* 基于银行精准定位客户选择的分类模型选择

陈寒冰 19213147 田瑶 19213224 涂庆 19213225

## 问题描述

### 1.1 问题陈述

有效的营销对于企业增加客户互动和终身价值非常重要。 电话是营销的一种重要方式。 通常，银行为了访问产品(定期存款)是否会被订阅，需要同一个客户有多个联系人,利用电话与顾客进行交流，我们可以根据收集到的这些信息来预测客户是否会预订定期存款(变量y)，找出最有可能预订产品的客户，便于精准营销。

### 1.2 数据集介绍

#### 1.2.1 数据来源

数据来源于UCI数据集，其中数据内容与葡萄牙银行机构的直接营销活动（电话）有关。市场营销活动基于电话。为了访问产品（银行定期存款）是（“是”）还是不（“否”）订阅，通常需要与同一客户进行多次联系。分类的目的是预测客户是否将订阅（是/否）定期存款（变量y）。

#### 1.2.2 数据集内容

原文件中共有四个数据集：

1）带所有示例的bank-additional-full.csv（41188）和20个输入，按日期排序（从2008年5月到2010年11月），非常接近[Moro et al。，2014]中分析的数据。  
2）bank-additional.csv，其中有10％的示例（4119），从1）中随机选择，还有20个输入。  
3）bank-full.csv，包含所有示例和17个输入，按日期排序（此数据集的旧版本，输入较少）。  
4）bank.csv，其中有10％的示例和17个输入，是从3个随机选择的（此数据集的旧版本，输入较少）。

本次作业采用了bank-full.csv数据表的内容进行分类算法模型的测试和对比，其中数据表中共有41188行记录，特征属性有20个，包括job,age,housing等等，标签1个：y-客户是否已订阅定期存款。本次作业采用了常用的分类算法：决策树、随机森林、Adboost、SVM、logistic回归、朴素贝叶斯、神经网络等。

#### 1.2.3 数据特征

根据数据的实际意义，我们将变量主要分为输入变量和输出变量，其中输入变量又可分为三类。

**属性信息:**

**输入变量:**

（1）银行客户信息:

1. Age：年龄
2. Job：工作类型(分类: ’管理员’，‘蓝领’，‘企业家’，‘女佣’，‘管理层’，‘退休’，‘自营职业’，‘服务’，‘学生’，‘技术人员’，‘失业’，‘未知’)
3. Marital： 婚姻状况(分类:“离婚”、“已婚”、“单身”、“未知”；注:“离婚”是指离婚或丧偶)
4. Education：教育程度(分类:“基础. 4y”，“基础. 6y”，“基础. 9y”，“高中”，“文盲”，“专业.课程”，“大学.学位”，“未知”)
5. default：信用违约(分类:“否”、“是”、“未知”)
6. housing：住房贷款(分类:“否”、“是”、“未知”)
7. loan：个人贷款(分类:“否”、“是”、“未知”)#与当前活动的最后一个联系人相关
8. contact：联系方式(分类:“移动电话”、“固定电话”)
9. month：月份，一年中的最后一个联系月(分类:“1月”、“2月”、“3月”，...，”11月”，“12月”)
10. day of week: 一周中的最后一个联系日(分类:“周一”、“周二”、“周三”、“周四”、“周五”)
11. duration：持续时间，上次联系持续时间，以秒为单位(数字)。重要注意事项:该属性对输出目标影响很大(例如，如果持续时间=0，则y=“否”)。然而，在呼叫被执行之前，持续时间是未知的。而且，通话结束后，y显然是已知的。因此，这种输入应该仅出于基准的目的而被包括在内，并且如果意图是具有现实的预测模型，则应该被丢弃。

（2）其他属性

1. campaign：市场活动，在此市场活动期间为该客户执行的联系人数(数字，包括最后一次联系)
2. pdays:上次与客户联系后经过的天数(数字；999表示之前没有联系客户)
3. previous:在此活动之前为该客户执行的联系人数量(数字)
4. poutcome:上一次营销活动的结果(分类:“失败”、“不存在”、“成功”)

（3）社会和经济背景属性

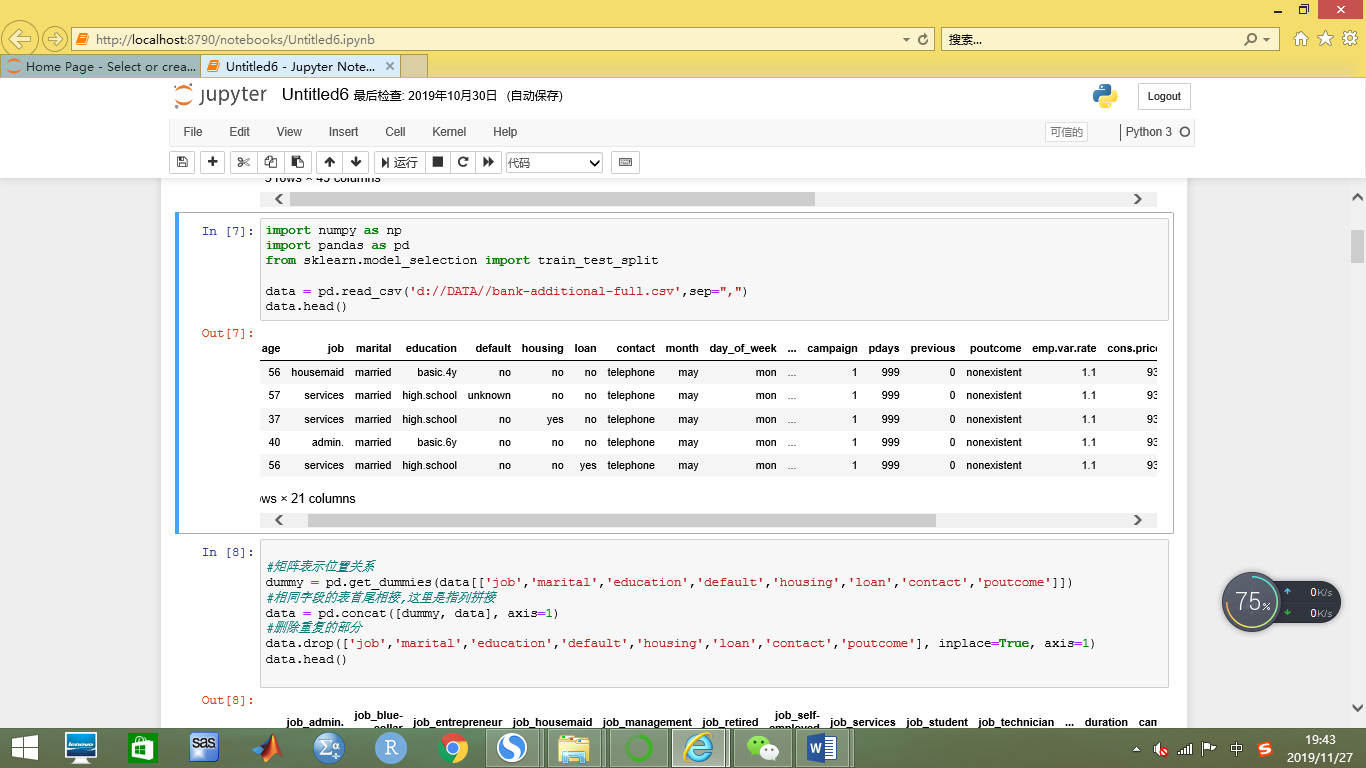
1. emp.var.rate：就业变动率-季度指标(数字)
2. cons.price.idx：消费者价格指数-月度指标(数字)
3. cons.conf.idx：消费者信心指数-月度指标(数字)
4. euribor3m：3个月费率-每日指标(数字)
5. nr.employed：雇员人数-季度指标(数字)

**输出变量(期望目标):**

1. y：客户预订定期存款了吗？(二进制:“是”、“否”)

#### 1.2.4 数据展示

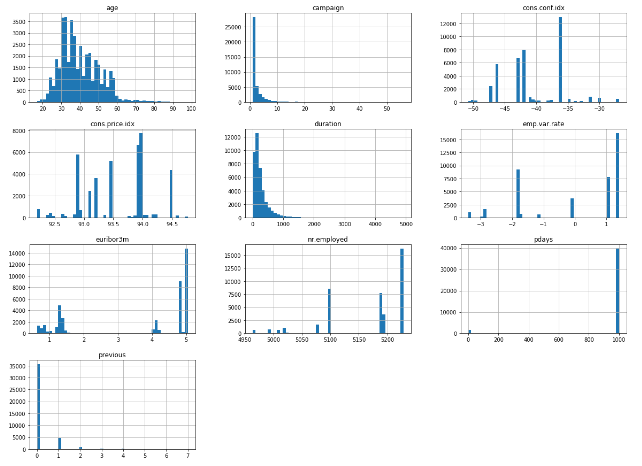
部分样本展示如下



## 数据预处理

### 2.1 数据可视化

#### 2.1.1 原始数据数值型的直方图

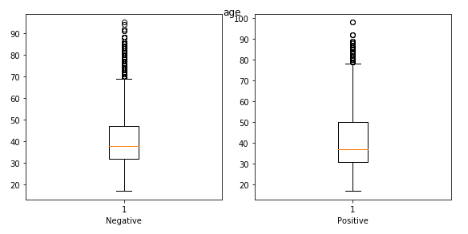


从直方图可以看出。原始数据中基本都是偏态数据，其中标签y中取值为0比取值为1结果为7.876724137931035,数据有不平衡的表现，但是不平衡程度可以接受，所以最后选择的模型评价指标是：模型在训练集和测试集上的表现分数。

#### 2.1.2 非数值型数据的可视化

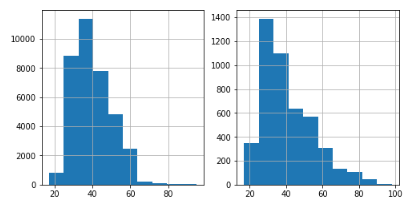
#### 2.1.2.1 属性“age”在不同标签y下的表现

箱线图：



由箱线图可以看出，存在一部分高龄人群，不过我们不认为这是异常值，没做删除，继续保留；

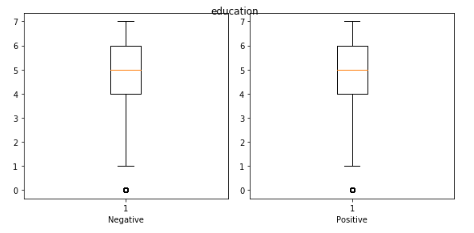
条形图：



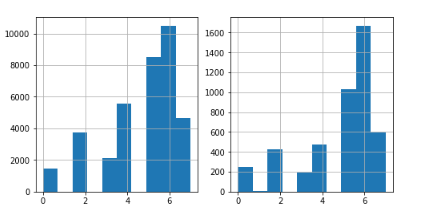
由条形图可以看出在age为20-40岁中，参与订阅（是/否）定期存款的人数最集中，且在两种情况下都呈现右偏；

2.1.2**.2** 属性“education”在不同标签y下的表现

箱线图：

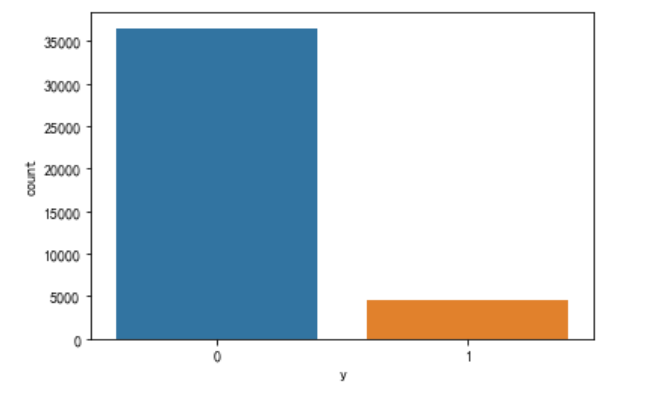


条形图：



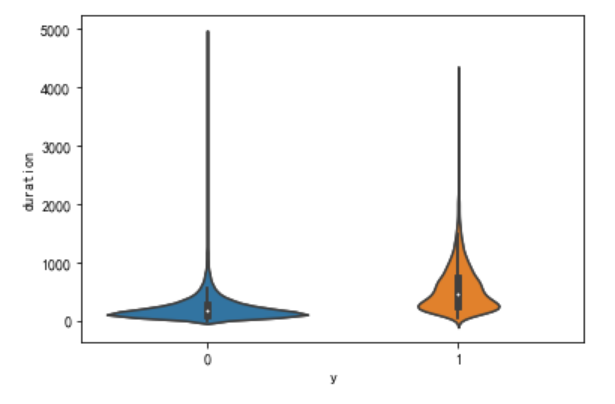
由图可知，学历越高的人参与度越高，且图像呈现左偏趋势。

#### 2.1.3 正负样例可视化



可见正负样例相差很大。

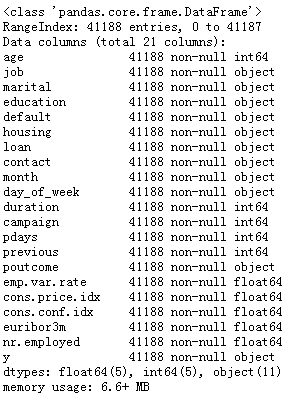
#### 2.1.4通话时长和是否购买关系：



购买用户的通话的平均时长大于不够买用户的平均时长，说明有购买意愿的用户更愿意与工作人员多沟通。

### 2.2 缺失值检查和处理

查看数据信息，发现无缺失值，因此无需处理。



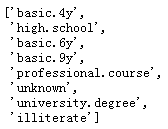
### 2.3 非数值型数据处理

1.标签编码

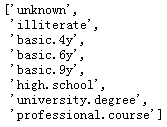
对标签y进行非数值型到数值型表示的LabelEncoder编码，其中‘y=yes’为1，反之为0。

2.特征专用编码

（1）能够将分类特征转换为分类数值，其中对‘education’按照年级从低到高进行排序，起初该属性的取值为：



排序后特征的取值排列为：

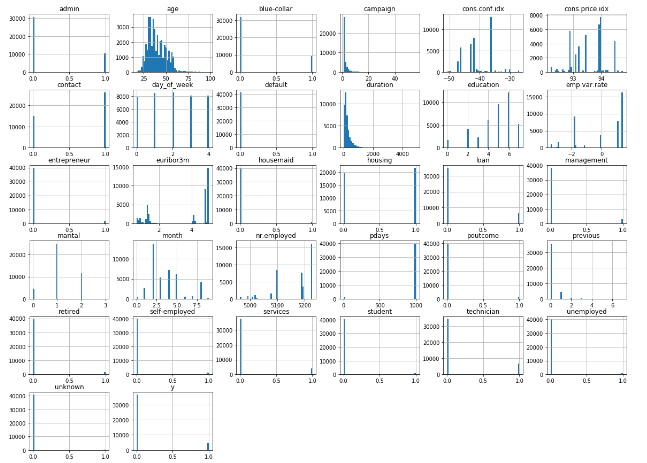


最后按照排序的索引对属性‘education’从0到7进行编码，其中也包含了数量关系；同样的处理day\_of\_week、month（按照现实顺序排序）

（2）将布尔值转化marital、housing、loan、default、contact、poutcome，形式用1,0表示；

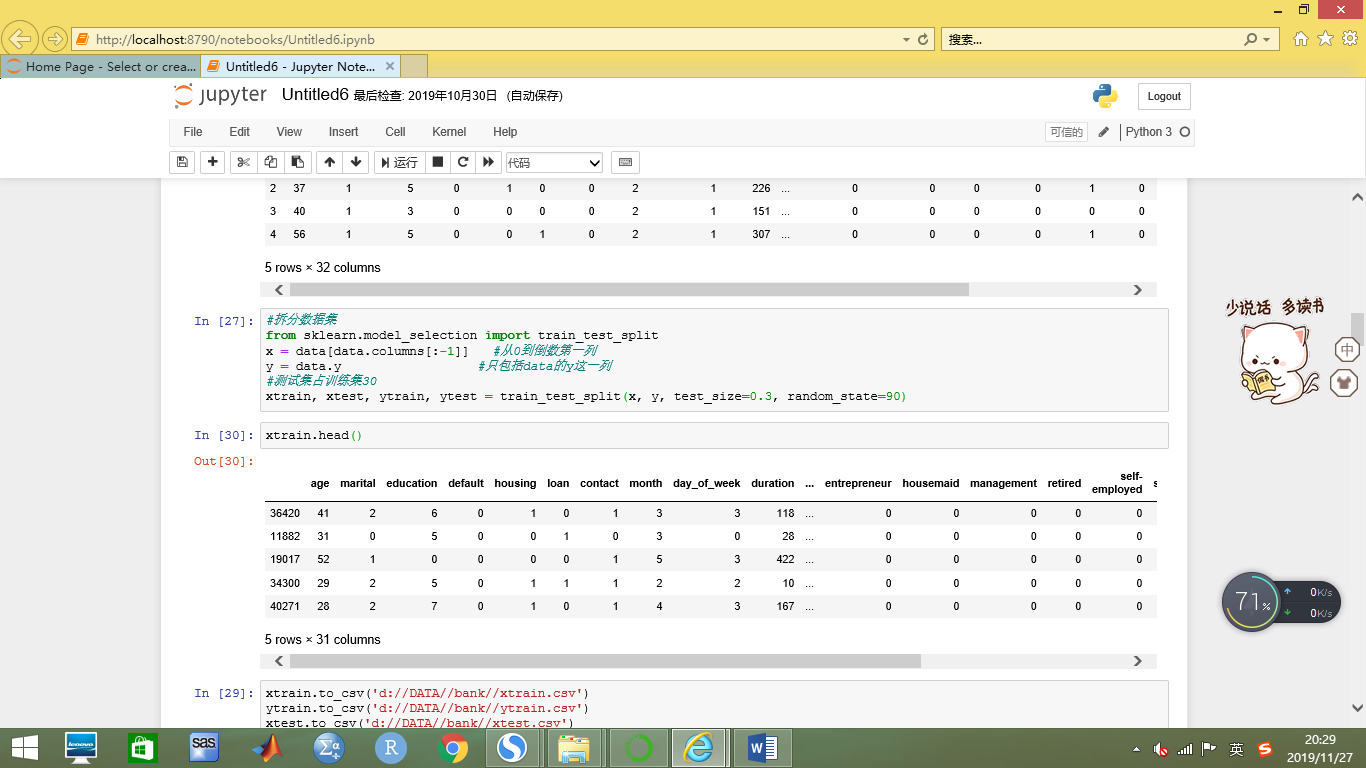
3.独热编码

名义变量，需要使用独热编码，将特征表示都转换为哑变量。这样的转换是让算法能够彻底领悟，特征取值是没有可计算性质的，是“有你就没有我”的不等概念。其中对‘job’属性使用了独热编码，最终我们的数据集从（41188，21）变为（41188，32）。

最后得出的数据集的所有特征的直方图如下：

### 2.5 划分训练集和测试集

直接从sklearn.model\_selection 导入函数 train\_test\_split，设置随机种子为90，训练集和测试集的样本比例为7:3，其中训练集中有28831个样本，测试集中有12357个样本。



## 分类算法模型

### 3.1 决策树模型

#### 3.1.1决策树模型原理

决策树（Decision Tree）是一种非参数的有监督学习方法，它能够从一系列有特征和标签的数据中总结出决策规则，并用树状图的结构来呈现这些规则，以解决分类和回归问题。决策树算法容易理解，适用各种数据，在解决各种问题时都有良好表现。决策树模型的基本流程如下：

决策树算法的核心是要解决两个问题：  
1）如何从数据表中找出最佳节点和最佳分枝？

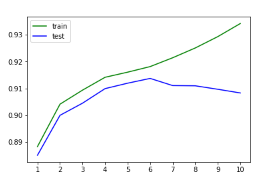
决策树需要找出最佳节点和最佳的分枝方法，对分类树来说，衡量这个“最佳”的指标叫做“不纯度”。通常来说，不纯度越低，决策树对训练集的拟合越好。在sklearn库中Criterion这个参数正是用来决定不纯度的计算方法的。sklearn提供了两种选择：  
1）输入‘entropy’，使用信息熵（Entropy）  
2）输入‘gini’，使用基尼系数（Gini Impurity）

以信息熵作为指标时，决策树的生长会更加“精细”，因此对于高维数据或者噪音很多的数据，信息熵很容易过拟合，基尼系数在这种情况下效果往往比较好。当模型拟合程度不足的时候，即当模型在训练集和测试集上都表现不太好的时候，使用信息熵。  
2）如何让决策树停止生长，防止过拟合？

为了让决策树有更好的泛化性，我们要对决策树进行剪枝。剪枝策略对决策树的影响巨大，正确的剪枝策略是优化决策树算法的核心。Sklearn库中主要的剪枝参数有：

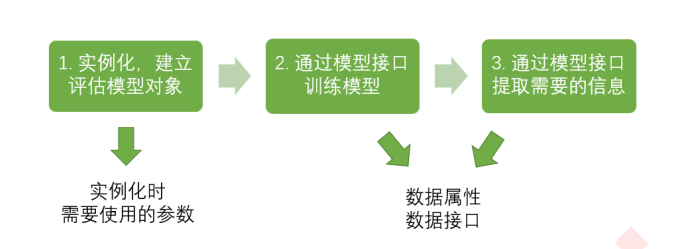
（1）max\_depth: 限制树的最大深度，超过设定深度的树枝全部剪掉，是用得最广泛的剪枝参数，在高维度低样本量时非常有效。决策树多生长一层，对样本量的需求会增加一倍，所以限制树深度能够有效地限制过拟合；

模型训练过程的max\_depth调参，采用了学习曲线的形式，数据集训练之后最终得到的max\_depth=6，学习曲线展示如下：



（2）min\_samples\_leaf & min\_samples\_split：可以保证每个叶子的最小尺寸，限制分枝的条件。

#### 3.1.2决策树模型训练结果

在jupyter notebook的sklearn库中有相应的决策树算法包，其实现的大致流程如下：

模型训练结果如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 分类决策树 | 0.9177968159273 | 0.9144614388605 | 223 ms ± 8.24 ms |

由表中的结果可以看出：

（1）决策树模型在训练集和测试集上的表现差异不大，可认为模型不存在过拟合现象；

（2）决策树模型在训练集和测试集上的表现分数均在0.91左右，说明模型分类效果较好；

（3）决策树模型的时间成本很低，运行时间不到1秒钟。

### 3.2 随机森林和Adaboost算法

#### 3.2.1随机森林和Adaboost算法

集成学习（ensemble learning）是时下非常流行的机器学习算法，它本身不是一个单独的机器学习算法，而是通过在数据上构建多个模型，集成所有模型的建模结果。本次作业主要采用了随机森林（装袋法典型）和Adaboost算法（提升算法典型），两个集成算法的区别如下图所示：

其中：

（1）装袋法的核心思想是构建多个相互独立的评估器，然后对其预测进行平均或多数表决原则来决定集成评估器的结果。装袋法的代表模型就是随机森林。

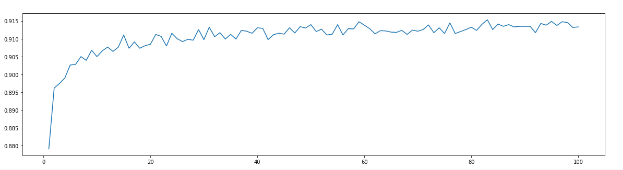
（2）提升法中，基评估器是相关的，是按顺序一一构建的。其核心思想是结合弱评估器的力量一次次对难以评估的样本进行预测，从而构成一个强评估器。提升法的代表模型有Adaboost、GBDT、Xgboost。

集成算法的核心是要解决以下问题：

（1）如何保证集成后的分类效果好于单个基分类器的效果？

Sklearn库中参数的调整，

1. n\_estimators：这是森林中树木的数量，即基评估器的数量。这个参数对随机森林模型的精确性影响是单调的，n\_estimators越大，模型的效果往往越好。对其画学习曲线来确定的最优取值，

随机森林：n\_estimators=83，图像如下：

2.random\_state: 控制的是生成森林的模式。

#### 3.2.2模型训练结果

Adaboost算法训练结果

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 随机森林 | 0.9934098713190 | 0.9060451565914 | 863 ms ± 21.7 ms |
| Adaboost | 0.9998265755618 | 0.8997329448895 | 49 s ± 530 ms |

（1）随机森林和Adaboost算法在调整影响模型效率的参数之后，在训练集和测试集上的表现差异依旧大，猜测模型存在过拟合现象；

（2）随机森林模型的时间成本不到一秒，Adaboost算法的时间成本为49s左右。

（3）随机森林和Adaboost算法在测试集上的表现分数均在0.9左右，低于决策树模型的分类效果。

### 3.3 支持向量机模型

#### 3.3.1 SVM原理

SVM 是有监督的学习模型，我们需要事先对数据打上分类标签，通过求解最大分类间隔来求解二分类问题。如果要求解多分类问题，可以将多个二分类器组合起来形成一个多分类器。SVM：Support Vector Machine中文名为支持向量机，在机器学习中，SVM 是有监督的学习模型。

用 SVM 计算的过程就是帮我们找到那个超平面（决策线C放到三维空间中，便是一个平面，这个决策面就是超平面）的过程，这个超平面就是我们的 SVM 分类器。

SVM 特有的一个概念：分类间隔。在极限位置B和A之间，存在多个最优决策面，而那个拥有“最大间隔”（max margin）的决策面就是SVM 要找的最优解。

对一个数据点进行分类，当超平面离数据点的“间隔”越大，分类的确信度（confidence）也越大。所以，为了使得分类的确信度尽量高，需要让所选择的超平面能够最大化这个“间隔”值。这个间隔如下图中的gap/2所示。

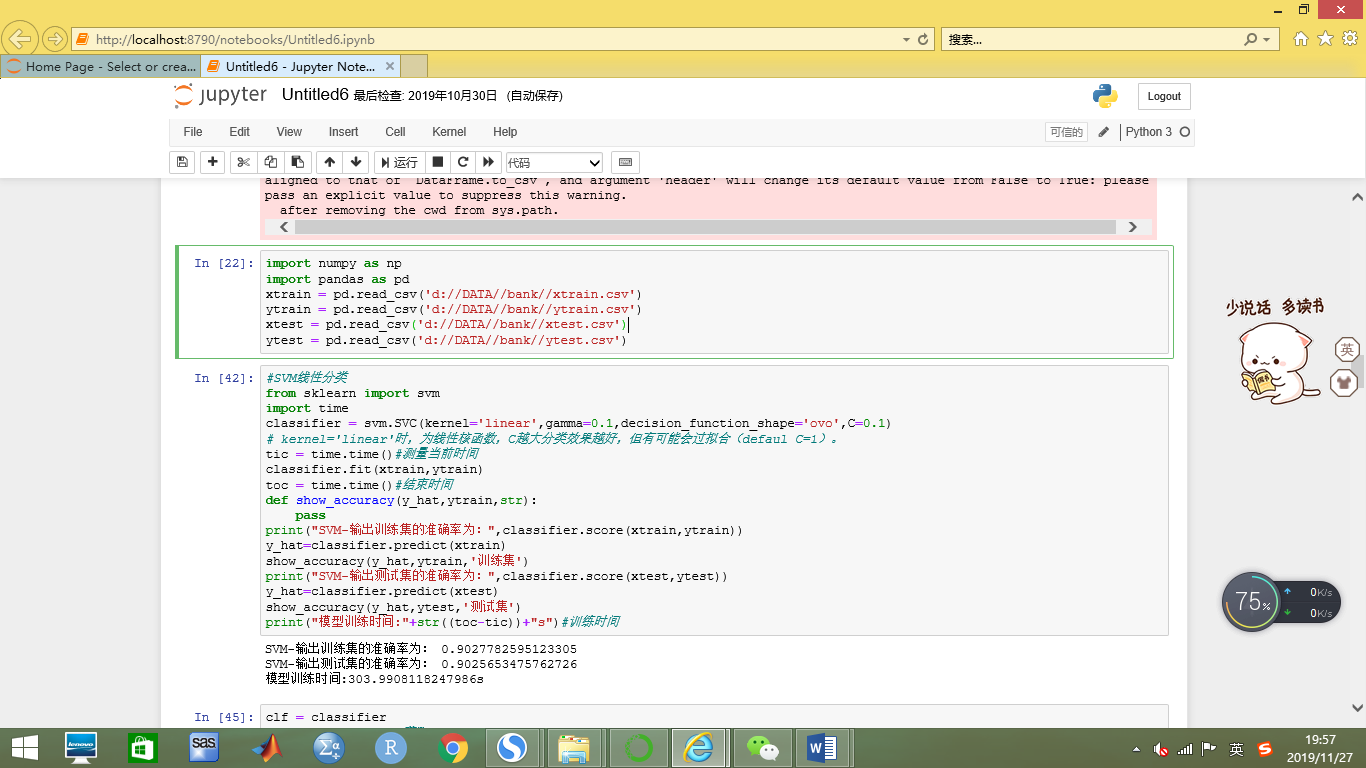


#### 3.3.2 SVM训练结果

选择核函数为线性函数，调用sklearn包中的svm对训练集进行训练，得到一个线性分类器

模型训练结果如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 支持向量机 | 0.9027782595123305 | 0.9025653475762726 | 303.9908118247986s |

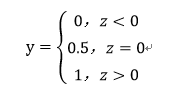


### 3.4 Logistic回归模型

#### 3.4.1 Logistic回归原理

Logistic 回归为概率型非线性回归模型，Logistic 回归虽然名字叫“回归”，但却是一种分类学习方法。使用场景大概有两个：第一用来预测，第二寻找因变量的影响因素，是研究二分类或多分类观察结果之间关系的一种多变量分析方法。Logistic 回归法是一种广义的线性回归分析模型，常用于数据挖掘。

Logistic回归是处理二分类问题的，所以输出的标记y={0,1}，并且线性回归模型产生的预测值z=wx+b是一个实值，所以我们将实值z转化成0/1值便可，这样有一个可选函数便是“单位阶跃函数”：



这种如果预测值大于0便判断为正例，小于0则判断为反例，等于0则可任意判断！

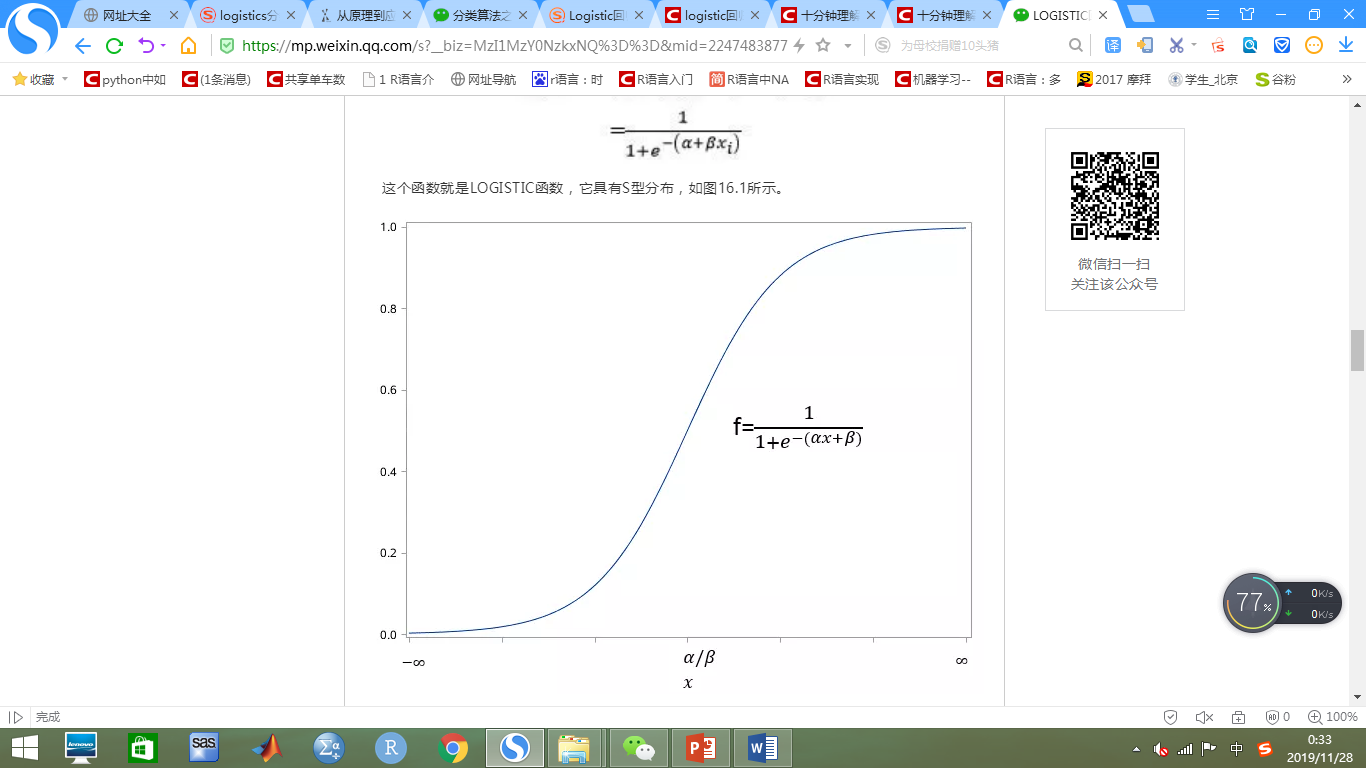
Logistic 回归通过使用其固有的 logistic 函数估计概率，来衡量因变量（我们想要预测的标签）与一个或多个自变量（特征）之间的关系。

然后这些概率必须二值化才能真正地进行预测。这就是 logistic 函数的任务，也称为 Sigmoid 函数。Sigmoid 函数是一个 S 形曲线，它可以将任意实数值映射到介于 0 和 1 之间的值，但并不能取到 0或1。然后使用阈值分类器将 0 和 1 之间的值转换为 0 或 1。

下面的图片说明了 logistic 回归得出预测所需的所有步骤。



下面是 logistic 函数（sigmoid 函数）的图形表示：

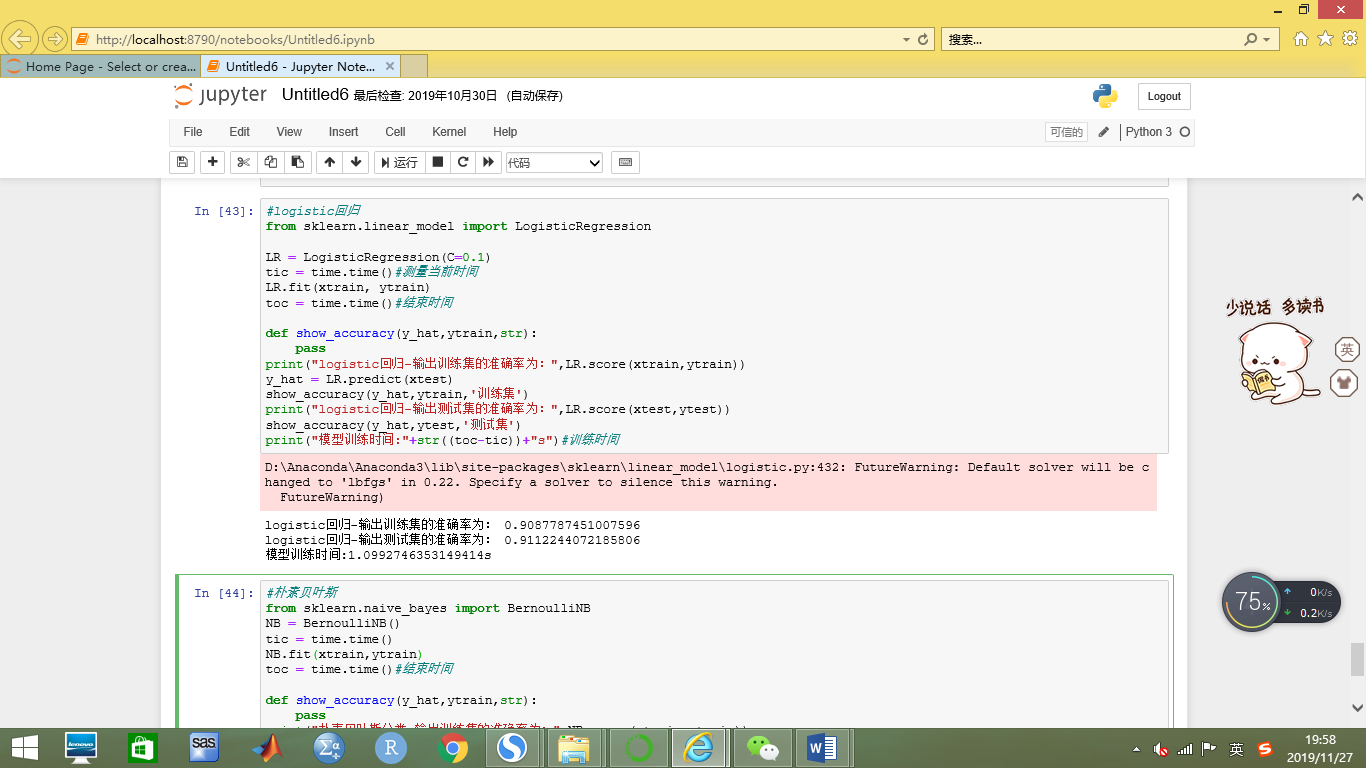


#### 3.4.2 Logistic回归训练结果

利用sklearn包中的LogisticRegression函数对训练集数据进行回归分析，得到一个logistic回归，将测试集放入模型中得到的准确率为91.12%，略优于SVM模型，与决策树分类算法结果相近，训练时间为1.099秒，远远小于SVM，效果较好。

模型训练结果如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| logistic回归 | 0.9087787451007596 | 0.9112244072185806 | 1.0992746353149414s |



### 3.5 朴素贝叶斯模型

#### 3.5.1 朴素贝叶斯原理

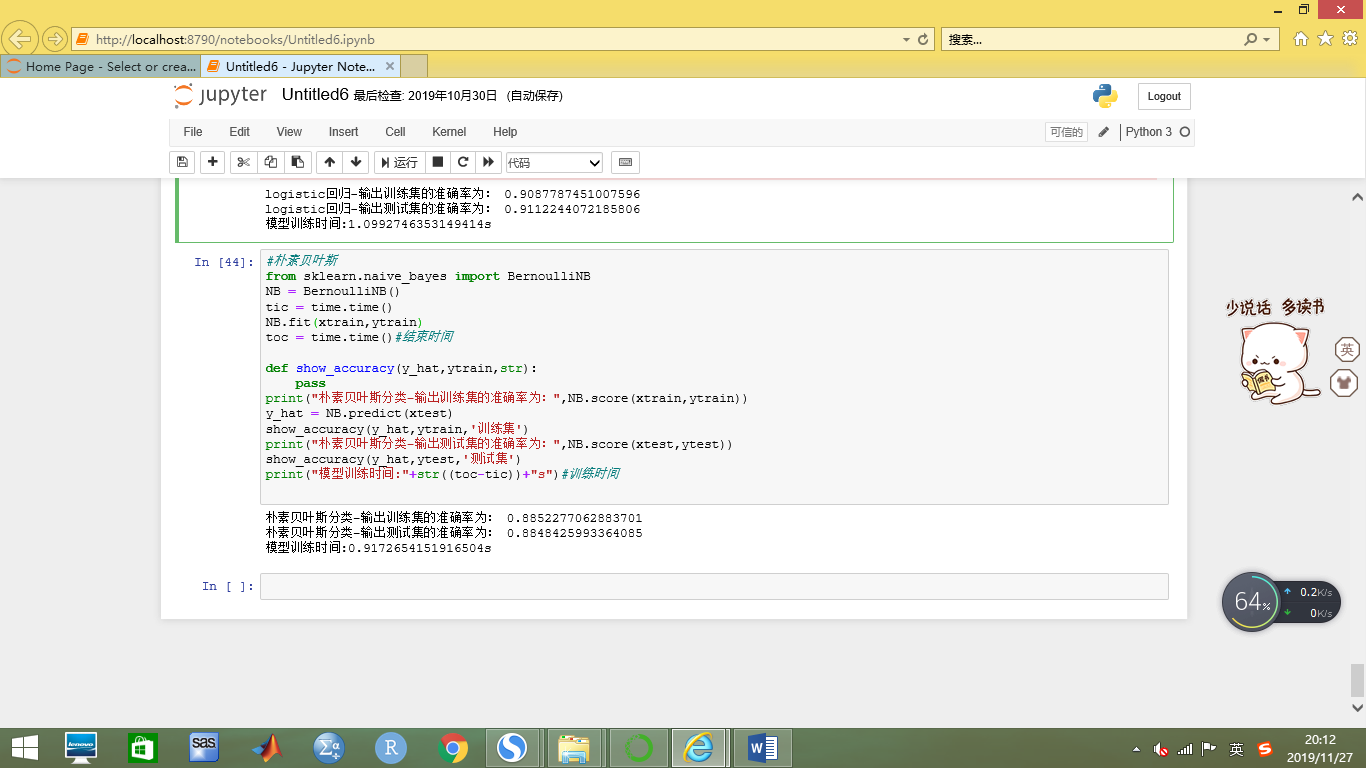
朴素贝叶斯（Naive Bayesian）是基于贝叶斯定理和特征条件独立假设的分类方法，它通过特征计算分类的概率，选取概率大的情况进行分类，因此它是基于概率论的一种机器学习分类方法。因为分类的目标是确定的，所以也是属于监督学习。

#### 3.5.2 朴素贝叶斯训练结果

利用sklearn包中的BernoulliNB函数对训练集数据进行回归分析，得到一个朴素贝叶斯分类器，将测试集放入模型中得到的准确率为88.48%，训练时间为0.91秒，处理时间较快，但效果相较而言略差。

模型训练结果如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 朴素贝叶斯 | 0.8852277062883701 | 0.8848425993364085 | 0.9172654151916504s |



### 3.6 神经网络模型与其它三种模型对比

#### 3.6.1神经网络原理

感知机是由两层神经元组成的网络模型，如图3所示。

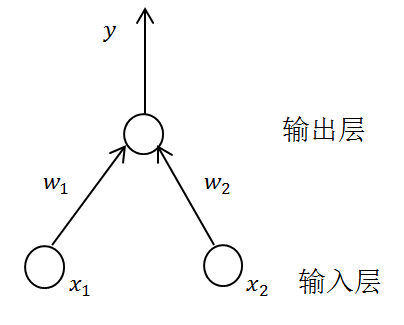


图 两个输入神经元的感知机网络图

感知机能够非常方便地进行逻辑运算。

假定是图2中的阶跃函数，有

“与”（与）：令，仅在时，

“或” （或）：令，当时，

“非” （非）：令，当时，[20]

如果给定了训练数据集，通过学习，可以得出阈值和权重（）。阈值可以看作一个输入为-1.0的固定的“哑结点”，它的连接权重令为。感知机的学习规则非常的容易理解，对于训练的样本（），假如现在感知机的输出为,那么感知机权重将进行这样的修正：

其中称为学习率。从(1)可得，如果感知机对训练样本做出了正确的预测结果，即，那么感知机将不会变化，相反感知机会根据预测结果错误程度来修正权重。

学习率需要预先指定，对于学习率的选择来说，就好比显微镜对焦。过大过小都会产生不好的效果。学习率的选择通常取决于经验给定。

上述逻辑中的与、或、非问题都属于线性可分的。能够证明，如果两种类型全部是线性可分的，那么感知机必将收敛，且适当的权向量也可求得：

关于非线性可分的问题，则需要使用如图4所示的多层的神经元。在图4中，中间层的那一层神经元称为隐层神经元，隐层神经元和输出层神经元都属于功能神经元，且都有激活函数。

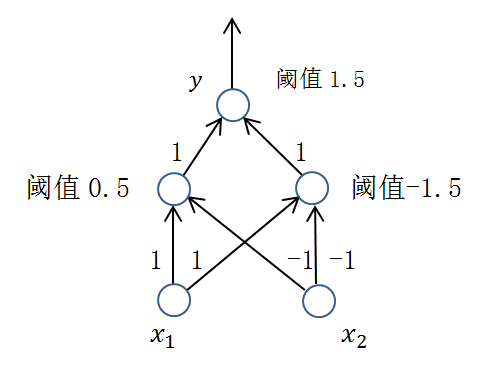
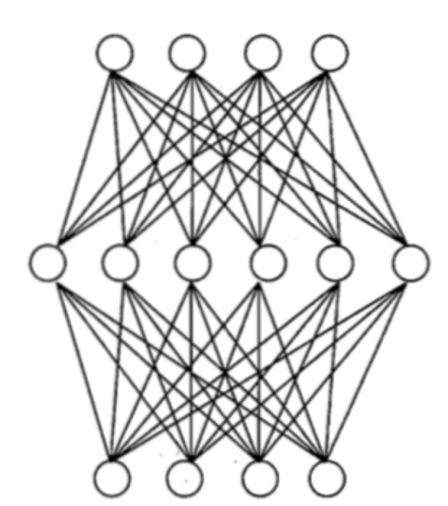
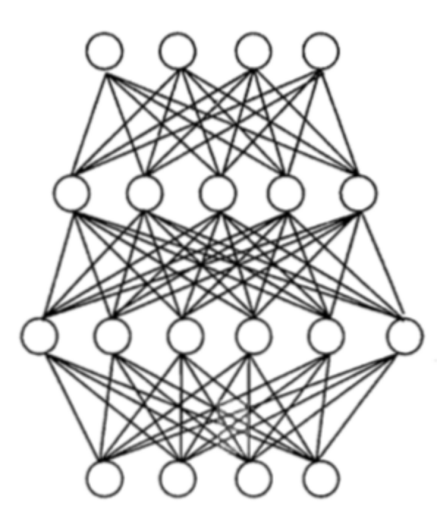


图 能解决异或问题的两层感知机网络结构示意图

图5所示的层级结构是常叫做“多层前馈神经网络”。每层神经元与下一层神经元全部相互连接，输入层神经元负责接收输入，也就是说，输入层神经元不会进行任何的函数处理，而仅仅是接收输入，功能神经元存在于隐层和输出层中。所以，图5(a)通常被称为“单隐层网络”或“两层网络”。包含了隐层的神经网络就能叫做多层神经网络。

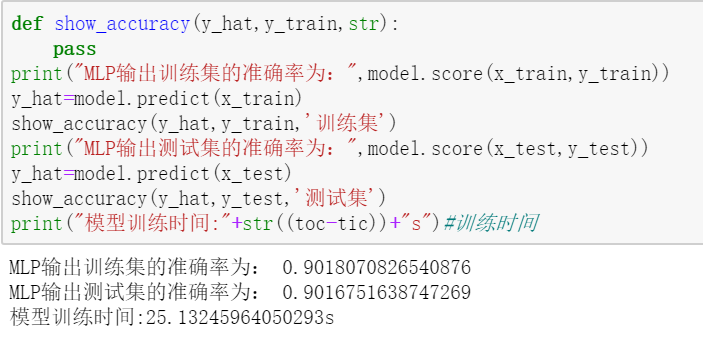
(a)单隐层前馈网络 (b)双隐层前馈网络

图 多层前馈型神经网络的结构示意图

# 3.6.2神经网络训练结果

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 神经网络 | 0.9018070826540876 | 0.9016751638747269 | 25.13245964050293s |





## 模型比较与总结

经过数据预处理，将原始数据转化为质量较高的数据集后，再利用七种不同的分类模型进行建模和预测，最终的结果见下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 训练集效率 | 测试集效率 | 时间成本 |
| 决策树 | 0.9177968159273 | 0.9144614388605 | 0.2170124053955078s |
| 随机森林 | 0.9934098713190 | 0.9060451565914 | 0.7600436210632324s |
| Adaboost | 0.9998265755618 | 0.8997329448895 | 42.28241848945618s |
| SVM | 0.90277825951233 | 0.90256534757627 | 303.9908118247986s |
| Logistic | 0.90877874510075 | 0.91122440721858 | 1.0992746353149414s |
| 朴素贝叶斯 | 0.88522770628837 | 0.88484259933641 | 0.9172654151916504s |
| 神经网络 | 0.901807082654 | 0.901675163874 | 25.1324596405029s |

由表我们可以得出以下结果：

1.模型效率：随机森林和Adaboost训练效率最高，但是存在过拟合现象；决策树测试集效率表现最佳；

2.运行时间：决策树最短，SVM最长；

3.朴素贝叶斯效率最低，因为NB更适合小样本数据集训练模型（训练集：测试集=7:3）；通过分类模型，当使用朴素贝叶斯模型效率很低时，应该考虑如Logistic、决策树、SVM等分类模型。

4.由于数据集不是很大，神经网络对数据量要求较大，所以在本数据集上表现不是最优；

最终选用效率最佳——决策树模型：用于预测银行精准定位客户选择是否订阅定期存款。