

# **《大数据分析方法》**

**( 2021年秋季学期 )**

**翟祥**  
**北京林业大学**

**E-mail: zhaixbh@126.com**

## 第2章

# 大数据商业应用



## 第2章 大数据商业应用

2.1 用户画像和精准营销

2.2 广告推荐

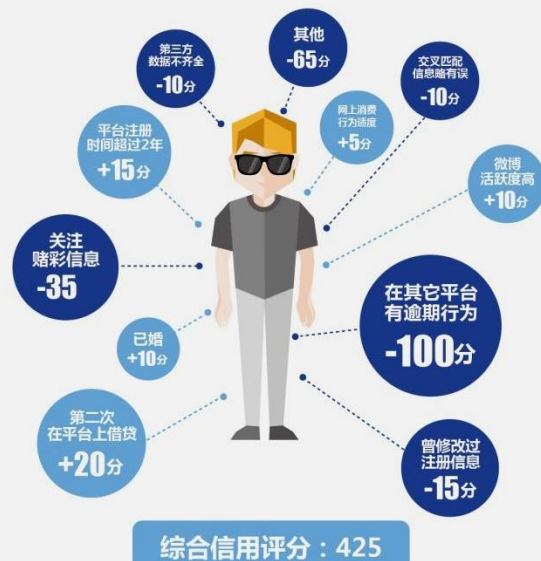
2.3 互联网金融

2.4 实战：个人贷款风险评估

## 2.1 用户画像和精准营销

### ● 2.1.1 用户画像概述

人在网络世界中的行为集合代表了他在网络世界中的“性格”，这个集合就描述了他的网络个性和用户特征（User Profile）。从数据拥有者，也就是企业角度来看，他们掌握了所有用户在网络世界中“某方面”的行为习惯，如用户浏览了哪些网页、搜索了哪些关键词、购买了哪些商品、留下了哪些评价等，企业都会收集汇总。如何将如此庞杂的数据转换为商业价值，成为现在企业越来越关注的问题。面对高质量、多维度的海量数据，如何建立精准的用户模型就显得尤为重要，用户画像的概念也就应运而生。



甲: 年龄 28岁 信用评分(低)



乙: 年龄 34岁 信用评分(高)

**用户画像**，即用户信息的标签化，是企业通过收集、分析用户数据后，抽象出的一个虚拟用户，可以认为是真实用户的虚拟代表。用户画像的核心工作就是为用户匹配相符的标签，通常一个标签被认为是人为规定的高度精练的特征标识。

1

用户画像从多维度对用户特征进行构造和刻画，包括用户的社会属性、生活习惯、消费行为等，进而可以揭示用户的性格特征。有了用户画像，企业就能真正了解了用户的所需所想，尽可能做到以用户为中心，为用户提供舒适快捷的服务。

2

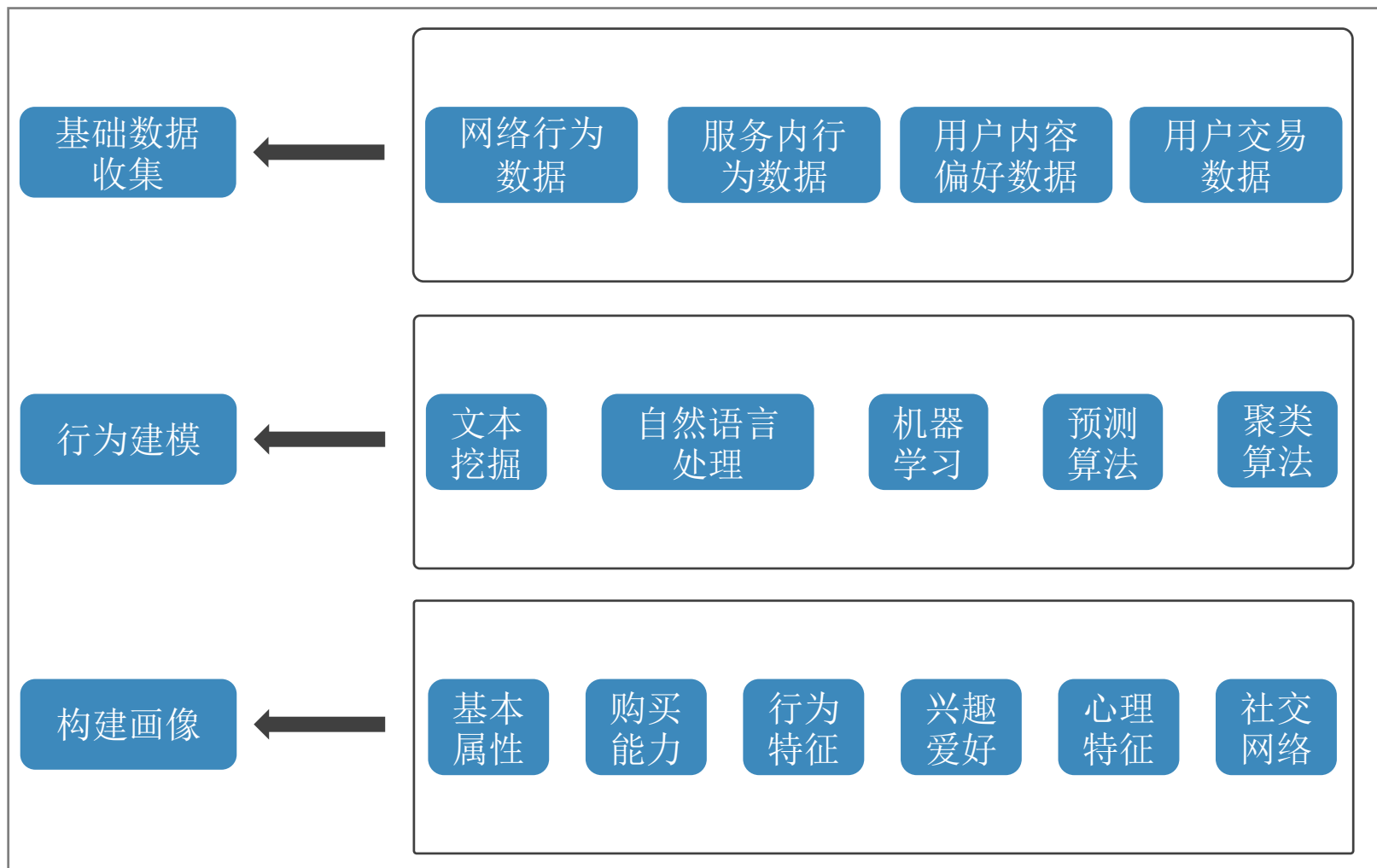
用户画像技术通过对用户的分析，让企业对用户的精准定位成为了可能。在这个基础上，依靠现代信息技术手段建立个性化的顾客沟通服务体系，将产品或营销信息推送到特定的用户群里中，既节省营销成本，又能起到最大化的营销效果。

## 2.1 用户画像和精准营销

### ● 2.1.2 用户画像的价值



### ● 2.1.3 用户画像构建流程



### 01

### 数据收集与分析

构建用户画像是为了将用户信息还原，构建一个用户数据模型。因此这些数据是基于真实的用户数据。用户数据可以大致分为网络行为数据、服务内行为数据、用户内容偏好数据、用户交易数据这四类。

- ◆ 网络行为数据：活跃人数、页面浏览量、访问时长、激活率、外部触点、社交数据等
- ◆ 服务内行为数据：浏览路径、页面停留时间、访问深度、页面浏览次数等
- ◆ 用户内容偏好数据：浏览/收藏内容、评论内容、互动内容、生活形态偏好、品牌偏好等
- ◆ 用户交易数据(交易类服务)：贡献率、客单价、连带率、回头率、流失率等
- ◆ 当然，收集到的数据不会是100%准确的，都具有不确定性，这就需要在后面的阶段中建模来再判断，比如某用户在性别一栏填的男，但通过其行为偏好可判断其性别为“女”的概率为80%。



### 02

### 数据建模

该阶段是对上阶段收集到数据的处理，进行行为建模，以抽象出用户的标签，这个阶段注重的应是大概率事件，通过数学算法模型尽可能地排除用户的偶然行为。

这时也要用到机器学习，对用户的行为、偏好进行猜测，好比一个  $y=kx+b$  的算法， $x$  代表已知信息， $y$  是用户偏好，通过不断的精确  $k$  和  $b$  来精确  $y$ 。

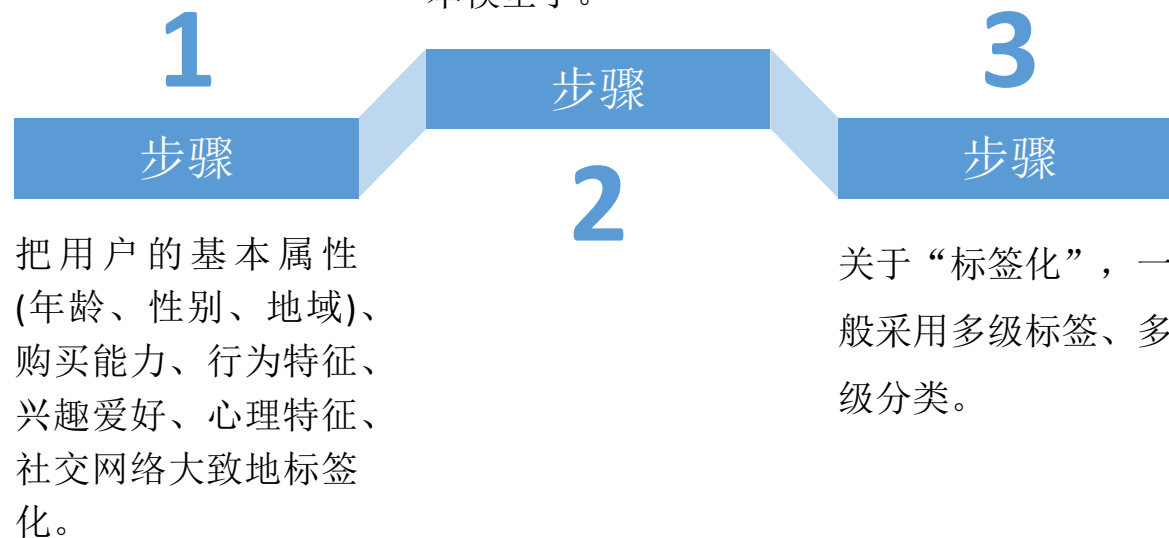
在这个阶段，需要通过定性与定量相结合的研究方法来建立很多模型来为每个用户打上标签以及对应标签的权重。

定性化研究方法就是确定事物的性质，是描述性的;定量化研究方法就是确定对象数量特征、数量关系和数量变化，是可量化的。

### 03

### 构建用户画像

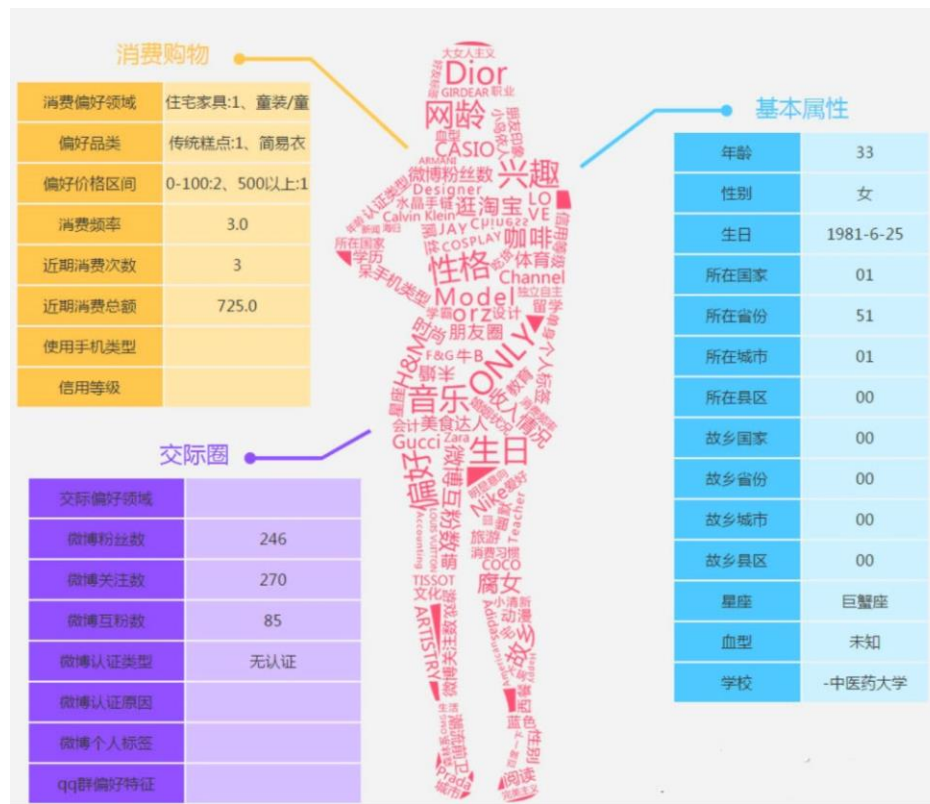
当一切数据标签化并赋予权重后，即可根据构建用户画像的目的来搭建用户画像基本模型了。



### 04

### 数据可视化分析

如图所示，这是把用户画像真正利用起来的一步，在此步骤中一般是针对群体的分析，比如可以根据用户价值来细分出核心用户、评估某一群体的潜在价值空间，以做出针对性的运营。



### ● 2.1.4 用户标签体系

从技术层面看，用户画像的过程比较乏味。但如何设计用户画像的标签体系却是一个看起来最简单、却最难以把握精髓的环节。

#### 问题

#### 什么是标签体系？

简单说就是你把用户分到多少个类里面去。当然，每个用户是可以分到多个类上的。这些类都是什么，彼此之间有何联系，就构成了标签体系。

标签体系的设计有两个常见要求，一是便于检索，二是效果显著。在不同的场景下，对这两点的要求重点是不同的。

一般来说，设计一个标签体系以下三种思路。

# 1. 结构化标签体系

结构化标签体系看起来整洁，又比较好解释，在面向品牌广告主交流时比较好用。性别、年龄这类人口属性标签，是最典型的结构化体系。

一级标签	二级标签
Finance	Bank Accounts, Credit Cards, Investment, Insurance, Loans, Real Estate, ...
Service	Local, Wireless, Gas & Electric, ...
Travel	Europe, Americas, Air, Lodging, Rail, ...
Tech	Hardware, Software, Consumer, Mobile, ...
Entertainment	Games, Movies, Television, Gambling, ...
Autos	Econ/Mid/Luxury, Salon/Coupe/SUV, ...
FMCG	Personal care, ...
Retail	Apparel, Gifts, Home, ...
Other	Health, Parenting, Moving, ...

Yahoo! 用户标签体系图

### 2. 半结构化标签体系

在用于效果广告时，标签设计的灵活性大大提高了。标签体系是不是规整，就不那么重要了，只要有效果就行。在这种思路下，用户标签往往是在行业上呈现出一定的并列体系，而各行业内的标签设计则以“逮住老鼠就是好猫”为最高指导原则，切不可拘泥于形式。

类别	描述	数据来源	用户规模
Intent	最近输入词表现出某种产品或服务需求的用户	BlueKai Intent	160+MM
B2B	职业上接近某种需求的用户	Bizo	90MM
Past Purchase	根据以往消费习惯判断可能购买某产品的用户	Addthis, Alliant	65+MM
Geo/Demo	地理上或人口属性上接近某标签的用户	Bizo, Datalogix, Expedia	
Interest/LifeStyle	可能喜欢某种商品或某种生活风格的用户	Forbes, i360, IXI, ...	103+MM
Qualified Demo	多数据源上达成共识验证一致的人口属性	多数据源	90+MM
Estimated Financial	根据对用户财务状况的估计作出的分类	V12	

半结构化标签体系图

### 3. 非结构化标签体系

非结构化，就是各个标签就事论事，各自反应各自的用户兴趣，彼此之间并无层级关系，也很难组织成规整的树状结构。非结构化标签的典型例子，是搜索广告里用的关键词。还有Facebook用的用户兴趣词，意思也一样。

半结构化标签操作上已经很困难了，非结构化的关键词为什么在市场上能够盛行呢？这主要是因为搜索广告的市场地位太重要了，围绕它的关键词选择和优化，已经形成了一套成熟的方法论。

## 第2章 大数据商业应用

2.1 用户画像和精准营销

2.2 广告推荐

2.3 互联网金融

2.4 实战：个人贷款风险评估



### ● 2.2.1 推荐系统

**个性化推荐**在我们的生活中无处不在。早餐买了几根油条，老板就会顺便问一下需不需要再来一碗豆浆；去买帽子的时候，服务员会推荐围巾。随着互联网的发展，这种**线下推荐**也逐步被搬到了**线上**，成为各大网站吸引用户、增加收益的法宝。



推荐系统的性能可以通过以下几个标准来判定

用户满  
意度

覆盖率

预测准  
确度

冷启动  
问题

过度推  
荐热门  
问题

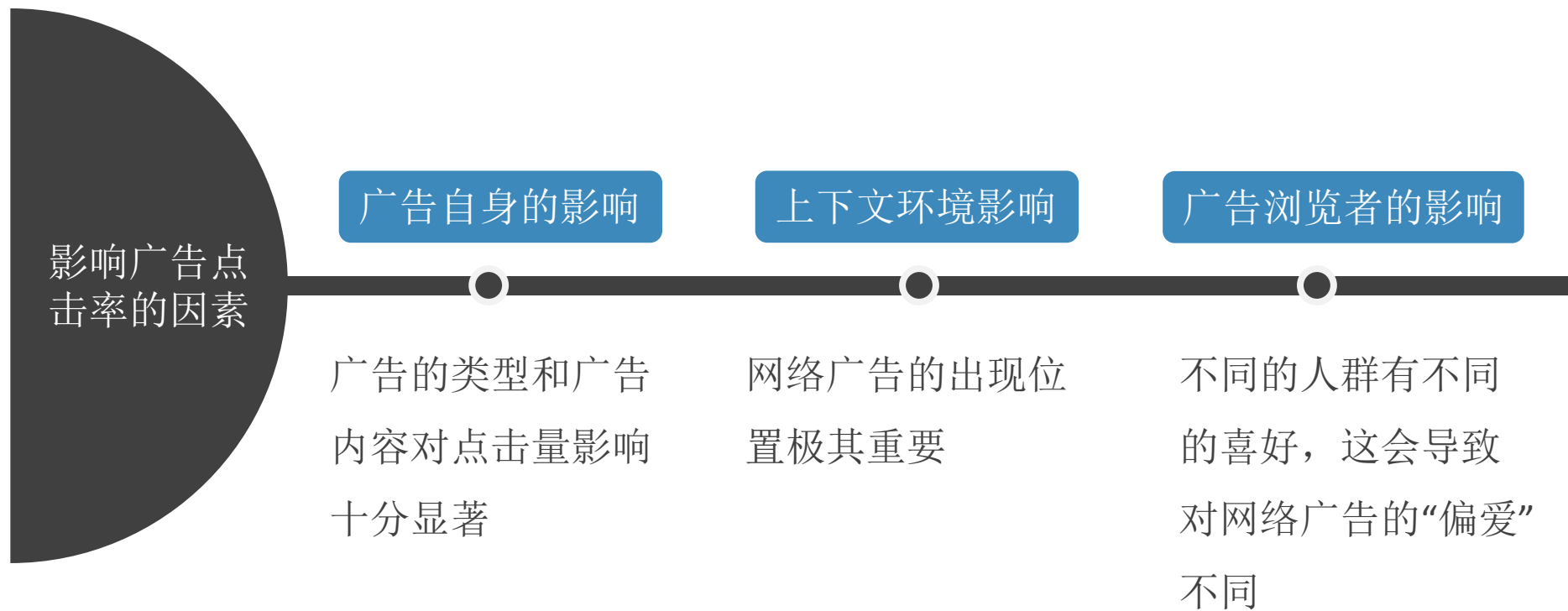
个性化  
评价



### ● 9.2.2 广告点击率及其评估

评价一个网络广告推广效果好坏的测量指标是多样的，例如，可以通过广告展示量、广告点击量、广告到达率、广告转化率等指标进行评价。其中，**广告点击率（Click-Through-Rate, CTR）**是当前最为普遍的评价方式，是反应网络广告推广质量最直接的量化指标。广告点击率的计算公式为如下：

$$\text{广告点击率 (CTR)} = \frac{\text{广告的点击次数}}{\text{广告的展示次数}}$$



## 广告点击率预估

对广告的点击率进行预测是十分有必要的。**对展示广告的网站来说**，针对不同页面、不同人群精准投放不同广告，可以使广告和网页做到紧密结合，使广告“无痕植入”，使浏览者在潜移默化中接受广告，提高广告被点击的可能性；**对商家来说**，不仅可以预估广告带来的收益，及时对广告进行调整，提升收益，还可以减少一些不必要的投放，减少支出；**对浏览者来说**，广告的精准投放更易被接受，不容易引起反感，增加点击广告的可能性。

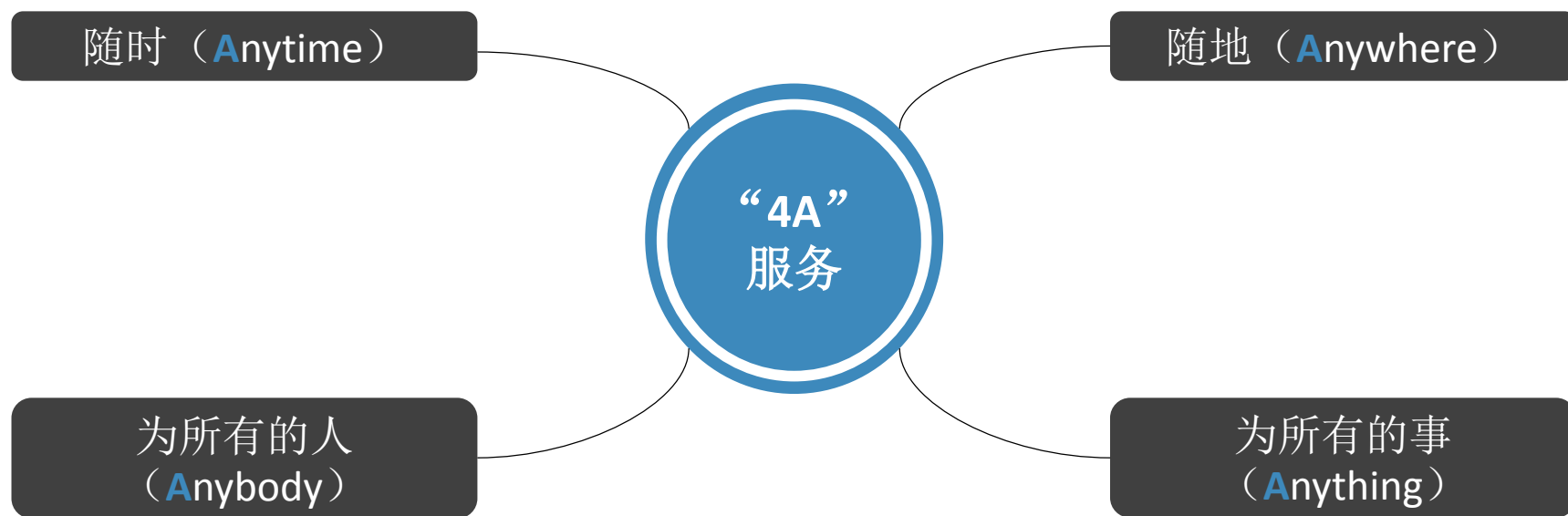
### (1) 直接估算法

$$p(x=k) = \binom{n}{k} p^k (1-p)^{n-k} = b(k; n, p) (k=0, 1, \dots, n)$$

### (2) 点击率预估模型计算方法

$$P(y=1|x) = \frac{1}{1 + e^{-(\beta + \sum \beta_i x_i)}}$$
$$\ln\left(\frac{p_1}{p_0}\right) = \beta_0 + \sum \beta_i x_i$$

### ● 2.2.3 基于位置的服务和广告推荐





### 基于位置服务的关键技术

定位  
技术

电子地  
图技术

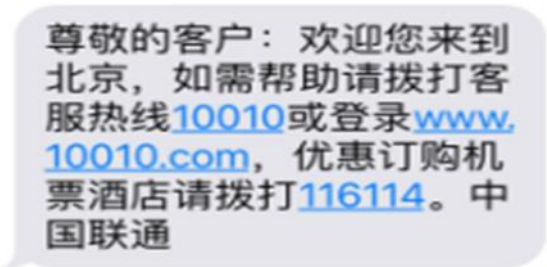
数据分  
析、挖  
掘技术

- (1) 定位技术：定位技术是基于位置服务的基础，目的是获取终端设备的物理位置。
- (2) 电子地图技术：电子地图是定位信息的载体，可以将位置信息直观、形象地展示给用户，可以将平面的地图“立体化”。目前成熟的电子地图有Google Map、高德地图、Bing Map等。
- (3) 数据分析与数据挖掘技术：对获取的数据进行分析和挖掘是提供多元化服务的基础。例如，借助驾驶人的日常轨迹对其推荐他日常经过的商店和产品等。

### 基于位置的广告推荐

与传统互联网广告不同，基于位置的广告推荐更多地会考虑“位置”这一选择条件，优先推荐当前地点附近的商家或产品，实现更加精准且个性化的广告投放，不仅能极大地提升用户体验，还可以迅速将用户从网上吸引到实体店面内，完成从线上到线下的无缝对接。

(1) “主动式”也称“推”式，指广告服务提供商根据用户所在位置，主动向客户发送广告，直到用户取消广告订阅或将广告屏蔽为止[10]。图7所示为主动式基于位置的广告推荐实例。



尊敬的客户：欢迎您来到北京，如需帮助请拨打客服热线10010或登录[www.10010.com](http://www.10010.com)，优惠订购机票酒店请拨打116114。中国联通

主动式基于位置的广告推荐实例



(2) “被动式”也称“拉”式，指用户通过关键词发起搜索，推荐系统根据搜索关键词、用户当前地理位置信息和用户其他特征返回出推荐结果。



被动式基于位置的广告推荐实例

由于基于位置的广告推荐一般通过移动智能设备获取用户位置，而此类设备一般都处于开机状态，故可以持续获取用户位置，这也为分析用户移动轨迹、分析用户习惯、建立用户画像奠定了基础。

由于涉及用户位置等隐私信息，基于位置的广告推荐服务的隐私问题备受关注。另外，如果广告发送频率过于频繁，用户会产生对广告的厌烦情绪，此时广告提供商应加强广告质量审查、合理控制广告发送频率。

## 第2章 大数据商业应用

2.1 用户画像和精准营销

2.2 广告推荐

2.3 互联网金融

2.4 实战：个人贷款风险评估

### ● 2.3.1 概述

互联网金融是指以依托于**支付、云计算、社交网络**以及搜索引擎等互联网工具，实现资金融通、支付和信息中介等业务的一种新兴金融。互联网金融是在实现安全、移动等网络技术水平上，被用户熟悉接受后自然而然为适应新的需求而产生的新模式及新业务。

“三步走战略”——平台、数据、金融



平台、数据、金融相互影响的格局



在这种形势下破局的点在哪里？就在于连接平台、用户、金融等方面的工具——大数据



### 2.3.2 大数据在互联网金融的应用方向

#### 1. 金融反欺诈与分析

金融企业通过收集和凝聚多方位的数据源信息形成精准全面的反欺诈信息库和反欺诈用户行为画像，结合大数据分析技术和机器学习算法进行欺诈行为路径的分析和预测，并对欺诈触发机制进行有效识别。

#### 2、构建更全面的信用评价体系

(1)构建完备的信用数据平台；  
(2)融合金融企业专业量化的信用模型和基于互联网的进货、销售、支付清算、物流等交易积累数据；  
(3)应用大数据技术进行信用模型的分布式计算部署，快速响应，高效评价，快速放款。

#### 3、高频交易和算法交易

高频交易主要采取“战略顺序交易”，即通过分析金融大数据，以识别出特定市场参与者留下的足迹。

#### 4、产品和服务的舆情分析

金融机构借助舆情采集与分析技术，抓取来自社交网站、论坛、贴吧和新闻网站的与金融机构及产品相关的信息，并数据挖掘算法进行分词、聚类、特征提取、关联分析和情感分析等，找出金融企业及其产品的市场关注度、评价正负性等信息。

### ● 2.3.3 机器学习在大数据金融中的作用

- ◆ 在企业数据的应用的场景下，人们最常用的主要是监督学习和无监督学习的模型，在金融行业中一个天然而又典型的应用就是风险控制中对借款人进行信用评估。因此互联网金融企业依托互联网获取用户的网上消费行为数据、通讯数据、信用卡数据、第三方征信数据等丰富而全面的数据，可以借助机器学习的手段搭建互联网金融企业的大数据风控系统。
- ◆ 除了在放贷前的信用审核外，互联网金融企业还可以借助机器学习完成传统金融企业无法做到的放贷过程中对借款人还贷能力进行实时监控，以及时对后续可能无法还贷的人进行事前的干预，从而减少因坏账而带来的损失。
- ◆ 目前互联网金融企业以及第三方征信公司在信用评估这方面比较常用的架构是规则引擎加信用评分卡。



### 1、信用评分算法

GBDT(Gradient Boosting Decision Tree)又叫MART(Multiple Additive Regression Tree), 该模型不像决策树模型那样仅由一棵决策树构成, 而是由多棵决策树构成, 通常都是上百棵树, 而且每棵树规模都较小(即树的深度会比较浅)。模型预测的时候, 对于输入的一个样本实例, 首先会赋予一个初值, 然后会遍历每一棵决策树, 每棵树都会对预测值进行调整修正, 最后得到预测的结果。

$$F(x)=F_0+\beta_1 T_1(x)+\beta_2 T_2(x)+\dots+\beta_m T_m(x)$$

其中,  $F_0$ 为设置的初值,  $T_i$ 是一棵棵的决策树(弱的分类器)。

GBDT作为一种boosting算法, 自然包含了boosting的思想, 即将一系列弱分类器组合起来构成一个强分类器。它不要求每个分类器都学到太多的东西, 只要求每个分类器都学一点点知识, 然后将这些学到的知识累加起来构成一个强大的模型。



### 2、分类模型的性能评估

分类模型应用较多的除上面讲的Logistic Regression和GBDT，还有Decision Tree、SVM、Random forest等。实际应用中不仅要知道会选用这些模型，更重要的是要懂得对所选用的模型的性能做评估与监控。

涉及到评估分类模型的性能指标有很多，常见的有Confusion Matrix(混淆矩阵),ROC,AUC, Recall, Performance, lift, Gini ,K-S之类。其实这些指标之间是相关与互通的，实际应用时只需选择其中几个或者是你认为是重要的几个即可，无须全部都关注。

### (1) 混淆矩阵的概念

混淆矩阵是监督学习中的一种可视化工具，主要用于比较分类结果和实例的真实信息。

矩阵中的每一行代表实例的预测类别，每一列代表实例的真实类别。

Confusion Matrix		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

混淆矩阵

- ◆ 真正率(True Positive Rate , TPR) 【灵敏度(sensitivity)】：  $TPR = TP / (TP + FN)$  ，即正样本预测结果数/ 正样本实际数
- ◆ 假负率(False Negative Rate , FNR)：  $FNR = FN / (TP + FN)$  ，即被预测为负的正样本结果数/正样本实际数
- ◆ 假正率(False Positive Rate , FPR)：  $FPR = FP / (FP + TN)$  ，即被预测为正的负样本结果数 /负样本实际数
- ◆ 真负率(True Negative Rate , TNR) 【特指度(specificity)】：  $TNR = TN / (TN + FP)$  ，即负样本预测结果数 / 负样本实际数



## (2) 由混淆矩阵计算评价指标

基于以上混淆矩阵，可以引申出以下指标进一步评价分类器性能：

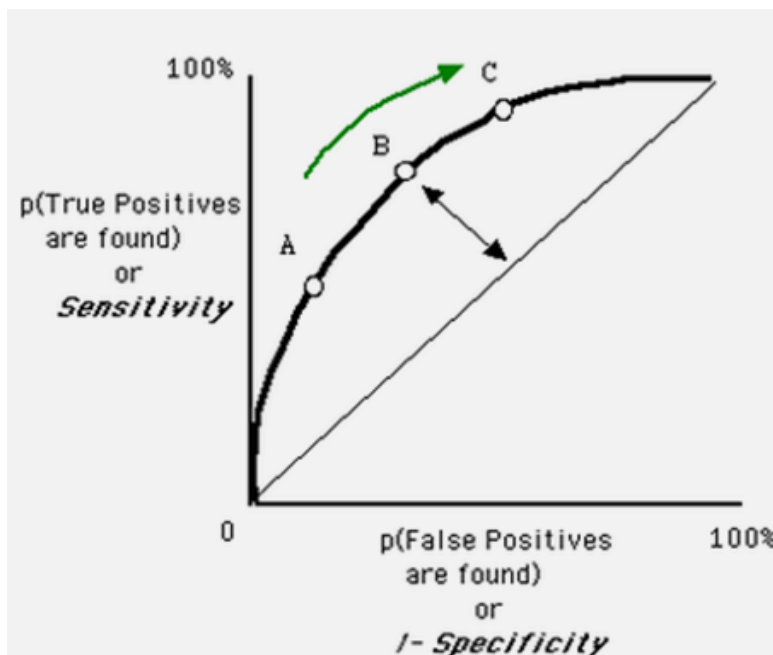
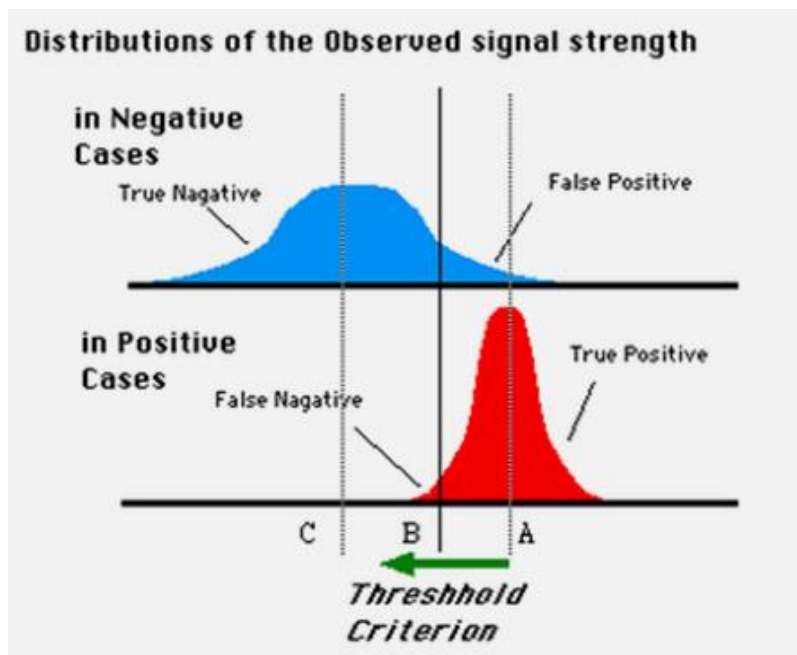
- ◆ 精确度(Precision):  $P = TP/(TP+FP)$
- ◆ 召回率(Recall):  $R = TP/(TP+FN)$ ，即真正率
- ◆ F-score: 查准率和查全率的调和平均值, 更接近于P, R两个数较小的那个:  $F=2 * P * R / (P + R)$
- ◆ 准确率(Aaccuracy): 分类器对整个样本的判定能力,即将正的判定为正，负的判定为负:  $A = (TP + TN)/(TP + FN + FP + TN)$
- ◆ 灵敏度(Sensitivity): 将正样本预测为正样本的能力,  $Sensitivity=TP/(TP+FN)$ ;
- ◆ 特异度(Specificity): 将负样本预测为负样本的能力,  $Specificity=TN/(TN+FP)$ ;
- ◆ AUC(Area Under roc Curve)值指处于ROC曲线下方的那部分面积大小；一个理想的分类模型其AUC值为1，通常其值在0.5至1.0之间，较大的AUC代表了分类模型具备较好的性能。

### ROC(Receiver Operating Characteristic):

ROC的主要分析工具为画在ROC空间的曲线，横轴为 $1 - \text{Specificity}$ ，纵轴为Sensitivity。

在分类问题中，一个阈值对应于一个特异性及灵敏度，并在ROC空间描出一个点P，当阈值连续移动时，P点也随即移动最终绘成ROC曲线。

ROC良好的刻画了不同阈值对样本的分辨能力，也同时反应出对正例和对反例的分辨能力，方便使用者根据实际需求选用合适的阈值。一个好的分类模型要求ROC曲线尽可能靠近图形的左上角；



## 第2章 大数据商业应用

2.1 用户画像和精准营销

2.2 广告推荐

2.3 互联网金融

2.4 实战：个人贷款风险评估

### 1、实战目的

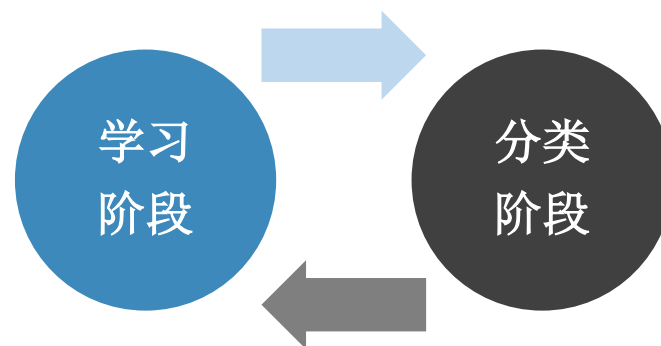
本次实验通过提取贷款用户相关特征（年龄、工作、收入等），使用Spark MLlib构建风险评估模型，使用相关分类算法将用户分为不同的风险等级，此分类结果可作为银行放贷的参考依据。本次实验为方便演示，选用逻辑回归算法将用户风险等级分类两类：高风险、低风险。有能力的同学可以尝试使用其他分类算法实现。

### 2、实验环境和实验数据

- ◆ 操作系统：CentOS6.5。
- ◆ 编程语言：Scala 2.10.4。
- ◆ 相关软件：Hadoop2.6.0、Spark1.6.0。
- ◆ 实验数据来源：<https://www.kaggle.com/>，数据内容解释详见书本。

### 3、实验过程

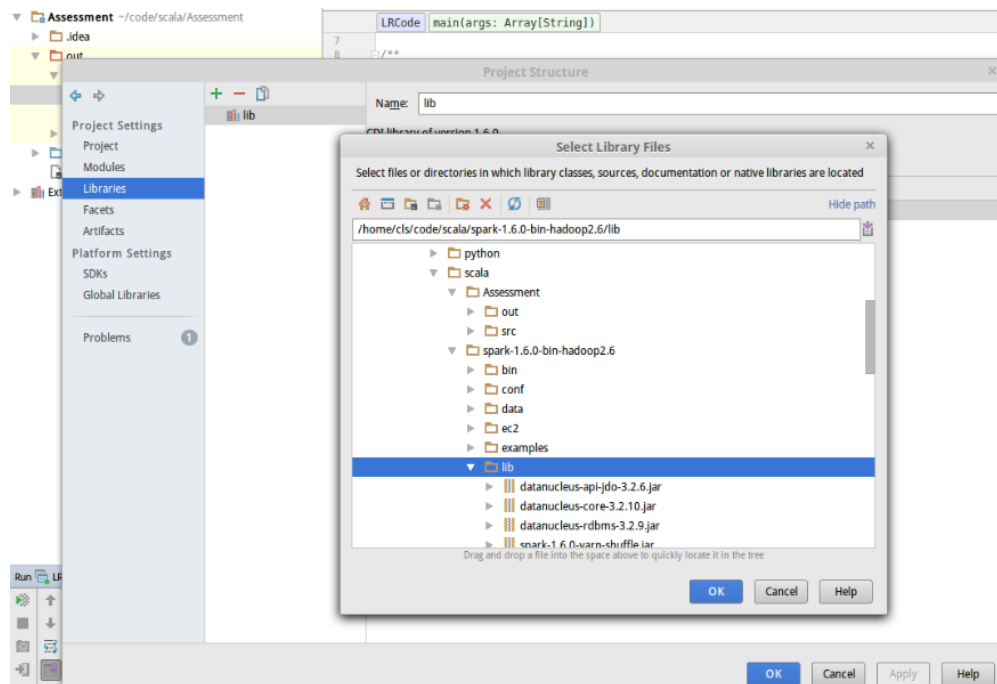
选定样本数据  
提取样本数据特征  
生成测试报告  
评估分类器性能



新样本进行特征提取  
对样本数据进行分类

### 4、实验步骤

#### (1) IDEA配置：



在IntelliJ IDEA中需要导入Spark开发包，Spark/lib中的jar包能满足基本的开发需求，开发者可以在菜单：File->project stucture->Libraries中设置。



## 2.4 实战：个人贷款风险评估

### (2) 代码步骤：

#### 第一页代码

获取数据：

```
val path = "hdfs://master:8020/input/adult.csv"
```

```
val rawData = sc.textFile(path)
```

简单的数据清洗。

```
/**
```

```
 * 取第一列为类标，其余列作为特征值
```

```
*/
```

```
val data = records.map{ point =>
```

```
  val firstdata = point.map(_._replaceAll(" ", ""))
```

```
  val replaceData = firstdata.map(_._replaceAll(",", " "))
```

```
  val temp = replaceData(0).split(" ")
```

```
  val label = temp(0).toInt
```

```
  val feature s = temp.slice(1, temp.size-1)
```

```
    .map(_._hashCode)
```

```
    .map(x => x.toDouble)
```

```
  LabeledPoint(label, Vectors.dense(features))
```

```
}
```

按照一定的比例将数据随机分为训练集和测试集。

这里需要程序开发者不断的调试比例以达到预期的准确率，值得注意的是，不当的划分比例导致“欠拟合”或“过拟合”的情况产生。

```
val splits = data.randomSplit(Array(0.8, 0.2), seed = 11L)
```

```
val training = splits(0).cache()
```

```
val test = splits(1)
```





## 2.4 实战：个人贷款风险评估

### 第二页代码

训练分类模型。

```
val model = new  
LogisticRegressionWithLBFGS().setNumClasses(2).run(t  
raining)
```

预测测试样本的类别。

```
val predictionAndLabels = test.map{  
case LabeledPoint(label,features) =>  
val prediction = model.predict(features)  
  (prediction,label)  
}
```

计算并输出准确率。

```
val metrics = new  
BinaryClassificationMetrics(predictionAndLabels)  
val auRoc = metrics.areaUnderROC()  
println("Area under Roc =" + auRoc)
```

输出权重最大的前10个特征。

```
val weights = (1 to model.numFeatures) zip  
model.weights.toArray  
println("Top 5 features:")  
weights.sortBy(_._2).take(5).foreach{case(k,w) =>  
println("Feature " + k + " = " + w)  
}
```

保存与加载模型。

```
val modelPath = "hdfs://master:8020/output/"  
model.save(sc, modelPath)  
val sameModel =  
LogisticRegressionModel.load(sc,modelPath)
```



## 2.4 实战：个人贷款风险评估

(3) 代码实例：

第  
一  
页  
代  
码

```
import org.apache.spark.mllib.classification.LogisticRegressionModel
import org.apache.spark.mllib.classification.LogisticRegressionWithLBFGS
import org.apache.spark.mllib.evaluation.{BinaryClassificationMetrics, MulticlassMetrics}
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.log4j.{Level, Logger}
import org.apache.spark.mllib.linalg.Vectors

object LRCode {
  def main(args:Array[String]): Unit = {
    val conf = new SparkConf()
      .setAppName("Logisitic Test")

    .setMaster("spark://master:7077")
    val sc = new SparkContext(conf)

    //屏蔽不必要的日志信息
    Logger.getLogger("org.apache.spark").setLevel(Level.WARN)
    Logger.getLogger("org.eclipse.jetty.server").setLevel(Level.OFF)

    //使用MLUtils对象将hdfs中的数据读取到RDD中
    val path = "hdfs://master:8020/input/adult.csv"
    val rawData = sc.textFile(path)
```

### (3) 代码实例：

```
val startTime = System.currentTimeMillis()
println("startTime:"+startTime)

//通过 “\t” 即按行对数据进行分割
val records = rawData.map(_.split("\t"))

/**
 * 取第一列为类标，其余列作为特征值
 */
val data = records.map{ point =>
    //去除集合中多余的空格
    val firstdata = point.map(_.replaceAll(" ", ""))
    //用空格代替集合中的逗号
    val replaceData=firstdata.map(_.replaceAll(",", " "))
    val temp = replaceData(0).split(" ")
    val label=temp(0).toInt
    val features = temp.slice(1,temp.size-1)
        .map(_.hashCode)
        .map(x => x.toDouble)
    LabeledPoint(label,Vectors.dense(features))
}
```

```
//按照3:2的比例将数据随机分为训练集和测试集
val splits = data.randomSplit(Array(0.8,0.2),seed = 11L)
val training = splits(0).cache()
val test = splits(1)

//训练二元分类的logistic回归模型
val model = new
LogisticRegressionWithLBFGS().setNumClasses(2).run(training)

//预测测试样本的类别
val predictionAndLabels = test.map{
    case LabeledPoint(label,features) =>
        val prediction = model.predict(features)
        (prediction,label)
}

//输出模型在样本上的准确率
val metrics = new
BinaryClassificationMetrics(predictionAndLabels)
val auRoc = metrics.areaUnderROC()
//打印准确率
println("Area under Roc =" + auRoc)
```

### (3) 代码实例：

#### 第三页代码

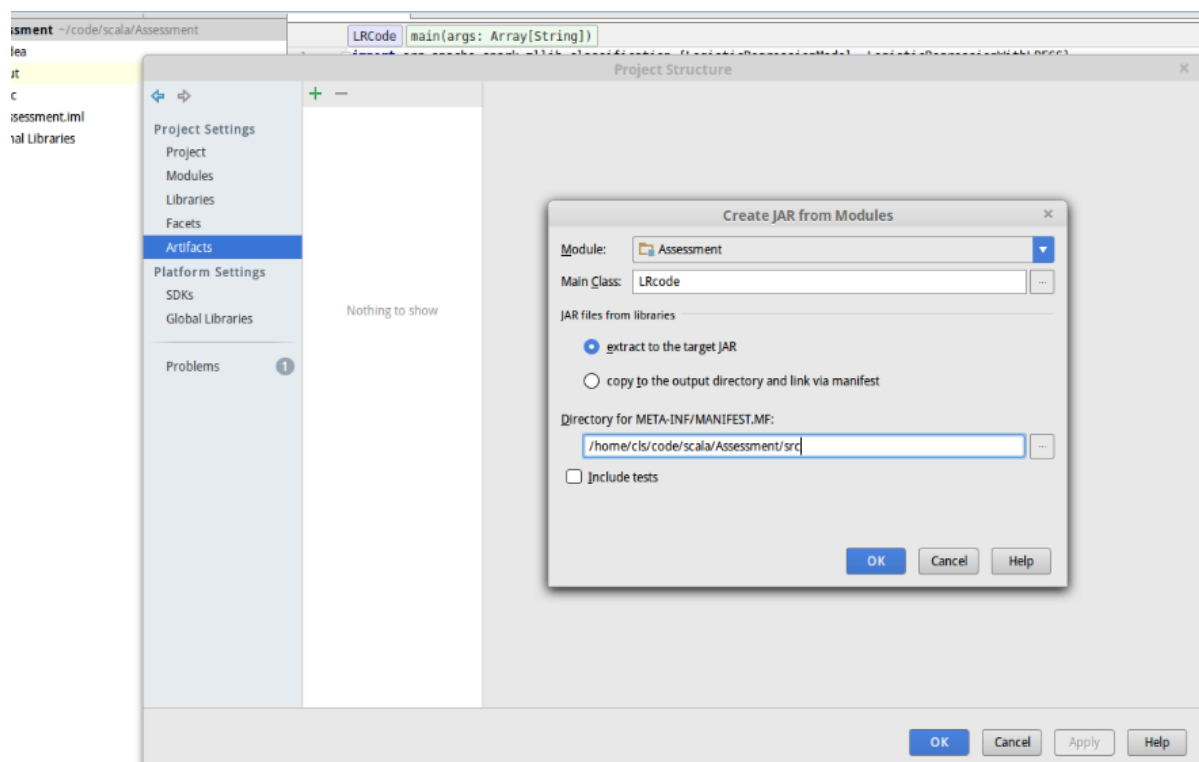
```
//计算统计分类耗时
val endTime = System.currentTimeMillis()
println("endtime:"+endTime)
val timeConsuming = endTime - startTime
println("timeConsuming:"+timeConsuming)

//输出逻辑回归权重最大的前5个特征
val weights = (1 to model.numFeatures) zip model.weights.toArray
println("Top 5 features:")
weights.sortBy(-_._2).take(5).foreach{case(k,w) =>
  println("Feature " + k + " = " + w)
}

//保存训练好模型
val modelPath = "hdfs://master:8020/output/"
model.save(sc, modelPath)
val sameModel = LogisticRegressionModel.load(sc,modelPath)

//关闭程序
sc.stop()
}
}
```

#### (4) 服务器运行:



- 菜单: File->project stucture 。
- 在弹窗最左侧选中Artifacts->左数第二个区域点击"+",选择jar, 然后选择from modules with dependencies, 然后会有配置窗口出现, 配置完成后, 勾选 Build On make (make 项目的时候会输出jar)->保存设置。(如图所示)。
- 然后菜单: Build->make project。
- 最后在项目目录下去找输出的jar包。



## 2.4 实战：个人贷款风险评估

### 5、实验结果

```
16/12/15 08:47:26 INFO optimize.LBFGS: Step Size: 0.02914
16/12/15 08:47:26 INFO optimize.LBFGS: Val and Grad Norm: 0.410850 (rel: 7.87e-08) 0.00364139
16/12/15 08:47:27 INFO optimize.LBFGS: Step Size: 1.000
16/12/15 08:47:27 INFO optimize.LBFGS: Val and Grad Norm: 0.410850 (rel: 2.33e-06) 0.00304491
16/12/15 08:47:27 INFO optimize.LBFGS: Step Size: 1.000
16/12/15 08:47:27 INFO optimize.LBFGS: Val and Grad Norm: 0.410847 (rel: 6.03e-06) 0.00153929
Area under Roc =0.7124486446831049
endtime:1481791655150
timeConsuming:23579
Top 5 features:
Feature 5 = 5.439472541058384E-4
Feature 13 = 5.7454444176084154E-5
Feature 12 = 8.020666775361346E-7
Feature 11 = 1.8109722375846685E-7
Feature 6 = 1.3149771490424918E-9
16/12/15 08:47:35 INFO Configuration.deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
16/12/15 08:47:35 INFO Configuration.deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
16/12/15 08:47:35 INFO Configuration.deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
16/12/15 08:47:35 INFO Configuration.deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
16/12/15 08:47:35 INFO Configuration.deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
16/12/15 08:47:40 INFO hadoop.ParquetFileReader: Initiating action with parallelism: 5
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
16/12/15 08:47:41 INFO mapred.FileInputFormat: Total input paths to process : 1
```

由图可知，该分类模型准确率约为71.2%，耗时为23579毫秒，权重最大的前五个特征为第5、6、11、12、13个特征。

感谢聆听

