**机器学习实验报告1**

**数据处理方法**

首先把数据里前后几组的数据打印出来，并且观察其中的属性和我们所要的结果的关联性。

然后我们发现 Age、Fare Cabin 等三个字段的数据有所缺失。用其他数据平均值补全Embarked是港口，用最多的数据补全，cabin太多缺失难以补全。

**代码**

import pandas as pd

from sklearn.feature\_extraction import DictVectorizer

from sklearn.tree import DecisionTreeClassifier

train\_data = pd.read\_csv('./train.csv')

test\_data = pd.read\_csv('./test.csv')

print(train\_data.info())

print('-'\*30)

print(train\_data.describe())

print('-'\*30)

print(train\_data.describe(*include*=['O']))

print('-'\*30)

print(train\_data.head())

print('-'\*30)

print(train\_data.tail())

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

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

train\_data['Fare'].fillna(train\_data['Fare'].mean(), *inplace*=True)

test\_data['Fare'].fillna(test\_data['Fare'].mean(),*inplace*=True)

print(train\_data['Embarked'].value\_counts())

train\_data['Embarked'].fillna('S', *inplace*=True)

test\_data['Embarked'].fillna('S',*inplace*=True)

features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']

train\_features = train\_data[features]

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

test\_features = test\_data[features]

dvec=DictVectorizer(*sparse*=False)

train\_features=dvec.fit\_transform(train\_features.to\_dict(*orient*='records'))

print(dvec.feature\_names\_)

clf = DecisionTreeClassifier(*criterion*='entropy')

clf.fit(train\_features, train\_labels)

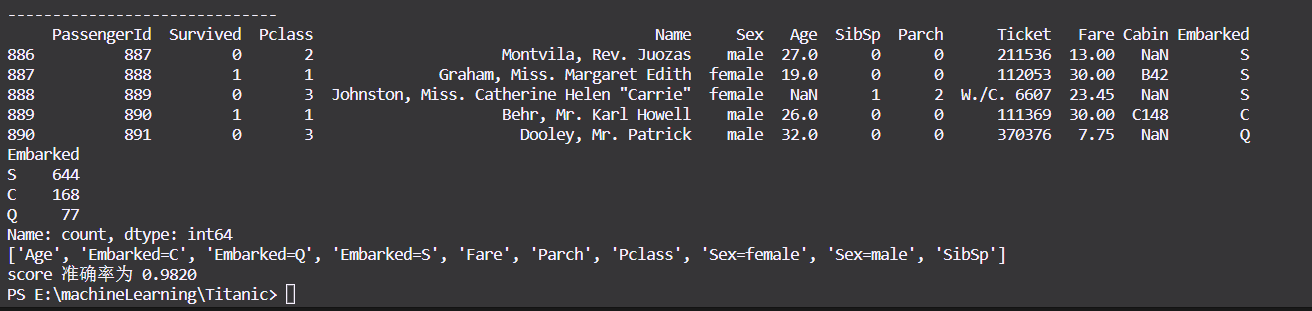
test\_features=dvec.transform(test\_features.to\_dict(*orient*='records'))

pred\_labels = clf.predict(test\_features)

acc\_decision\_tree = round(clf.score(train\_features, train\_labels), 6)

print(*u*'score 准确率为 %.4lf' % acc\_decision\_tree)

**结果截图**



**决策树算法描述**

1. 导入所需的库：导入 pandas 库用于数据处理，导入 DictVectorizer 用于特征向量化，导入 DecisionTreeClassifier 用于构建决策树分类器。
2. 数据加载：使用 pandas 的 read\_csv 函数加载训练数据和测试数据。
3. 数据探索：使用一系列方法探索数据集，包括 info()、describe() 和 head() 函数。这些函数用于查看数据集的基本信息、统计特征和数据样本。
4. 数据清洗：对缺失值进行处理。使用平均年龄填充年龄中的缺失值，使用票价的均值填充票价中的缺失值，使用登录最多的港口填充登录港口的缺失值。
5. 特征选择：根据问题和数据集选择需要的特征列。选择了'Pclass'、'Sex'、'Age'、'SibSp'、'Parch'、'Fare' 和 'Embarked' 这些特征。
6. 特征向量化：使用 DictVectorizer 将特征向量化，将分类变量转换为数值型特征。
7. 构造决策树分类器：使用 DecisionTreeClassifier 构造一个基于信息熵的决策树分类器。
8. 决策树训练：使用训练数据集的特征和标签进行训练，调用决策树分类器的 fit 方法。
9. 特征向量化：将测试数据集的特征向量化，使用 DictVectorizer 的 transform 方法。
10. 决策树预测：使用训练好的决策树分类器对测试数据集进行预测，调用决策树分类器的 predict 方法。
11. 计算准确率：使用训练数据集计算决策树的准确率，调用决策树分类器的 score 方法。
12. 输出结果：打印决策树的准确率。

**分类结果**

