**一、Spark的环境搭建与运行**

Spark运行模式

1. 本地单机模式(所有进程在同一个JVM)

2. 集群单机模式(进程在不同节点上运行)

3. 基于mesos

4. 基于YARN(Hadoop 2)

spark shell:

scala -> .bin/spark-shell

python -> .bin/pyspark

SparkContext

**Resilient Distributed Dataset** (RDD) (容错性，一个节点由于非用户原因任务失败，在余下节点上自动重建)

val rddFromTextFile=sc.textFile(“LICENSE”)

* Transformation: 对数据集里所有记录执行某种函数
* Action: 运行计算或聚合，结果返回SparkContext
  + map: 将输入映射为新的输出
    - val intsFromStringsRDD=rddFromTextFile.map(line => line.size)
    - intsFromStringsRDD.count #记录数目
    - intsFromStringsRDD.sum #记录总和
  + Spark转换操作是延后的，在spark上调用一个转换操作并不会立即触发相应的计算。所以如果transformation以后没有调用action，就不触发实际计算。
* RDD缓存
  + 数据缓存在集群内存中: rddFromTextFile.cache
  + 数据可以直接从内存中读出从而加快计算

**Broadcast variable**广播变量 / **Accumulator** 累加器

* 只读变量，可以让工作节点高效访问相同数据的应用场景
* val broadcastAList=sc.broadcast(List(“a” , “b” , “c” , “d”, “e”))
* sc.parallelize(List(“1” , “2” , “3”)).map(x=>broadcastAList.value ++ x).collect
  + collect将整个RDD以scala集合的形式返回驱动程序
  + 返回值：Array(List(a,b,c,d,e,1),List(a,b,c,d,e,2),List(a,b,c,d,e,3))

SBT配置文件build.sbt各行代码之间的空行是必需的

Scala程序在.scala文件里

**二、设计机器学习系统**

营销

个性化（隐式）、推荐（显式）：一对一

目标营销、客户细分：将用户分成不同的组

机器学习模型：

* supervised learning监督学习
  + 使用已标记数据学习
  + e.g. 推荐引擎，回归，分类
* unsupervised learning 无监督学习
  + 不需要标记数据
  + 试图学习或是提取数据背后的结构或从中抽取最重要的特征
  + e.g. 聚类，降维，文本处理

**机器学习系统组成**：

用户行为🡪数据收集🡪数据清理与转换🡪【模型训练🡪模型测试】🡪模型部署与整合🡪反馈到第一步

数据清理与转换：

* 预处理
  + 数据过滤
  + 处理数据缺失、不完整或有缺陷
  + 处理可能的异常、错误和异常值
  + 合并多个数据源
  + 数据汇总
* 转换和特征提取
  + 类别数据编码为对应数值
  + 文本数据提取有用信息
  + 处理图像或音频
  + 数值数据转换为类别数据（例如年龄分段）
  + 特征工程：对现有变量进行组合或转换来生成新特征

模型训练与测试回路：

* 模型选择
  + 对特定任务最优建模方法的选择（有时选择多个模型组合，ensemble method集成学习法）
  + 对特定模型最佳参数的选择
* cross-validation交叉验证
  + 在训练数据集上运行模型，在测试数据集上测试效果

**三、4 从数据中提取有用特征**

特征：用于模型训练的变量 （几乎所有机器学习模型都是与用向量表示的数值特征打交道。）

* 数值特征（年龄）
  + 各个特征的权值在【特征值到输出／目标变量】的映射中扮演重要角色
* 类别特征（性别，职业）
  + 名义(nominal)变量，各个取值之间没有顺序关系
  + 有序(ordinal)变量，取值之间存在顺序关系
  + 常用“k之1”的方法将类别特征用数字表示，一个变量的取值可以用长度为k的二元向量来表示
* 派生特征
  + 对数值数据进行某种概括，并期望他能让模型学习更加容易
  + 数值特征到类别特征的转换：区间特征（年龄，地理位置，时间）
* 文本特征（描述，评论）
  + natural language processing自然语言处理
    - tokenization将文本用空白、标点分开
    - stop words removal 删除the, and, but之类的单词
    - stemming 把词简化为基本的形式
    - vectorization 用向量表示处理好的词，使用稀疏矩阵表示某个词是否出现过
* 正则化特征 normalization
  + 正则化特征：对数据集里的单个特征进行转换，如减去平均值（特征对齐）或正则转换（使平均值为0，标准差为1）
  + 正则化特征向量：对数据中某一行的所有特征进行转换，以让转换后的特征向量的长度标准化。
* 其他特征（图像，音频，位置）

**五、Spark构建分类模型**

常见的分类模型：

* 线性模型：简单，相对容易扩展到非常大的数据集
* 决策树：非线性，训练过程计算量大，较难扩展，但性能很好
* 朴素贝叶斯模型：易训练，高效且并行，可作为很好的模型测试基准

线性模型：

* 对样本的预测结果（目标或者因变量）进行建模，对输入变量（特征或者自变量）应用简单的线性预测函数
* y=f(wTx) #y是目标变量，w是权重向量，x是输入的特征向量
  + (wTx)是关于权重向量w和特征向量x的线性预测器
  + 模型的拟合、训练、优化：给定输入数据的特征向量和相关的目标值，存在一个权重向量能够最好对数据进行拟合，拟合就是最小化模型输出与实际值的误差。目标是找到一个权重向量能够最小化所有训练样本的损失（误差）和。
  + 这个损失是由损失函数计算的
    - 逻辑损失logistic loss 《==》逻辑回归
      * 连接函数 1/(1+exp(-wTx))
      * 损失函数 log(1+exp(-ywTx)) #y为输出值，正类1，负类-1
    - 合页损失 hinge loss 《==》支持向量机 Support Vector Machine
      * SVM不是概率模型，但可以基于模型对正负的估计预测类别
      * y=wTx
      * 损失函数（合页损失）max(0,1-ywTx)

朴素贝叶斯模型(naïve Bayes)：

* 概率模型，通过计算给定数据点在某个类别的概率来进行预测。假定每个特征分配到某个类别的概率是独立分布的（嘉定各个特征之间条件独立）。
* 属于某个类别的概率表示为若干概率乘积的函数：
  + 条件概率（某特征在指定类别的条件下出现的概率）
  + 先验概率（该类别的概率）

决策树：

* 非概率模型，可以表达复杂的非线性模式和特征相互关系，适合应用集成方法（ensemble method），决策树森林是多个决策树的继承
* 决策树的叶子代表值为0或1的分类，树枝代表特征。每个步骤中通过评估特征分裂的信息增益选出分割数据集最优的特征。信息增益通过计算节点不纯度（节点标签不相似或不同质的程度）减去分割后的两个子节点不纯度的加权和。

P92