



大数据导论实验



实验二 数据理解、数据预处理及决策树的应用

主讲教师：叶允明

实验教师：谢佳、房敏

目录

◆ 实验二任务

◆ 数据说明

◆ 预备知识

◆ 实验步骤

本学期实验总体安排

实验课程共**4**个学时，**2**个实验项目，总成绩为**30**分。

实验一 (15分)

Hadoop环境配置与 基本操作

掌握Hadoop分布式环境的配置方法；理解Mapreduce作业的原理和操作方法。

实验二 (15分)

数据理解、数据预处理及决策树的应用

通过应用案例实践数据预处理方法；
编码实现一个经典的数据挖掘算法。

实验二任务

实验任务：提供银行精准营销解决方案

该数据集与葡萄牙银行机构的直接营销活动有关。这些营销活动是以电话为基础的。**营销活动的目的是希望客户能够认购自己的产品。**通常来说，银行机构的客服人员至少需要联系一次客户来得知客户是否将认购银行的产品（定期存款）。**案例数据是随机挑选的部分银行客户基本资料数据和与客户联系的相关数据**，包括年龄、婚姻状况、受教育程度、联系持续时间、联系次数等。



数据集	银行营销
记录数量	4521
属性	16
类别	2

因此，与该数据集对应的任务是**分类任务**，而**分类目标是预测客户是 (yes) 否 (no) 认购定期存款（变量y）**。

数据说明

客户基本资料的数据:

属性	描述	类型	缺失值
age	年龄	数值	无
job	工作类型	分类	unknown
marital	婚姻状况	分类	无
education	受教育程度	分类	unknown
default	是否存在拖欠	分类	无
balance	平均年度余额（欧元）	数值	无
housing	是否有住房贷款	分类	无
loan	是否有个人贷款	分类	无

数据说明

与客户联络的相关数据:

属性	描述	类型	缺失值
contact	联络方式	分类	unknown
day	最后一次联系是几号	数值	无
month	最后一次联系的月份	分类	无
duration	最后一次联系的持续时间（秒）	数值	无
campaign	此营销活动期间和此客户联系的次数，包括最后一次联系	数值	无
pdays	从上一个营销活动最后一次联系客户后经过的天数，-1表示之前未联系过客户	数值	无
previous	此营销活动之前和此客户联系的次数	数值	无
poutcome	上一次营销活动的结果	分类	unknown
y (目标)	客户是否将订购银行定期存款	分类	无

预备知识

Pandas库的介绍

Pandas是python第三方库，它是一个强大的分析结构化数据的工具集；它的使用基础是Numpy（提供高性能的矩阵运算），用于数据挖掘和数据分析，同时也提供数据清洗功能。

```
pip install pandas
```

工欲善其事必先利其器！

预备知识

Pandas的核心数据结构

维数	名称	描述
1	Series	带标签的一维同构数组
2	DataFrame	带标签的，大小可变的，二维异构表格

- **Series** 是带标签的一维数组，可存储整数、浮点数、字符串、Python 对象等类型的数据。
- **DataFrame** 是一个表格型的数据结构，类似于 Excel 或 sql 表，它含有一组有序的列，每列可以是不同的值类型（数值、字符串、布尔值等）。

预备知识

Pandas库的基本操作

- 查看数据
`df.describe()`、`df.index`、`df.columns`、`df.values`
- 选取特定列和行的数据
`df[col]`、`df[[col1,col2]]`、`df.iloc[[row1, row2], [col1, col2]]`
- 增加、删除、修改列的值
`del df[col]`、`df.pop`
- 导入导出文件
`pd.read_csv`、`df.to_csv`

更多学习请参考pandas中文网: <https://www.py pandas.cn/>

实验步骤



数据读取

① 读取CSV文件，获取所有数据

```
import numpy as np
import pandas as pd
df = pd.read_csv("D:/大数据/实验二/bank.csv")
cols = df.columns.values
cols = [col.replace('\\', '') for col in cols[0].split(';')]
print(cols)
```

```
['age', 'job', 'marital', 'education', 'default', 'balance', 'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays', 'previous', 'poutcome', 'y']
```

```
datas = []
for i in range(len(df)):
    s = df.iloc[i].values[0]
    datas.append([item.replace('\\', '') for item in s.split(';')])
datas = np.array(datas)
print(datas)
```

```
[['30' 'unemployed' 'married' ... '0' 'unknown' 'no']
['33' 'services' 'married' ... '4' 'failure' 'no']
['35' 'management' 'single' ... '1' 'failure' 'no']
...
['57' 'technician' 'married' ... '0' 'unknown' 'no']
['28' 'blue-collar' 'married' ... '3' 'other' 'no']
['44' 'entrepreneur' 'single' ... '7' 'other' 'no']]
```

实验步骤



数据读取

```
df_frame = {}  
for i in range(len(cols)):  
    df_frame[cols[i]] = datas[:,i]  
df = pd.DataFrame(df_frame)  
df
```

② 转化为DataFrame格式

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no
5	35	management	single	tertiary	no	747	no	no	cellular	23	feb	141	2	176	3	failure	no
6	36	self-employed	married	tertiary	no	307	yes	no	cellular	14	may	341	1	330	2	other	no
7	39	technician	married	secondary	no	147	yes	no	cellular	6	may	151	2	-1	0	unknown	no
8	41	entrepreneur	married	tertiary	no	221	yes	no	unknown	14	may	57	2	-1	0	unknown	no
9	43	services	married	primary	no	-88	yes	yes	cellular	17	apr	313	1	147	2	failure	no
10	39	services	married	secondary	no	9374	yes	no	unknown	20	may	273	1	-1	0	unknown	no
11	43	admin.	married	secondary	no	264	yes	no	cellular	17	apr	113	2	-1	0	unknown	no
12	36	technician	married	tertiary	no	1109	no	no	cellular	13	aug	328	2	-1	0	unknown	no
13	20	student	single	secondary	no	502	no	no	cellular	30	apr	261	1	-1	0	unknown	yes
14	31	blue-collar	married	secondary	no	360	yes	yes	cellular	29	jan	89	1	241	1	failure	no
15	40	management	married	tertiary	no	194	no	yes	cellular	29	aug	189	2	-1	0	unknown	no

实验步骤

1

数据读取

③ 类型转换

将age、balance、duration、campaign、pdays和previous六个字段从字符类型转为整数类型。

```
df[['age','balance','duration','campaign','pdays','previous']] = df[['age','balance','duration','campaign','pdays','previous']].astype(int)
```

```
df.dtypes
```

age	int32
job	object
marital	object
education	object
default	object
balance	int32
housing	object
loan	object
contact	object
day	object
month	object
duration	int32
campaign	int32
pdays	int32
previous	int32
poutcome	object
y	object

Length: 17, dtype: object

调用`astype`后返回的是一个新的对象，因此需要对原来的数据列进行赋值。之后再调用`dtypes`参数即可查看属性类型。

实验步骤

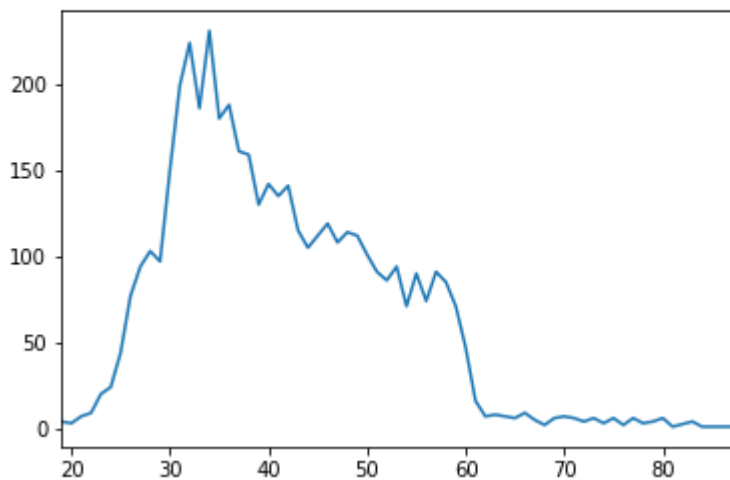


数据读取

④ 数据可视化

```
df['age'].value_counts().sort_index().plot.line()
```

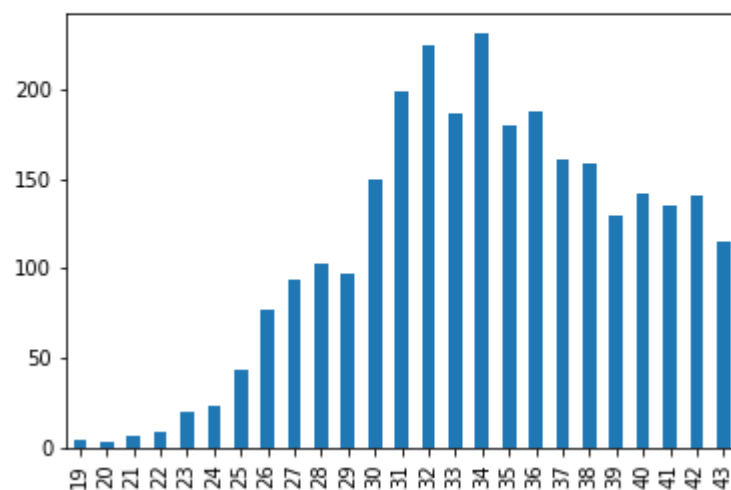
<matplotlib.axes._subplots.AxesSubplot at 0x247287b8be0>



折线图

```
df['age'].value_counts().sort_index().head(25).plot.bar()
```

<matplotlib.axes._subplots.AxesSubplot at 0x24729ad5898>



柱状图

实验步骤

2

数据预处理

① 筛选重复值并删除

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
4516   False
4517   False
4518   False
4519   False
4520   False
Length: 4521, dtype: bool
```

```
df.drop_duplicates()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no
...
4516	33	services	married	secondary	no	-333	yes	no	cellular	30	jul	329	5	-1	0	unknown	no
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	9	may	153	1	-1	0	unknown	no
4518	57	technician	married	secondary	no	295	no	no	cellular	19	aug	151	11	-1	0	unknown	no
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	6	feb	129	4	211	3	other	no
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	3	apr	345	2	249	7	other	no

② 将yes/no分类映射为0/1分类

```
class_mapping = {'no':0, 'yes':1}
df['default'] = df['default'].map(class_mapping)
df['housing'] = df['housing'].map(class_mapping)
df['loan'] = df['loan'].map(class_mapping)
df['y'] = df['y'].map(class_mapping)
```

实验步骤

2

数据预处理

③ 将月份和日期转化成距离现在的天数，并删除原有的月份与日期

```
month_dict = {'oct': '10', 'may': '05', 'apr': '04', 'jun': '06', 'feb': '02', 'aug': '08',  
              'jan': '01', 'jul': '07', 'nov': '11', 'sep': '09', 'mar': '03', 'dec': '12'}  
df['month'] = df['month'].map(month_dict)  
df['date'] = '2019' + '-' + df['month'] + '-' + df['day']  
df['date'] = pd.to_datetime(df['date'], format="%Y-%m-%d")  
df['date'] = pd.to_datetime('2020-01-01', format="%Y-%m-%d") - df['date']  
df['date'] = df['date'].dt.days  
print(df[['date']])
```

	date
0	74
1	235
2	260
3	212
4	241
...	...
4516	155
4517	237
4518	135
4519	329
4520	273

```
del(df['day'])  
del(df['month'])
```

实验步骤

2

数据预处理

④ 将字符型分类变量转化为数值型并处理缺失值

```
jobs = df['job'].unique()
job_mapping = {jobs[i]: i for i in range(jobs.shape[0])}

maritals = df['marital'].unique()
marital_mapping = {maritals[i]: i for i in range(maritals.shape[0])}

educations = df['education'].unique()
education_mapping = {educations[i]: i for i in range(educations.shape[0])}

contacts = df['contact'].unique()
contact_mapping = {contacts[i]: i for i in range(contacts.shape[0])}

poutcomes = df['poutcome'].unique()
poutcome_mapping = {poutcomes[i]: i for i in range(poutcomes.shape[0])}

df['marital'] = df['marital'].map(marital_mapping)
df['job'] = df['job'].map(job_mapping)
df['education'] = df['education'].map(education_mapping)
df['contact'] = df['contact'].map(contact_mapping)
df['poutcome'] = df['poutcome'].map(poutcome_mapping)
df
```

	age	job	marital	education	default	balance	housing	loan	contact	\
0	30	0	0	0	0	1787	0	0	0	
1	33	1	0	1	0	4789	1	1	0	
2	35	2	1	2	0	1350	1	0	0	
3	30	2	0	2	0	1476	1	1	1	
4	59	3	0	1	0	0	1	0	1	
...	
4516	33	1	0	1	0	-333	1	0	0	
4517	57	4	0	2	1	-3313	1	1	1	
4518	57	5	0	1	0	295	0	0	0	
4519	28	3	0	1	0	1137	0	0	0	
4520	44	6	1	2	0	1136	1	1	0	
	duration	campaign	pdays	previous	poutcome	y	date			
0	79	1	-1	0	0	0	74			
1	220	1	339	4	1	0	235			
2	185	1	330	1	1	0	260			
3	199	4	-1	0	0	0	212			
4	226	1	-1	0	0	0	241			
...			
4516	329	5	-1	0	0	0	155			
4517	153	1	-1	0	0	0	237			
4518	151	11	-1	0	0	0	135			
4519	129	4	211	3	2	0	329			
4520	345	2	249	7	2	0	273			

[4521 rows x 16 columns]

实验步骤

2

数据预处理

⑤ 将连续属性离散化

➤ 分箱法

把数据按特定的规则进行分组，实现数据的离散化，增强数据稳定性，减少过拟合风险。

➤ 二分法

对某个属性出现的连续值从小到大排列，取每两个点的中点进行划分，选取其中信息增益最大的点作为最佳分裂点。

```
bins = [18, 25, 35, 45, 55, 100] # 指定年龄的分界点
df['age'] = pd.cut(df['age'], bins, labels=False)
bins = [-np.inf, 4137.1, 11587.2, np.inf]
df['balance'] = pd.cut(df['balance'], bins, labels=False)
print(df[['age', 'balance']])
```

	age	balance
0	1	0
1	1	1
2	1	0
3	1	0
4	4	0
...
4516	1	0
4517	4	0
4518	4	0
4519	1	0
4520	2	0

分箱

⑥ 保存数据

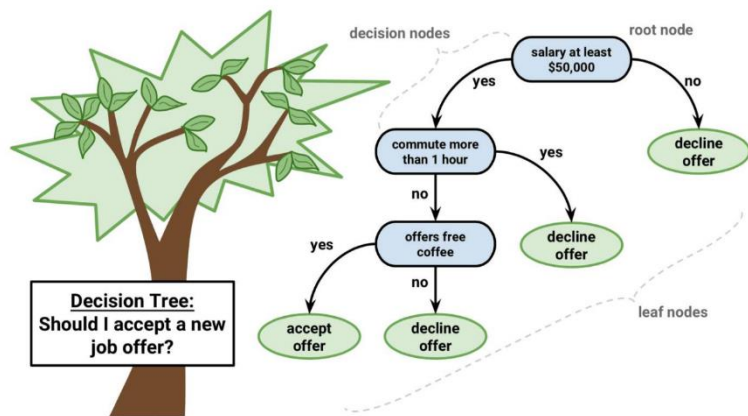
```
df.to_csv('D:/大数据/实验二/after_bank.csv')
```

实验步骤

3

构建决策树

决策树是一种树形结构的分类器，树内部的每一个节点代表的是对一个特征的测试，树的分支代表该特征的每一个测试结果，而树的每一个叶子节点代表一个类别。通常根据特征的信息增益或其他指标，构建一棵决策树。



实验步骤



构建决策树

一棵决策树的生成过程主要分为以下3个部分:

(1) **特征选择**: 特征选择是指从训练数据中众多的特征中选择一个特征作为当前节点的分裂标准, 如何选择特征有着很多不同量化评估标准 (**信息增益、信息增益率、基尼系数**等), 从而衍生出不同的决策树算法。

(2) **决策树生成**: 根据选择的特征评估标准, 从上至下**递归**地生成子节点, 直到数据集不可分则停止决策树停止生长。

(3) **剪枝**: 决策树容易过拟合, 一般来需要**剪枝**, 缩小树结构规模、缓解过拟合。剪枝技术有预剪枝和后剪枝两种。

实验步骤



构建决策树

实验任务：

请**编码实现某种决策树算法**，解决银行精准营销的分类问题。

实验要求：

- ① 编程语言不限
- ② 决策树算法不限，可选择ID3、C4.5、C5.0、CART等。
- ③ **不允许调用sklearn等现成库**
- ④ 将预处理后的数据自行划分成训练集和测试集，采用交叉验证可作为加分项。
- ⑤ 输入训练数据和相关参数（如树的最大深度等）构建决策树。
- ⑥ 输入测试数据，使用训练好的决策树进行预测，输出预测结果和准确率。

实验步骤



构建决策树

提交要求:

- ① **实验报告**内容包括预处理的过程及结果、构建决策树的基本过程、测试结果和预测准确率。决策树构建过程需描述训练集和测试集的划分方法，特征选择的方法，剪枝的方法等。
- ② 复现实验结果所需文件，包含**决策树代码**和相应的**数据文件**。
- ③ 测试集预测**结果文件**



大数据导论实验



同学们，请开始实验吧！