哈尔滨工业大学（深圳）

统计机器学习
实验指导书

实验二 构建决策树模型实现银行借贷预测

目录

1. 实验目的

1. 学会理解决策树模型的原理并掌握其构建方法；
2. 学会调用 Python 自编程、调用 Sklearn 库实现决策树模型的定义、训练与预测功能。

2. 实验内容

1. **任务**：对本次实验给出的**银行借贷数据集**，使用 **Python 自编程**构造出一棵决策树，实现借贷与否的预测。实验要求选用一种合适的算法，来构造决策树模型，结合精确率 P、召回率 R 以及 F1 值来评价模型；
2. **附加题**：对本次实验给出的**银行借贷数据集**，使用 **Sklearn 库编程**完成决策树模型预测银行借贷与否。实验要求要有调参过程，要评价模型，绘制出决策树（选做）。

3. 实验环境

1. Windows10
2. PyCharm

4. 实验要点

下面以**课本上第 71 页的样例数据**为例，分别用 Python 自编程、调用 Sklearn 库两种方式构建一个决策树模型，实现申请贷款批准与否的预测。其中数据集如下：

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

4.1 Python 自编程(以 C4.5 算法为例)

4.1.1 导入必要库

```
import numpy as np
import pandas as pd
from math import log
```

4.1.2 导入数据

导入数据，读取 xlsx 文件并查看所有的表头（注意：请根据文件的实际路径获取数据）

```
df = pd.read_excel(r"课本样例数据.xls")
#获取数据集和每个维度的名称
df = df.drop(['ID'], axis=1)    #把第 1 列 ID 数据去除掉
datasets = df.values
labels = df.columns.values
print(datasets)
print(labels)
return datasets, labels
```

4.1.3 定义节点类

```
class Node:
    def __init__(self, root=True, label=None, feature_name=None,
feature=None):
```

```
        self.root = root
        self.label = label
        self.feature_name = feature_name
        self.feature = feature
        self.tree = {}
        self.result = {'label:': self.label, 'feature': self.feature,
'tree': self.tree}

    def __repr__(self):
        return '{}'.format(self.result)

    def add_node(self, val, node):
        self.tree[val] = node

    def predict(self, features):
        if self.root is True:
            return self.label
        return self.tree[features[self.feature]].predict(features)
```

4.1.4 定义二叉树

模型相关的理论知识：

熵： $H(x) = -\sum_{i=1}^{n} p_i \log p_i$

条件熵： $H(X|Y) = \sum P(X|Y) \log P(X|Y)$

信息增益： $g(D, A) = H(D) - H(D|A)$

信息增益比： $g_R(D, A) = \frac{g(D,A)}{H(A)}$

代码相关

```
class DTree:
    def __init__(self, epsilon=0.1):
        self.epsilon = epsilon
        self._tree = {}

    # 熵
    def calc_ent(datasets):
        ######自行定义熵的计算方法#######

    # 经验条件熵
    def cond_ent(self, datasets, axis=0):
```

```
        ######自行定义经验条件熵的计算方法#######

    # 信息增益比
    def info_gain_ratio_train(self, datasets):
        ######自行定义信息增益比的计算方法，并返回最大值#######

    def train(self, train_data):
        """
        input:数据集 D(DataFrame 格式)，特征集 A，阈值 eta
        output:决策树 T
        """
        _, y_train, features = train_data.iloc[:, :-1],
train_data.iloc[:, -1], train_data.columns[:-1]
        epsilon = self.epsilon
          ######请自行编程完成 C4.5 算法构建一颗决策树#######

# 1 若 D 中实例属于同一类 Ck，则 T 为单节点树，并将类 Ck 作为结点的类标记，返回 T
# 2 若 A 为空，则 T 为单节点树，将 D 中实例树最大的类 Ck 作为该节点的类标记，返回 T
# 3 计算最大信息增益比，选择信息增益比最大的特征 Ag
# 4 如果 Ag 的信息增益小于阈值 eta,则置 T 为单节点树，并将 D 中是实例数最大的类 Ck 作为
该节点的类标记，返回 T
# 5 否则，对 Ag 的每一个可能的值 ai，依照 Ag=ai 将 D 分割为若干个非空子集 Di，将 Di 中实
例树最大的类作为标记，构建子节点，由结点及其子节点构成树 T，返回 T
# 6 对第 i 个子节点，以 Di 为训练集，以 A-|Ag|为新的特征集，递归调用 1~5，得到树 T，返
回 T

    def fit(self, train_data):
        self._tree = self.train(train_data)
        return self._tree

    def predict(self, X_test):
        return self._tree.predict(X_test)
```

4.1.5 调用生成树

```
datasets, labels = create_data()
train_data = pd.DataFrame(datasets, columns=labels)
print(train_data)
dt = DTree()
tree = dt.fit(train_data)
print(tree)
```

4.1.6 预测模型

预测当输入为［'老年'，'否'，'否'，'一般'］时，银行是否借贷。（注意：本次实验内容中要同感知机实验任务一样，自行分割出测试集做预测，不能自己编造测试集）

```
print(dt.predict(['老年', '否', '否', '一般']))
```

4.1.7 补充知识

在二分类任务中，各指标的计算基础都来自于对正负样本的分类结果，用混淆矩阵表示为：

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

（1）**精确率**：分类正确的正样本个数占分类器判定为正样本的样本个数的比例。
分类正确的正样本个数：即真正例(TP)。
分类器判定为正样本的个数：包括真正例(TP)和假正例(FP)

$$P = \frac{TP}{TP + FP}$$

召回率：分类正确的正样本个数占真正的正样本个数的比例。
分类正确的正样本个数：即真正例(TP)。
真正的正样本个数：包括真正例(TP)和假负例(FN)

$$R = \frac{TP}{TP + FN}$$

F1 分数：精确率和召回率的调和均值。

$$F1 = \frac{2TP}{2TP + FP + FN}$$

注意：这 3 个评价指标需自行定义，并用在实验任务中。

4.2 使用 Sklearn 库编程（以 CART 算法为例）

4.2.1 安装依赖库

（1）pandas 依赖处理 Excel 的 xlrd 模块，安装命令是：pip install xlrd
（2）决策树分类器依赖 sklearn 模块，安装命令是：pip install scikit-learn

（3）决策树分类器依赖 pydotplus 模块和 Graphviz：
（4）安装 pydotplus，安装命令是：pip install pydotplus
（5）安装 Graphviz，下载 graphviz-install-2.44.1-win64.exe 并安装，下载地址：https://www2.graphviz.org/Packages/stable/windows/10/cmake/Release/x64/
按照指令安装，并添加安装路径至 PATH 环境变量

4.2.2 导入数据

引入 pandas 模块，导入数据，读取 xlsx 文件并查看所有的表头（注意：请根据文件的实际路径获取数据）

```
import pandas as pd
#读取 xlsx 文件并查看所有的表头
df = pd.read_excel(r"../课本样例数据.xls")
cols=df.columns.values
print(cols)
```

4.2.3 转换文字数据集为数字数据集

将所有分类映射为 0/1 分类 或者 1/2/3 分类，获取所有的数据(除去第一列的 ID 信息，以及第一行的表头信息)，并转为二维数组的形式

```
age = {'青年': 1, '中年': 2, '老年': 3}
job = {'是': 1, '否': 0}
housing = {'是': 1, '否': 0}
credit = {'一般': 1, '好': 2, '非常好': 3}
loan = {'是': 1, '否': 0}
df['年龄'] = df['年龄'].map(age)
df['有工作'] = df['有工作'].map(job)
df['有自己的房子'] = df['有自己的房子'].map(housing)
df['信贷情况'] = df['信贷情况'].map(credit)
df['类别'] = df['类别'].map(loan)
df = df.drop(['ID'], axis=1)
datas=df.values
print(datas)
```

4.2.4 划分训练集和测试集

注意：因课本上例子的数据集太少，所以没有做训练集和测试集的分类，而**实验作业中是要数据集分类的，可参考实验一感知机中的内容。**

```
feature =['age','no job','no housing','credit']
classname =['no loan','loan']


#划分数据集
```

```
X = [x[0:4] for x in datas]
#print(X)
Y = [y[-1] for y in datas]
#print(Y)
```

4.2.5 构建决策树和训练模型

引入 sklearn 模型，构建决策树，训练模型

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz

tree_clf = DecisionTreeClassifier(max_depth=4)
tree_clf.fit(X, Y)
```

4.2.6 预测模型

与前面 python 编程一样，预测当输入为 ['老年','否','否','一般']时，银行是否借贷，注意需要转化为数值类型。（注意：本次实验内容中要自行分割出测试集做预测，不能自己编造测试集）

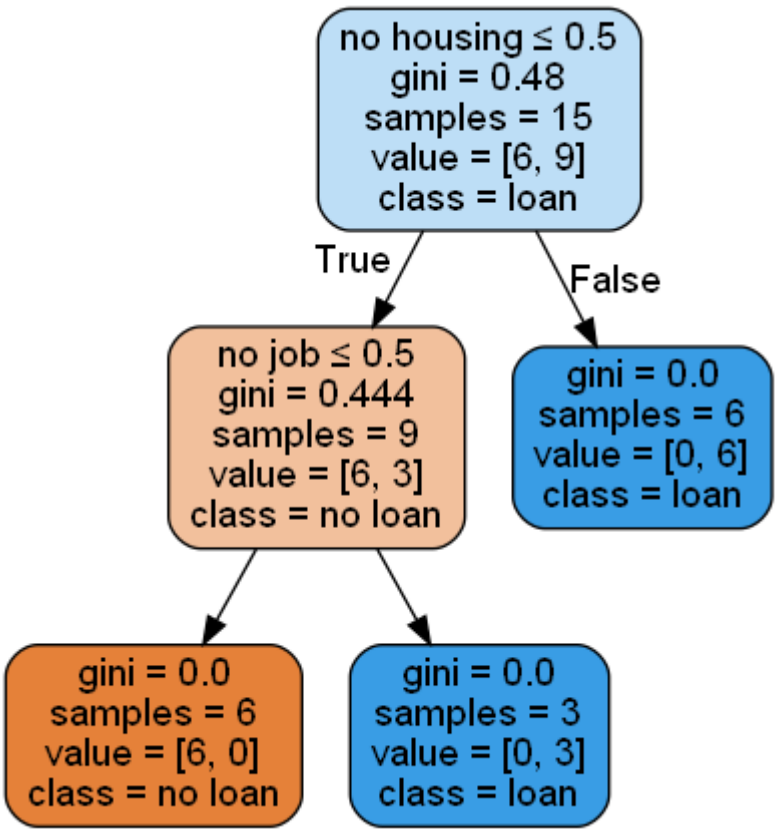
```
print(tree_clf.predict([[3, 0, 0, 1]]))
```

4.2.7 可视化决策树模型

引入 pydotplus 模块，可视化决策树模型（注意：请根据文件的实际路径保存图片）

```
import pydotplus
dot_data=export_graphviz(
          tree_clf,
          out_file=None,
          feature_names=feature,
          class_names=classname,
          rounded=True,
          filled=True,
   special_characters=True)
graph=pydotplus.graph_from_dot_data(dot_data)
graph.write_png("../loan.png")
```

运行结果如下：



可能会遇到这样的问题，如下：

报错信息：

pydotplus.graphviz.InvocationException: Program terminated with status: 1. stderr follows: Format: "png" not recognized. Use one of:

请参考网上的解决方法：https://www.iteye.com/blog/peter1981-2517186

4.2.8 补充知识

1）如何调参

构建模型中很重要的一步是调参。在 sklearn 中，模型的参数是通过方法参数来决定的，以下给出 sklearn 中，决策树的参数：

```
DecisionTreeClassifier(criterion="gini",
                       splitter="best",
                       max_depth=None,
                       min_samples_split=2,
                       min_samples_leaf=1,
                       min_weight_fraction_leaf=0.,
                       max_features=None,
                       random_state=None,
                       max_leaf_nodes=None,
                       min_impurity_decrease=0.,
                       min_impurity_split=None,
                       class_weight=None,
                       presort=False)
```

通常来说，**较为重要的参数有**：

criterion：用以设置用信息熵还是基尼系数计算

string, optional (default="gini")

(1).criterion='gini',分裂节点时评价准则是 Gini 指数。

(2).criterion='entropy',分裂节点时的评价指标是信息增益。

splitter：指定分支模式

string, optional (default="best")。指定分裂节点时的策略。

(1).splitter='best',表示选择最优的分裂策略。

(2).splitter='random',表示选择最好的随机切分策略。

max_depth：最大深度，防止过拟合

int or None, optional (default=None)。指定树的最大深度。

如果为 None，表示树的深度不限。直到所有的叶子节点都是纯净的，即叶子节点中所有的样本点都属于同一个类别。或者每个叶子节点包含的样本数小于 min_samples_split。

min_samples_leaf：限定每个节点分枝后子节点至少有多少个数据，否则就不分枝

int, float, optional (default=2)。表示分裂一个内部节点需要的做少样本数。

(1).如果为整数，则 min_samples_split 就是最少样本数。

(2).如果为浮点数(0 到 1 之间)，则每次分裂最少样本数为 ceil(min_samples_split * n_samples)

2）如何定义评价指标函数

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()  #决策树分类器
clf.fit(X_train, y_train)  #拟合数据
y_pred = clf.predict(X_test)  #得出预测结果,从测试数据集

from sklearn import metrics
p = metrics.precision_score(y_test, y_pred)  #计算精度 P
r = metrics.recall_score(y_test, y_pred)  #计算召回率 R
f1 = metrics.f1_score(y_test, y_pred)  #计算 F1
```