://baike.baidu.com/item/K-means/_blank)聚类方法的代表，它是数据点到原型的某种距离作为优化的目标函数，利用函数求极值的方法得到迭代运算的调整规则。K-means算法以[欧式距离](https://baike.baidu.com/item/%E6%AC%A7%E5%BC%8F%E8%B7%9D%E7%A6%BB" \t "https://baike.baidu.com/item/K-means/_blank)作为相似度测度，它是求对应某一初始聚类中心向量V最优分类，使得评价指标J最小。算法采用[误差平方和](https://baike.baidu.com/item/%E8%AF%AF%E5%B7%AE%E5%B9%B3%E6%96%B9%E5%92%8C" \t "https://baike.baidu.com/item/K-means/_blank)准则函数作为聚类准则函数，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把得到紧凑且独立的簇作为最终目标。

算法过程如下：

1）从N个文档随机选取K个文档作为[质心](https://baike.baidu.com/item/%E8%B4%A8%E5%BF%83" \t "https://baike.baidu.com/item/K-means/_blank)

2）对剩余的每个文档测量其到每个[质心](https://baike.baidu.com/item/%E8%B4%A8%E5%BF%83" \t "https://baike.baidu.com/item/K-means/_blank)的距离，并把它归到最近的质心的类

3）重新计算已经得到的各个类的[质心](https://baike.baidu.com/item/%E8%B4%A8%E5%BF%83" \t "https://baike.baidu.com/item/K-means/_blank)

4）迭代2～3步直至新的质心与原质心相等或小于指定阈值，算法结束

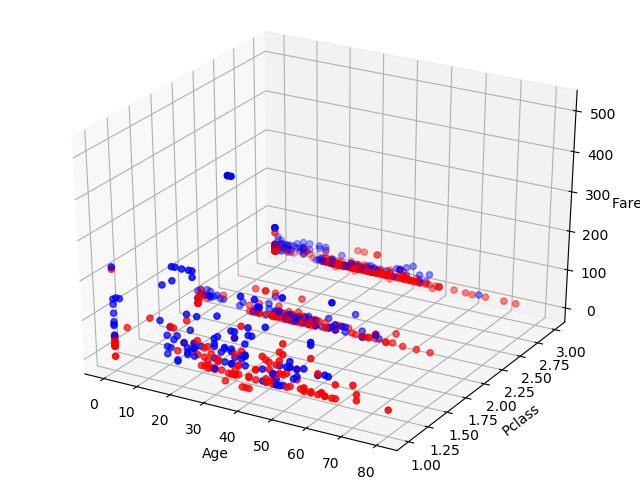
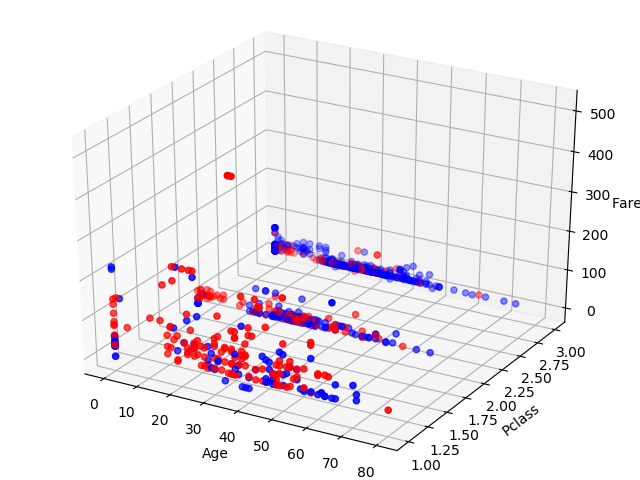
1. 均值漂移（Mean Shift）

均值漂移算法是一种基于密度梯度上升的非参数方法，通过迭代运算找到目标位置，实现目标跟踪。它显著的优点是算法计算量小，简单易实现,很适合于实时跟踪场合；但是跟踪小目标和快速移动目标时常常失败，而且在全部遮挡情况下不能自我恢复跟踪。通过实验提出应用核直方图来计算目标分布，证明了均值漂移算法具有很好的实时性特点。

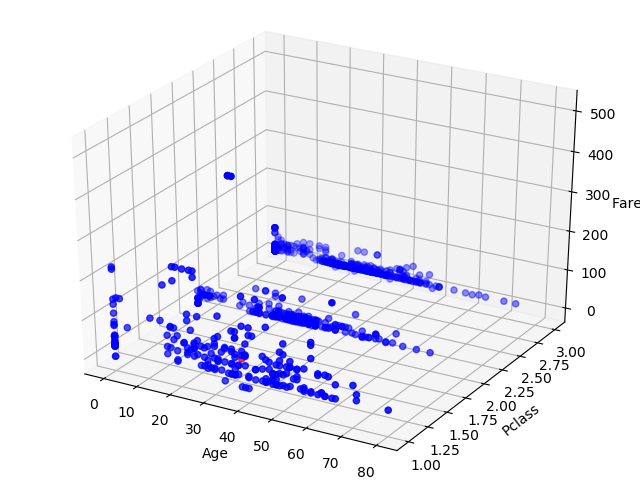
算法思想：设想在一个有N个样本点的特征空间，初始确定一个中心点center，计算在设置的半径为D的圆形空间内所有的点（xi）与中心点center的向量，计算整个圆形空间内所有向量的平均值，得到一个偏移均值。将中心点center移动到偏移均值位置。重复移动，直到满足一定条件结束。

1. 实验结果

图一 图二



正确的分类如图一；L-means聚类模型结果如图二；Meanshift算法结果如图三。

图三

预测结果的精确度、召回率、F1指数和支持率输出如下

