# 用银行营销数据学习数据挖掘：探索预测客户购买行为的模型

## 0 了解数据

来源： UCI Machine Learning Repository

（<https://archive.ics.uci.edu/ml/datasets/bank+marketing>）

数据集信息：

这份数据与葡萄牙银行机构的直接营销活动有关。这些营销活动基于电话呼叫。通常需要多次联系同一客户，以确定是否会订阅产品（银行定期存款）。

属性信息：

输入变量：

1-age：年龄（数值型）

2-job：职业类型（分类型）

'admin.'：行政人员

'blue-collar'：蓝领工人

'entrepreneur'：企业家

'housemaid'：家庭主妇

'management'：管理人员

'retired'：退休人员

'self-employed'：自雇人士

'services'：服务行业人员

'student'：学生

'technician'：技术人员

'unemployed'：失业人员

'unknown'：未知职业

3-marital：婚姻状况（分类型）

'divorced'：离异或丧偶

'married'：已婚

'single'：单身

'unknown'：未知婚姻状态

4-education：教育程度（分类型）

'basic.4y'：4年制基础教育

'basic.6y'：6年制基础教育

'basic.9y'：9年制基础教育

'high.school'：高中教育

'illiterate'：文盲

'professional.course'：职业课程教育

'university.degree'：大学本科以上教育

'unknown'：未知教育程度

5-default：默认信用：是否有信用违约？（分类型：'no'否，'yes'是，'unknown'未知）

6-housing：住房贷款：是否有住房贷款？（分类型：'no'否，'yes'是，'unknown'未知）

7-loan：个人贷款：是否有个人贷款？（分类型：'no'否，'yes'是，'unknown'未知）

8-contact：联系方式：联系沟通类型（分类型：'cellular'：手机，'telephone'：电话）

9- month：月份：年度最后一次联系月份（分类型）

10- day\_of\_week：星期几：最后一次联系星期几（分类型）

11-duration：持续时间：最后一次联系持续时间，以秒为单位（数值型）。重要说明：此属性对输出目标影响很大（例如，如果持续时间为0，则y='no'）。然而，在进行通话之前并不知道持续时间。而且，在通话结束后，y显然是已知的。因此，此输入仅应包括在基准目的中，并且如果意图是拥有真实的预测模型，则应将其丢弃。

12-campaign：营销活动：在此次营销活动和针对此客户期间执行的联系次数（数值型，包括最后一次联系）

13-pdays：客户上次从以前的营销活动中联系后经过的天数（数值型；999表示先前没有联系过客户）

14-previous：以前的联系：在此次营销活动和针对此客户之前执行的联系次数（数值型）

15-poutcome：以前营销活动的结果（分类型）

'failure'：营销活动失败

'nonexistent'：之前未进行过营销活动

'success'：营销活动成功

16-emp.var.rate：就业变化率 - 季度指标（数值型）

17-cons.price.idx：消费者物价指数 - 月度指标（数值型）

18-cons.conf.idx：消费者信心指数 - 月度指标（数值型）

19-euribor3m：欧元区3个月间隔利率 - 日度指标（数值型）

20-nr.employed：雇员数量 - 季度指标（数值型）

输出变量：

y：客户是否已经订阅了定期存款？（二元：'yes'是，'no否'）

## 1 导入数据

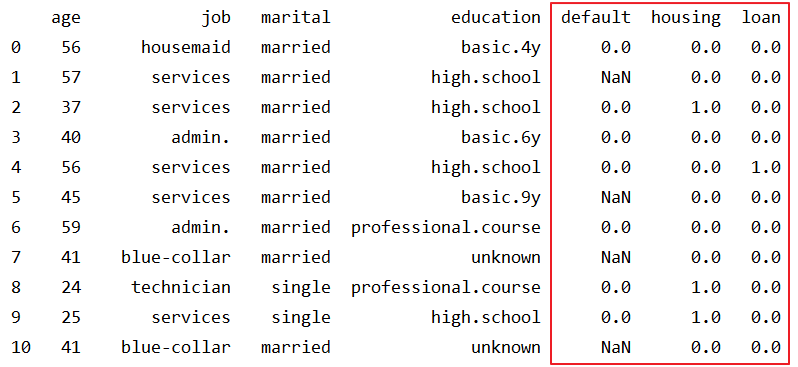
使用pandas库中的read\_csv()方法读取名为'bank.csv'的数据文件，并将其保存在名为'df'的DataFrame对象中。参数'encoding'指定了数据文件的编码方式为'utf-8-sig'，参数'sep'指定了数据文件中的字段分隔符为';'。

## 2 数据预处理

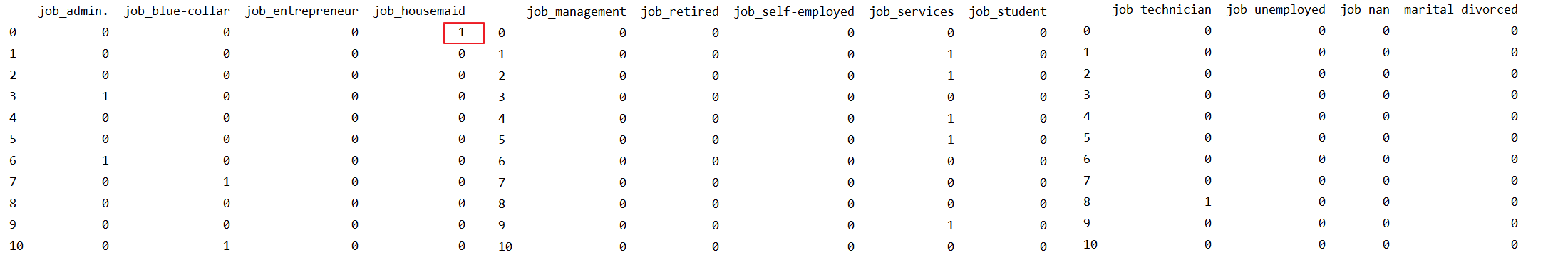
### 2.1 分类数据数值化

此处的分类数据分为两种情况，一是只含有两种取值（除了 'unknown' 缺失值以外有两种取值），二是含有多种取值的分类数据，如教育程度（education）。对于第一种情况，我们可以直接使用 0 和 1 来代替其两种取值。对于第二种情况，我们可以使用哑编码或者独热编码将分类数据转换为数值数据，此处，我们使用哑编码。

处理后数据（部分）如图，我们可以看出对于default而言，unknown转换为了NaN，其他取值使用0和1替换。



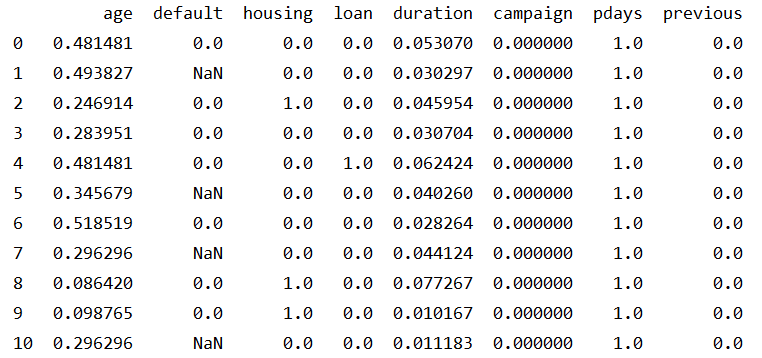
对于job而言，我们进行了哑编码之后的结果如下图，我们可以看出第一条数据为000100000000，表示housemaid。



### 2.2 数值型数据规范化

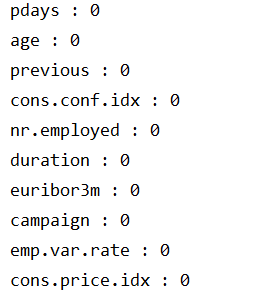
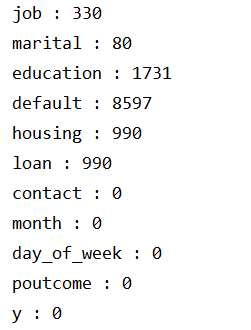
数值型数据的规范化是指将数据缩放到指定的范围内，通常是将数据缩放到[0,1]或[-1,1]的区间内。进行规范化的原因是不同变量具有不同的量纲，这会影响机器学习算法的性能，导致一些变量对结果的贡献更大，而另一些变量对结果的贡献更小。Python中可以使用MinMaxScaler来进行规范化。

处理后数据如图，我们可以看出通过规范化可以将数值投射到[0,1]。

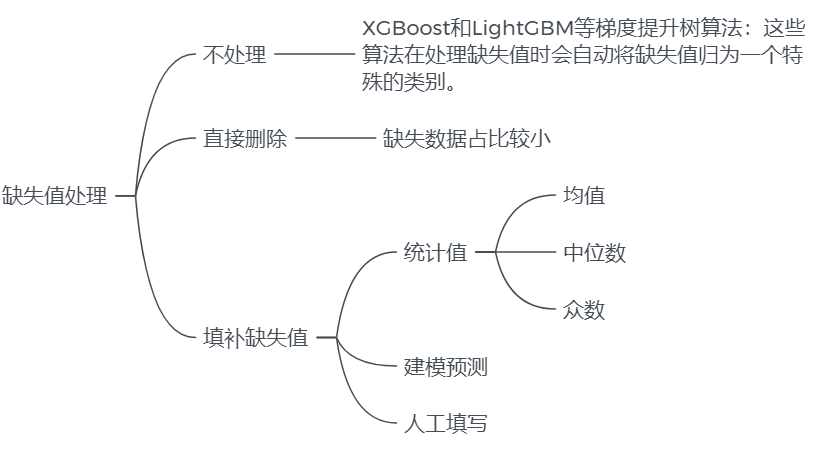


### 2.3 缺失值处理

我们需要检查数据中是否存在缺失值。数据中包含数值型数据和分类型数据。对于数值型数据（age，duration，campaign，pdays，previous，emp.var.rate，cons.price.idx，cons.conf.idx，euribor3m，nr.employed），我们可以使用 isnull() 方法来检查是否存在缺失值。对于分类型数据（），'unknown' 表示数据缺失，我们已将'unknown'替换为NaN，现使用isnull()方法进行统计。

从结果可以看出，数值型数据没有缺失值，分类型中job, marital, education, default, housing, loan中存在缺失值。常见的缺失值处理方法如图所示：



我们将目前存在缺失值的字段分为两种情况进行处理：

* 对于含有缺失值数据占比较小的字段（job 和 marital），我们可以直接删除这些缺失值所在的数据。
* 对于缺失值占比较大的字段（housing，loan，education 和 default），我们可以使用不含缺失值的数据作为训练集，使用随机森林来预测缺失值，并用预测结果来填充缺失值。

## 3 模型建立

### 3.1 建模思路

对于银行营销数据，我们希望可以通过相关信息来确定客户是否会订阅产品（银行定期存款），即我们需要建立模型，使其可以根据输入变量（年龄，职业等）来预测输出变量（客户是否已经订阅定期存款），这是一个分类问题。

分类（classification）是指将数据按照某种标准或属性分成不同的类别或组别的过程。在分类问题中，数据通常被表示为特征（features）或属性（attributes），每个特征或属性具有一个值。分类的任务是对数据集进行学习并构造一个拥有预测功能的分类模型，用于预测未知样本的类标号。

以下是几种常用的分类方法：

决策树（Decision Tree）：通过树状结构来描述各种决策，其中每个叶节点表示一个类别，而非叶节点表示一个判断条件。

朴素贝叶斯（Naive Bayes）：基于贝叶斯定理的分类方法，它假设各个特征之间是相互独立的。

逻辑回归（Logistic Regression）：一种广义线性模型，适用于二元分类或多元分类。

支持向量机（Support Vector Machine）：通过构建最优超平面（或者多个最优超平面）来实现分类。

随机森林（Random Forest）：一种基于决策树的集成学习方法，通过多个决策树的投票来决定分类结果。

K近邻算法（K-Nearest Neighbor，KNN）：根据待分类数据点周围K个最近邻数据点的类别来判断该数据点所属的类别。

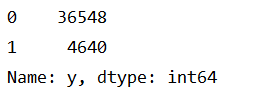
为了寻找最优模型，我们可以尝试多种不同的模型（例如逻辑回归、随机森林、支持向量机等），并对每个模型寻找最优模型参数。然后，将每个模型的预测结果与实际结果进行比较，选择表现最好的模型作为最终的模型。

### 3.2 划分数据集与重采样

在划分数据集时，通常将数据划分为训练集、验证集和测试集三个部分。划分比例的具体选择取决于具体的应用场景和数据量大小等因素，一般而言，通常按照 60%-20%-20% 或 70%-15%-15% 的比例进行划分，我们选择按照60%-20%-20%划分数据集。

训练集用于训练模型，确定模型的参数和结构。通常使用大部分数据作为训练集，以便使模型能够充分学习数据的特征和规律。验证集用于调整模型的超参数，如学习率、正则化参数等。在训练过程中，验证集的表现可以帮助我们选择最好的模型，并进行模型的调整，从而避免过拟合。测试集用于评估模型的泛化性能，检查模型是否过拟合或欠拟合。测试集应该是独立于训练集和验证集的，这样才能客观地评估模型的性能。

在对数据中输出变量y进行统计后发现，负样本（y=0）远多于正样本（y=1），负样本占大多数，这会导致模型的预测结果偏向负样本，而对于正样本的预测不够准确，从而影响了模型的准确性和可靠性。为减弱数据不均衡问题带来的不利影响，我们将采用重采样方法，重采样包括欠采样和过采样两种。



欠采样指的是从多数类（即负例）中删除一些样本，使得正例和负例的数量接近。欠采样可以减少数据量，提高训练速度，但是可能会造成信息损失，且如果欠采样不合理，可能会导致欠拟合的问题。

过采样指的是从少数类（即正例）中生成新的样本，使得正例和负例的数量接近。过采样可以增加数据量，提高模型的泛化性能，但是如果过采样不合理，可能会导致过拟合的问题。

我们将使用SMOTE实现过采样以平衡数据，SMOTE（Synthetic Minority Over-sampling Technique）是根据少数类样本之间的相似性生成新的样本，以达到平衡数据集的目的。具体来说，SMOTE算法的步骤如下：

1.对于每个少数类样本，计算其与所有少数类样本之间的欧氏距离，并将距离最近的K个样本记录下来，这些样本称为其K近邻样本。

2.对于每个少数类样本，从其K近邻样本中随机选择一个样本，然后在少数类样本与该样本之间的连线上随机生成一个新的样本，即生成的新样本的每个特征值都是原始样本和该K近邻样本对应特征值的线性插值。

3.将生成的新样本添加到原始的少数类样本集中，从而构成一个新的平衡的数据集。

### 3.3 模型的训练与评估

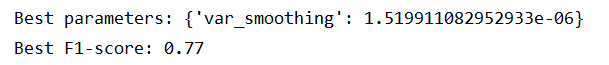
常见的分类模型包括逻辑回归（Logistic Regression）、决策树（Decision Tree）、随机森林（Random Forest）、朴素贝叶斯（Naive Bayes）、支持向量机（Support Vector Machine, SVM）、K近邻算法（K-Nearest Neighbors）、神经网络（Neural Networks）等。我们将选择逻辑回归、朴素贝叶斯、支持向量机和随机森林在训练集上进行训练，并分别进行参数寻优，在交叉验证集上进行评估，随机森林表现更优，所以最终选择在随机森林模型在测试集上进行测试。

通过网格调参得到各模型在训练集上的最优参数：

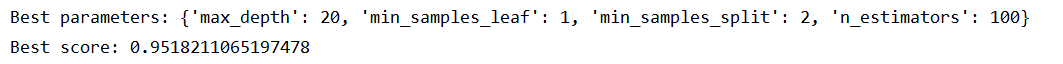
逻辑回归：



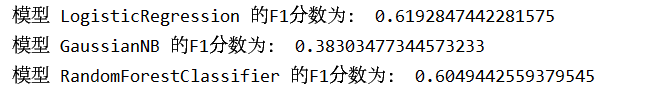
朴素贝叶斯：



随机森林：



使用各模型最佳参数对训练集数据进行训练，并在交叉验证集上进行测试，打印各模型正样本的f1-score：



最后，我们将选择f1得分最高的随机森林模型进行应用，打印正样本的f1-score：



【逻辑回归模型】LogisticRegression()函数常用参数：

penalty：正则化项，可以选择L1或L2，默认是L2。

C：正则化强度的倒数，值越小表示正则化强度越大，默认为1.0。

solver：优化算法选择，包括：'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'。其中newton-cg、lbfgs和sag只适用于penalty为L2的情况，而liblinear则适用于L1和L2正则化。而saga可以用于L1和L2，也支持弹性网络正则化（elastic-net regularization）。

max\_iter：优化算法的最大迭代次数，默认为100。

multi\_class：分类方式选择，'ovr'表示一对多，'multinomial'表示softmax分类。默认是'ovr'。

class\_weight：用于处理不平衡数据集，给不同类别的样本赋予不同的权重。可以是balanced，也可以是自定义的权重。

【朴素贝叶斯】GaussianNB()函数常用参数：

priors: 先验概率，默认为None。可以指定先验概率，例如priors=[0.1, 0.9]表示第一个类别的先验概率为0.1，第二个类别的先验概率为0.9。

var\_smoothing: 方差平滑项，默认为1e-9。用于避免出现方差为0的情况，通过在每个特征的方差上添加一个平滑项来实现。方差平滑项越大，表示对方差的估计更加平滑，模型的复杂度更低，但可能会降低模型的拟合能力。

其中，priors参数用于指定先验概率，var\_smoothing参数用于平滑方差。

【随机森林】RandomForestClassifier()常用参数：

n\_estimators：决策树的数量。

criterion：衡量切分质量的指标，可以选择基尼系数（gini）或信息增益（entropy），默认是gini。

max\_depth：树的最大深度。

min\_samples\_split：分裂内部节点所需的最小样本数。

min\_samples\_leaf：在叶节点上所需的最小样本数。

max\_features：寻找最佳分割时要考虑的特征数量。

【SVM】SVC()函数常用参数：

C：正则化参数，控制对误分类点的惩罚。C值越小，对误分类点的容忍度越高。

kernel：核函数，常用的有线性核函数、多项式核函数、径向基函数（RBF）核函数等。

degree：多项式核函数的次数，仅在kernel为'poly'时生效。

gamma：核函数系数，当kernel为'rbf'、'poly'或'sigmoid'时生效。gamma值越小，高斯核的范围越大，分类曲线变得平缓；gamma值越大，高斯核的范围越小，分类曲线变得更加陡峭。

probability：是否使用概率估计。设为True时，SVM采用Platt缩放计算概率，默认为False。

class\_weight：用于标示不同类别的权重，可以是字典、'balanced'或None。如果是字典，则字典中的键为类别标签，对应的值为该类别的权重。如果是'balanced'，则计算每个类别的权重，使得所有类别的权重之和相等。

decision\_function\_shape：决策函数的形式，可以是'ovr'（一对多）或'ovo'（一对一）。

shrinking：是否使用启发式收缩，True表示使用，False表示禁用。启发式收缩可以加速计算，但在某些情况下可能导致精度下降。

tol：迭代终止条件，当模型在连续max\_iter次迭代中精度变化小于tol时，迭代终止。默认为1e-3。

## 4 展望

最终建立的随机森林模型还有提高的可能性，我们可以考虑以下几个方面来提高F1得分：

* 特征工程：通过选择更好的特征或者构造新的特征，使得模型更能捕捉样本之间的差异和关联，提高模型的分类性能。
* 模型集成：将多个模型进行集成，可以获得更好的分类性能，如投票、平均、加权平均等集成方法。
* 模型选择：尝试使用其他算法或者改进的模型结构，如神经网络、深度学习等，以期获得更好的分类性能。