

利用python进行数据分析：

□ 分析步骤

- 定义数据分析目标：明确挖掘数据的目标和达到的效果。
- 数据取样：采集目标相关样本数据子集，确保数据的相关性、可靠性、有效性。
- 数据探索：对样本数据探索、审核、加工处理，保证样本数据的质量。
- 数据预处理：改善数据质量，包括数据筛选、数据变量转换、缺失值数据处理等。
- 挖掘建模：确定分析问题类型（分类，聚类、关联等），选择相应算法构建模型。
- 模型评价：从建立模型中找到一个最好的模型，并应用到实际业务中。



数据探索：

- 数据质量分析：主要任务是检查原始数据中是否存在脏数据，即不符合要求，不能直接处理的数据，包括缺失值分析、异常值分析、一致性分析。
- 数据特征分析
 - 分布分析：揭示数据的分布特征和分布类型，通过绘制频率分布表、茎叶图等直观分析
 - 统计量分析：用统计量指标对定量数据进行统计描述，常从集中趋势和离中趋势两个方面进行分析。
 - 相关性分析：分析连续变量之间线性相关程度的强弱，并用适当的统计指标表示出来。



数据预处理：

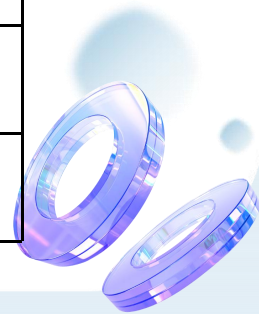
- ❑ 数据清洗：删除原始数据集中的无关数据、重复数据、平滑噪声数据、无关数据，处理缺失值和异常值。
- ❑ 数据集成：将多个数据源合并存放在一个一致的数据存储（如数据仓库）中的过程。
- ❑ 数据变换：主要是对数据进行规范化处理，将数据转换成适当的形式，以适用于挖掘任务和算法的需要。
- ❑ 数据规约：产生更小但保持数据完整性的新数据集，在规约后的数据集上进行分析和挖掘更有效率。



数据预处理：

□ Python主要数据预处理函数

函数名	函数功能	所属扩展库
interpolate	一维、高维数据插值	Scipy
unique	去除数据中重复元素，得到单值元素列表	Pandas/Numpy
isnull	判断是否是空值	Pandas
notnull	判断是否非空值	Pandas
PCA	对指标变量矩阵进行主成分分析	Scikit-Learn
random	生成随机矩阵	Numpy



挖掘建模：

□ 分类与预测

- 分类：构造一个分类模型，输入样本的属性值，输出对应的类别，将每个样本映射到预先定义好的类别
- 预测：建立两种或两种以上变量间相互依赖的函数模型，然后进行预测和控制
- 实现过程：学习步，通过归纳分析训练样本集来建立分类模型得到分类规则；分类步，先用一直的测试样本集评估分类规则的准确率，如果准确率是可以接受的，则使用该模型对未知类标号的待测样本集进行预测



挖掘建模:

□ 主要回归模型分类

回归模型名称	试用条件	算法描述
线性回归	因变量与自变量是线性关系	对一个或多个自变量和因变量之间的线性关系进行建模 可用最小二乘法求解模型系数
非线性回归	因变量与自变量之间不都是线性关系	对一个或多个自变量和因变量之间的非线性关系进行建模。 如果非线性关系可以通过简单的函数变换转化成线性关系，用线性回归的思想求解；如果不能转化，用非线性最小二乘法方法求解
Logistic	因变量一般有1和0两种取值	是广义线性回归模型的特例，利用Logistic函数将因变量的取值范围控制在0和1之间，表示取值为1的概率
主成分回归	参与建模的自变量之间具有多重共线性	主成分回归是根据主成分分析的思想提出来，是对最小二乘法的一种改进，它是参数估计的一种有偏估计。可以消除自变量之间的多重共线性

应用举例一： 对某银行在降低贷款拖欠率的数据进行逻辑回归建模，数据示例如下表

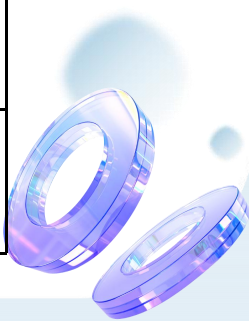
	A	B	C	D	E	F	G	H	I	
1	年龄	教育	工龄	地址	收入	负债率	信用卡负债	其他负债	违约	
2	41	3	17	12	176.00	9.30	11.36	5.01	1	
3	27	1	10	6	31.00	17.30	1.36	4.00	0	
4	40	1	15	14	55.00	5.50	0.86	2.17	0	
5	41	1	15	14	120.00	2.90	2.66	0.82	0	
6	24	2	2	0	28.00	17.30	1.79	3.06	1	
7	41	2	5	5	25.00	10.20	0.39	2.16	0	
8	39	1	20	9	67.00	30.60	3.83	16.67	0	
9	43	1	12	11	38.00	3.60	0.13	1.24	0	
10	24	1	3	4	19.00	24.40	1.36	3.28	1	
11	36	1	0	13	25.00	19.70	2.78	2.15	0	
12	27	1	0	1	16.00	1.70	0.18	0.09	0	
13	25	1	4	0	23.00	5.20	0.25	0.94	0	
14	52	1	24	14	64.00	10.00	3.93	2.47	0	
15	37	1	6	9	29.00	16.30	1.72	3.01	0	
16	48	1	22	15	100.00	9.10	3.70	5.40	0	
17	36	2	9	6	49.00	8.60	0.82	3.40	1	
18	36	2	13	6	41.00	16.40	2.92	3.81	1	



挖掘建模:

□ 常用的分类与预测算法

算法分析	算法描述
回归分析	回归分析是确定去测属性（数值型）与其他变量间相互依赖的定量关系最常用的统计学方法。包括线性回归、非线性回归、Logistic回归、岭回归、主成分回归、偏最小二乘回归等模型
决策树	决策树采用自顶向下的递归方式，在内部节点进行属性值的比较，并根据不同的属性值从该节点向下分支，最终得到的叶节点是学习划分的类
人工神经网络	人工神经网络是一种模仿大脑神经网络和功能而建立的信息处理系统，表示神经网络的输入与输出变量之间关系的模型
贝叶斯网络	贝叶斯网络又称信度网络，是Bayes方法的扩展，是目前不确定知识表达和推理领域最有效的理论模型之一
支持向量机	支持向量机是一种通过魔种非线性映射，把低纬的非线性可分转化为高维的线性可分，在高维空间进行线性分析的算法

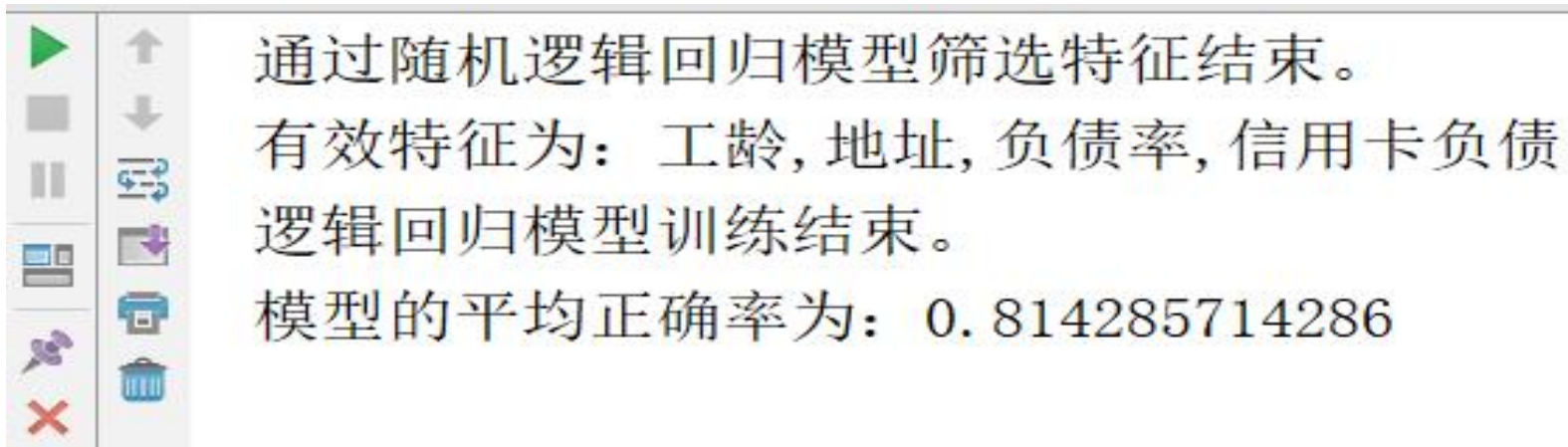


Python代码

```
1  -*- coding: utf-8 -*-
2  #逻辑回归 自动建模
3  import pandas as pd
4
5  #参数初始化
6  filename = 'D://data/bankloan.xls'
7  data = pd.read_excel(filename)
8  x = data.iloc[:, :8].as_matrix()
9  y = data.iloc[:, 8].as_matrix()
10
11  from sklearn.linear_model import LogisticRegression as LR
12  from sklearn.linear_model import RandomizedLogisticRegression as RLR
13  rlr = RLR() #建立随机逻辑回归模型, 筛选变量
14  rlr.fit(x, y) #训练模型
15  rlr.get_support() #获取特征筛选结果, 也可以通过.scores_方法获取各个特征的分
16  print(u'通过随机逻辑回归模型筛选特征结束。')
17  print(u'有效特征为: %s' % ','.join(data.columns[rlr.get_support()]))
18  x = data[data.columns[rlr.get_support()]].as_matrix() #筛选好特征
19  lr = LR() #建立逻辑回归模型
20  lr.fit(x, y) #用筛选后的特征数据来训练模型
21  print(u'逻辑回归模型训练结束。')
22  print(u'模型的平均正确率为: %s' % lr.score(x, y)) #给出模型的平均正确率, 本例为81.4%
```



运行结果



通过随机逻辑回归模型筛选特征结束。
有效特征为：工龄, 地址, 负债率, 信用卡负债
逻辑回归模型训练结束。
模型的平均正确率为：0.814285714286

结果分析：随机逻辑回归剔除变量，分别剔除了x2、x8、x1、x5，最终构建模型包含的变量为常量x3、x4、x6、x7。在建立逻辑回归模型时，使用了默认的阈值0.25。



聚类分析：在没有给定划分类别的情况下，根据数据相似度进行样本分组的一种方法。常用

聚类方法

类别	包括的主要算法
划分方法	K-Means算法、K-MEDOIDS算法、CLARANS算法
层次分析法	BIRCH算法、CURE算法、CHAMELEON算法
基于密度的方法	DBCSCAN算法、DENCLUE算法、OPTICS算法
基于网格的方法	STING算法、CLIOUE算法、WAVE——CLUSTER算法
基于模型的方法	统计学方法、神经网络方法



常用聚类分析算法

算法名称	算法描述
K-Means	K-均值聚类也称为快速聚类法，在最小化误差函数的基础上将数据划分为预定的类数K。该算法原理简单并便于处理大量数据
K-中心点	K-均值算法对孤立点的敏感性，K-中心点算法不采用簇中对象的平均值作为簇中心，而选用簇中离平均值最近的对象作为簇中心
系统聚类	系统聚类也称为多层次聚类，分类的单位由高到低呈树形结构，且所处的位置越低，其包含的对象就越少，但这些对象间的共同特征越多。该聚类方法只适用在小数据量的时候使用，数据量大的时候速度会非常慢



K-Means聚类算法

□ 算法过程

- 从N个样本数据中随机选取K个对象作为初始的聚类中心
- 分别计算每个样本到各个聚类中心的距离，将对象分配到距离最近的聚类中
- 所有对象分配完成后，重新计算K个聚类的中心
- 与前一次计算得到的K个聚类中心比较，如果聚类中心发生变化，转第二步，否则转下一步
- 当质心不发生变化时停止并输出聚类结果

