南京大学 人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

Tencent 腾讯

智能应用建模

# 第七部分：腾讯智能钛机器学习平台和腾讯云

彭浩源

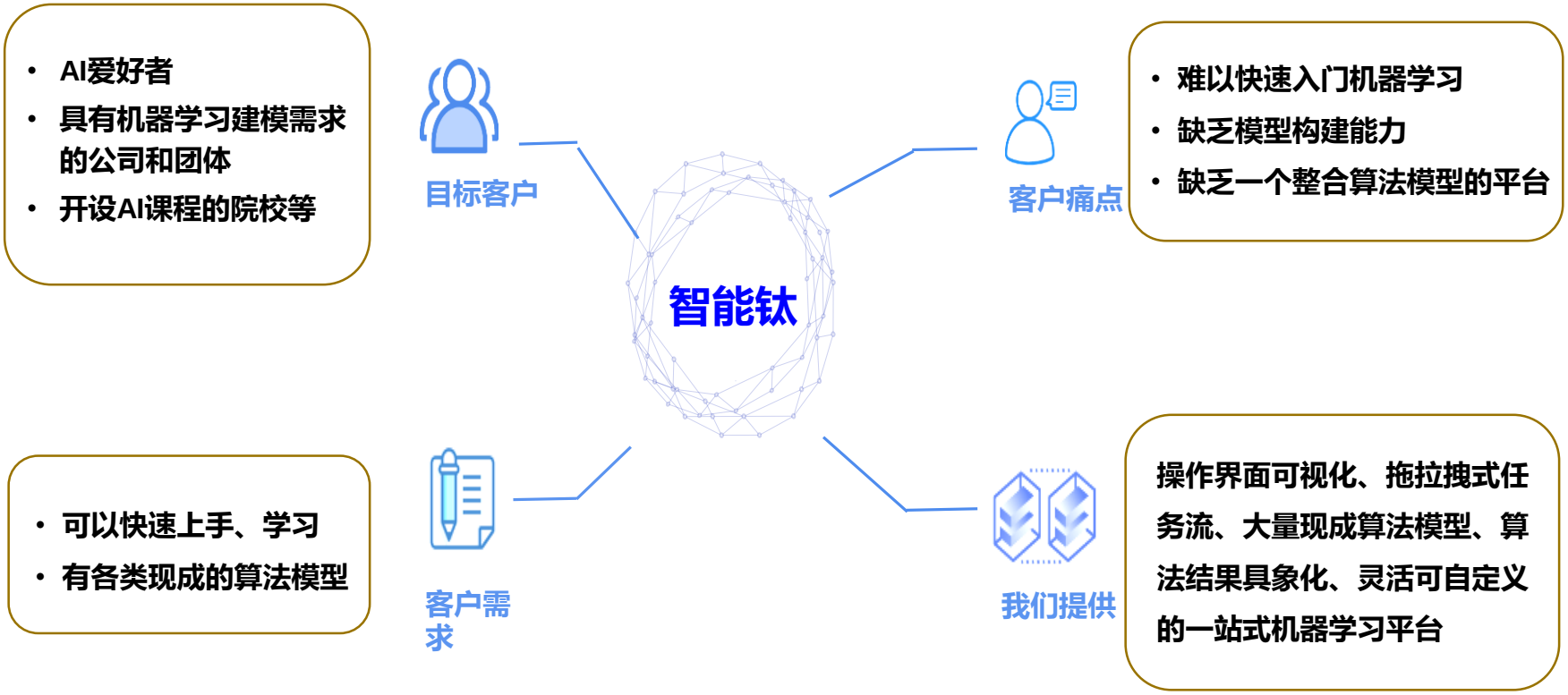
2019年7月5日

# 内容安排

- 智能钛机器学习平台概述
- 智能钛平台内置算法
- 智能钛平台组件和notebook
- 智能钛平台实践

# 智能钛产品概述

- 智能钛是一款内置丰富模型算法，通过拖拽操作就能快速上手的机器学习平台



# 内容安排

- 智能钛机器学习平台概述
- **智能钛平台内置算法**
- 智能钛平台组件和notebook
- 智能钛平台实践

# 数据预处理

# 数据预处理——数据准备

- 数据切分：
  - 按照切分比例将数据集切分成两个集合。
- 数据采样：
  - 对数据集中的数据进行采样，TIONE 提供了多种采样方法，包括按比率采样、按照样本数采样、上采样和下采样四种，其中上采样和下采样主题处理数据不平衡的情况。
- 缺失值填充：
  - 可以指定一列或者多列数据，对数据的缺失值进行填充。对于数值数据，可以填充 0 值、均值、中位数、最小值、最大值。对于类别数据，可以填充指定的值。

# 数据预处理——数据准备

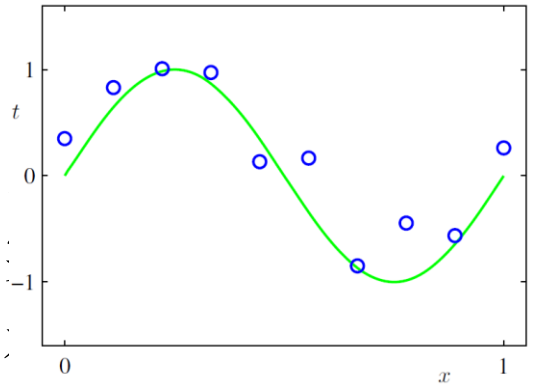
- 生成id列：
  - 在现有的数据集中增加一列，为 ID 列，该列数据不一定有序，但是即互不相同。生成 ID 列有利于数据存库。
- 样本去重：
  - 对数据集中的样本进行去重处理。这里的去重处理是去重整行重复的样本。
- 选择特征列：
  - 选择特征列类似于 SQL 中的 select 功能。对于一个初始的数据集，进行分析建模时，通常只会选择其中的多列进行建模。

# 背景知识

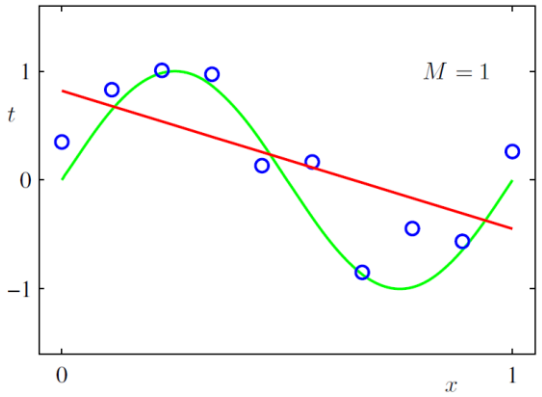
- 训练集，验证集和测试集：
  - 在机器学习领域中一般将数据分为互不重叠的三部分：训练集，验证集和测试集。
  - 训练集用于建立机器学习模型。
  - 验证集用于机器学习模型的调优。
  - 在测试集上机器学习模型的效果进行最终评价。
- 为什么需要这样划分？
  - 由于“过拟合”现象的存在，在训练集上误差更小的机器学习模型未必是一个更好的模型。

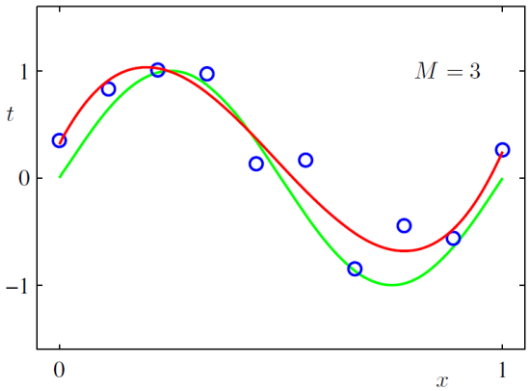
# 背景知识

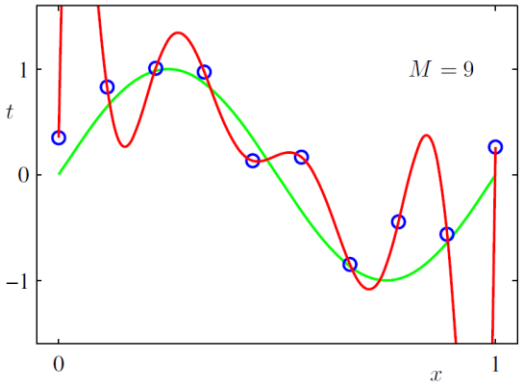
- 过拟合现象：




  - 机器学习模型学 噪声，从而导致学习结果和数据的真 现象。
  - 过拟合的表现： 的效果优异，但在验证和测试数据上模型的效果较差。



# 背景知识

- 参数和超参数
  - 参数(parameter)：机器学习模型根据训练集计算或训练得到的值，如上页例子中使用三次函数$f(x) = ax^3 + bx^2 + cx + d$拟合绿色曲线时，模型根据训练集中的各个数据点，计算出来的$a, b, c, d$四个变量的取值。
  - 超参数(hyperparameter)：用来确定采用什么机器学习模型的参数，如采用三次函数拟合绿色曲线或采用九次函数拟合绿色曲线时，次数就是模型的超参数，3或9是这个超参数的两个不同取值。
- 模型的“调优”一般指选择超参数，使模型在验证集上表现更好。

# 特征工程

# 什么是特征？

- 在机器学习中，特征指的是用来描述样本的属性
- 例如在评估一个人的信用状况时，我们需要考虑一个人的年龄，收入，工作等属性，此时年龄，收入，工作等属性就是评估信用状况这一任务所需要的特征。
- 机器学习模型构造一个从特征到输出结果的映射

# 特征转换

- 离散化：
  - 将连续的特征数据变为离散的类别。
  - 分为等频离散化和等值离散化两种。
- 归一化：
  - 不同的特征的取值范围不同，为了消除这一影响，可以对输入特征进行归一化：
  - 分为最大最小归一化和标准归一化。
- 二值化：
  - 二值化是一个将数值特征转换为二值特征的处理过程。它通过一个阈值控制划分。 值大于阈值的特征二值化为1，否则二值化为0。

# 特征转换

- One-hot编码：
  - 将离散型特征的每一种取值都看成一种状态，若你的这一特征中有N个不相同的取值，那么我们就可以将该特征抽象成N种不同的状态，one-hot 编码保证了每一个取值只会使得一种状态处于“激活态”，也就是说这N种状态中只有一个状态位值为1，其他状态位都是0。

# 特征选择

- 卡方选择：
  - 计算每个特征对应的卡方统计量，并按照从大到小的顺序进行排序，然后保留排名前若干位的特征。
  - 对于某个特征，假设预测目标与该特征相互独立，然后计算实际值与该假设下理论值的偏差程度（卡方统计量）。如果偏差较小，那么上述假设更有可能成立；如果偏差较大，那就否定上述假设，认为预测目标是与该特征相关的。
  - 因此，卡方统计量越大，说明该特征与预测目标越相关，因此该特征的重要性越高。

# 特征选择

- 基于方差的特征选择：
  - 在机器学习中，低方差的特征往往对最终的结果影响较小。通过基于方差的特征选择，过滤掉低方差的特征，有利于后续的机器学习建模。

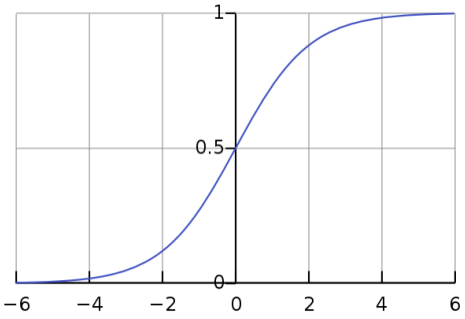
- 基于信息的特征选择：
  - 基于信息的特征选择，平台上该模块共包括4种算法：信息增益（Information Gain）、基尼系数（Gini）、信息增益率（Information Gain Ratio）以及对称不确定性(Symmetry Uncertainly)。通过这四个值来确定最重要的特征。

# 特征选择

- 基于降维的特征选择：
  - PCA：主成分分析。主成分分析是最常用的一种降维方法。通过降维的方式选择最主要的特征用于机器学习建模。

# 传统分类算法

# Logistic回归




- 一种常用的二分类算法，输入特征为连续值。

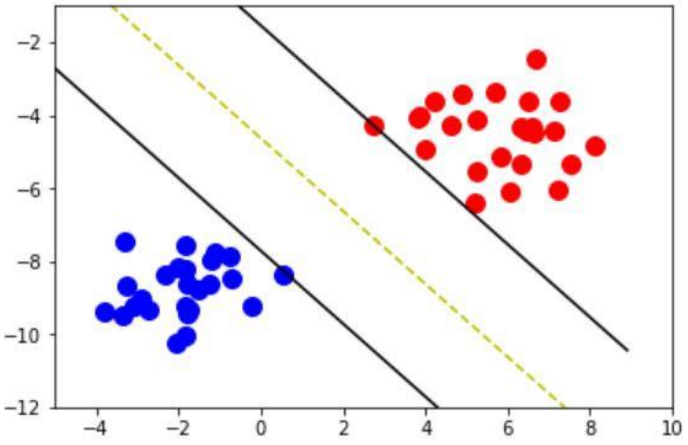
$$f(x) = \frac{1}{1 + e^{-x}}$$

- 模型参数：一个代表权重的向量θ
- 训练目标：直接学习θ的取值，即每个输入特征各自的权重，使得一类样本的特征向量与θ的内积尽可能大，另一类样本与θ的内积尽可能小。

- 为了有效训练，将向量内积通过一个sigmoid函数，使得一类样本的输出尽可能接近1，另一类样本的输出尽可能接近0.

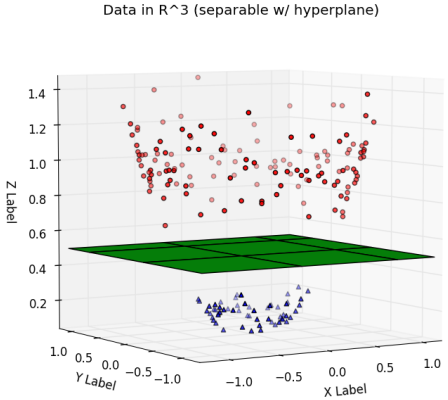
# 支持向量机（SVM）

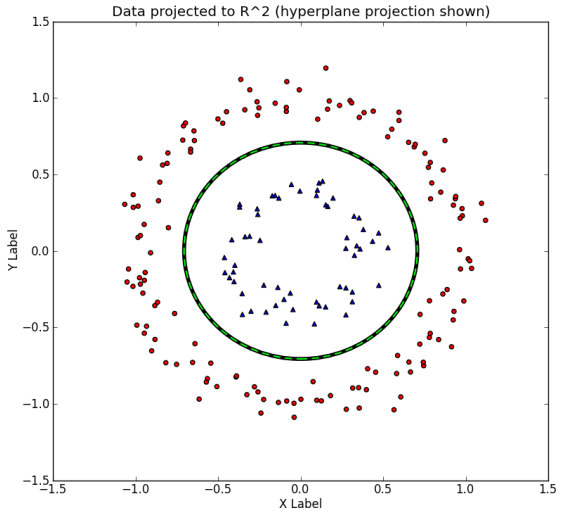
- 一种常用的二分类算法，输入特征为连续值。
- 寻找最优的分割超平面，使得该超平面离最近的样本点的距离尽可能远。



# 支持向量机（SVM）

- 对于线性不可分的情况，SVM会使用核函数进行升维，使其变成线性可分的情况。

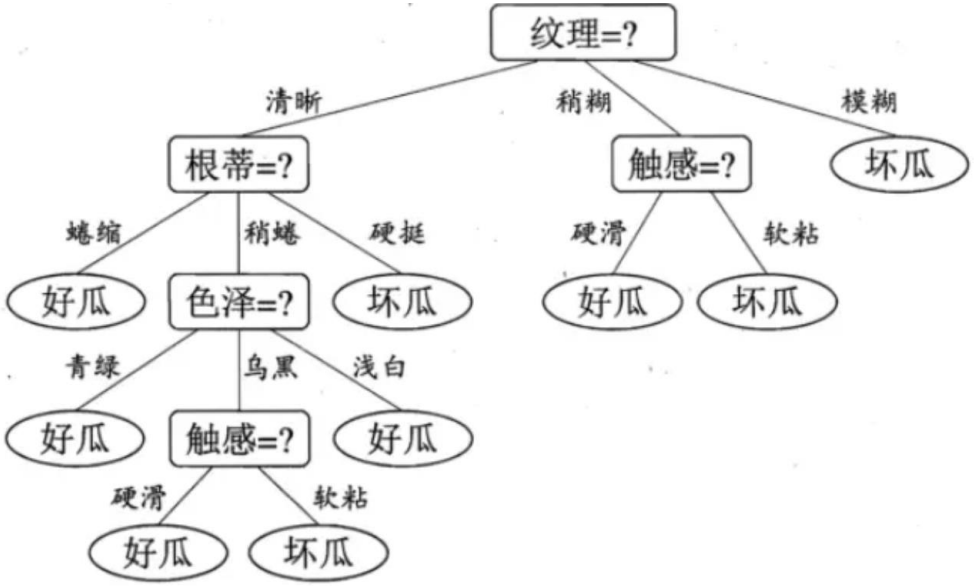



- 多分类情况可以转换成训练多个one vs rest的二分类器的问题。

# 决策树

- 常用多分类模型，输入特征一般为离散值。
- 一棵决策树可以看成是一系列if-then规则的集合，结果具备较强的解释性。这些规则是从训练数据中学习得到的。



# 决策树

- 构造决策树的方法：先构造包含所有训练样本的根节点，然后选择一个最优的特征，根据该特征的取值将样本分裂为若干个子节点。对各个子节点递归执行上述过程，直到节点中样本均为同一类别/无法划分/树达到最大深度为止。
- 希望划分后，每个子节点内部的样本越纯越好。
- 衡量指标：
  - 信息增益：样本分布越均匀，熵越大。划分后，熵下降的程度称为信息增益。
  - Gini系数：随机取两个样本，它们不属于同一个类别的概率

# 随机森林

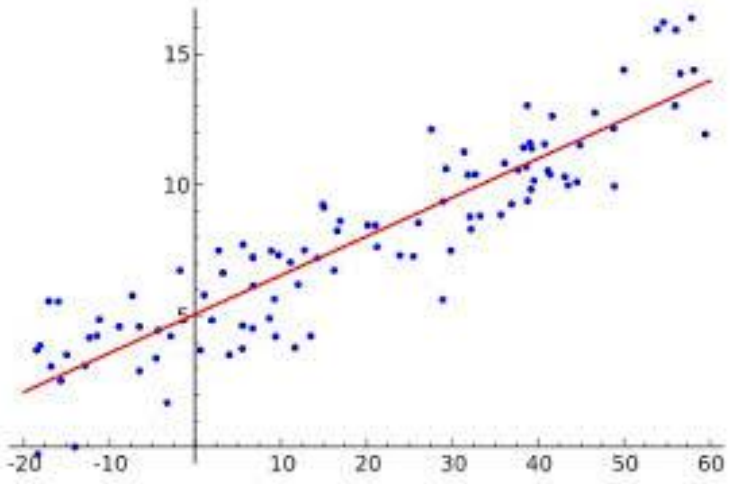
- 为了减少扰动，构造多棵决策树并用投票机制获取分类结果的算法。
- 输入特征为离散值。

- 设训练数据的样本总数为m。对于每一棵树，都从训练数据中有放回地采样出m个样本用于生成这棵树。同时每一棵树只随机选取样本特征中的一部分用于划分节点。
- 新的样本进入随机森林后，森林中的各决策树通过投票机制确定样本的分类。

# 传统回归算法

# 线性回归

- 使用输入特征的线性变换预测目标值的算法。即 $f(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b$
- 算法的最优解是最小化训练数据的平方误差。



# 决策树回归

- 将训练数据所在的空间划分为若干个子空间，每个子空间的输出值为该空间内所有训练数据的平均值。
- 根节点包含全部训练样本。对于每个节点，遍历各个可能的特征和划分点，找到划分后平方误差最小的特征和切分点，并据此将样本划分为两部分。
- 递归地对子节点进行划分，直到无法划分/达到最大深度为止。

# 随机森林回归

- 构造多个决策树回归模型，并将它们的均值作为输出的算法。

- 设训练数据的样本总数为m。对于每一棵树，都从训练数据中有放回地采样出m个样本用于生成这棵树。同时每一棵树只随机选取样本特征中的一部分用于划分节点。
- 新的样本进入随机森林后，森林中的各决策树输出结果的平均值作为随机森林的输出。

# 梯度提升树回归

- 梯度提升（gradient boosting）属于Boosting算法的一种，也可以说是Boosting算法的一种改进，它与传统的Boosting有着很大的区别，它的每一次计算都是为了减少上一次的残差(residual)，而为了减少这些残差，可以在残差减少的梯度(Gradient)方向上建立一个新模型。所以说，在Gradient Boosting中，每个新模型的建立是为了使得先前模型残差往梯度方向减少， 与传统的Boosting算法对正确、错误的样本进行加权有着极大的区别。

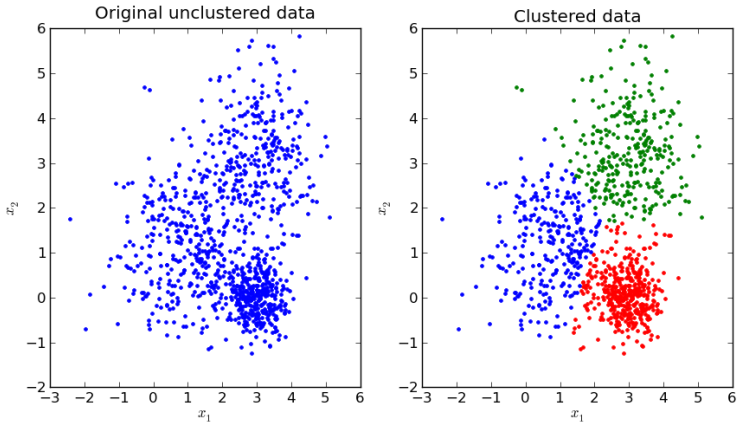
- 梯度提升算法的核心在于，每棵树是从先前所有树的残差中来学习。利用的是当前模型中损失函数的负梯度值作为提升树算法中的残差的近似值，进而拟合一棵回归树。

# 传统聚类算法

# K-MEANS算法

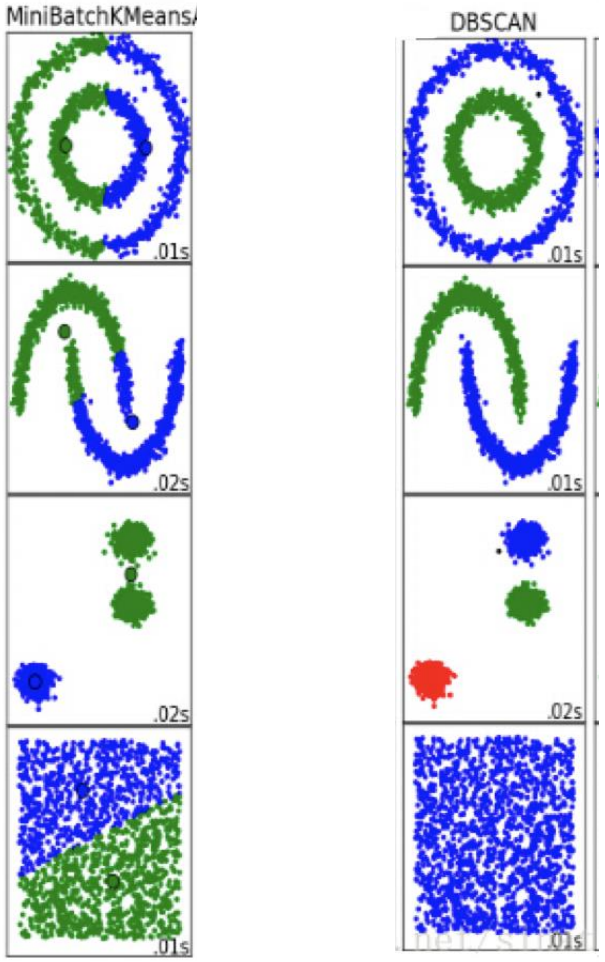
- 给定一个没有类别标签的样本的集合，根据样本间的距离将样本划分为若干类。
- K-MEANS需要指定超参数K，即划分的类别数

- **K-MEANS执行流程：**
  1. **随机选取K个样本作为各类的中心点。**
  2. **对每一个其它样本，计算它到各个中心点的距离。将该样本归为样本离中心点距离最近的一类。**
  3. **计算每一类样本的均值，作为新的中心点。**
  4. **重复1-3步骤，直到在1-3步骤中，没有任何一个样本点的类别发生了变化。**



# DBSCAN算法

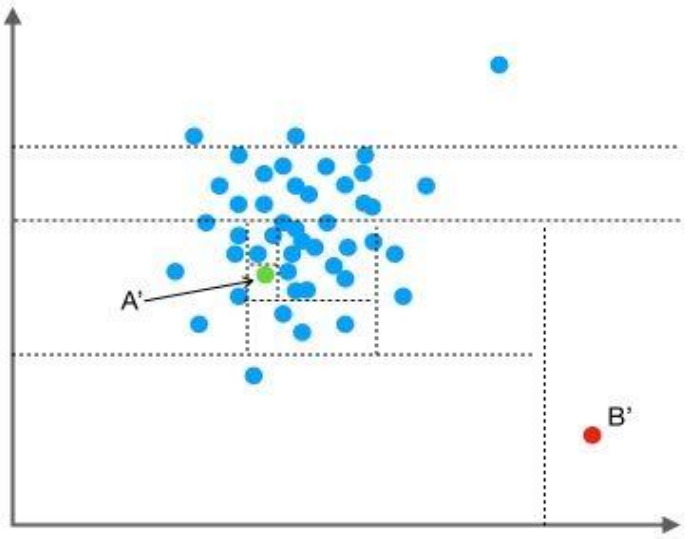
- DBSCAN算法是一种基于密度的聚类算法。该算法可将具有足够高密度的区域划分为簇，并在具有噪声的数据中发现任意形状的簇。



# 异常检测算法

# IsolationForest算法

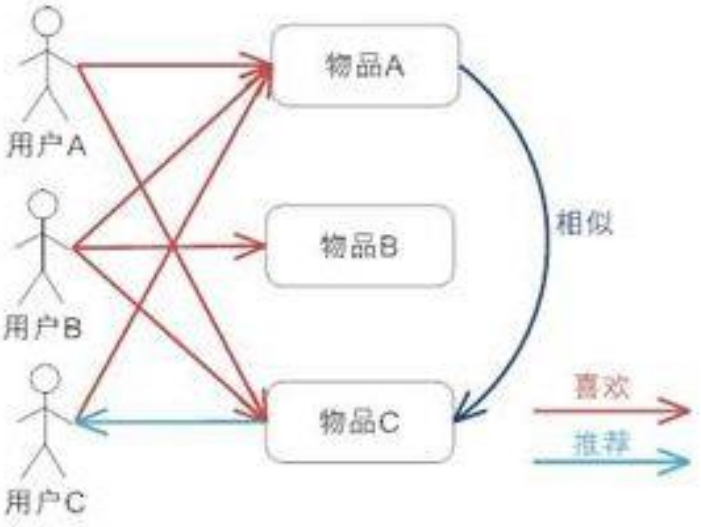
- Isolation Forest 是一种基于孤立森林的异常点检测算法，该算法首先构建 n 棵树，每棵树都从原始数据中有放回的采样 m 个样本进行训练，每棵树在训练的时候都完全采用了随机选择特征以及特征分裂点的方式，然后再将每棵树的训练结果进行汇总就可以得到每个样本成为异常点的概率（0到 1 之间的浮点值），该值越大越有可能是异常点。



# 推荐算法

# Item-CF和User-CF算法

- Item-CF算法：
  - 根据所有用户对物品的评价，发现物品直接的相似度，然后根据用户的历史偏好信息，将类似的物品推荐给用户。
  - 如下图，用户A喜欢物品A和物品C；用户B喜欢物品A、B、C；用户C喜欢物品A。从这些用户的历史喜好中，可以认为物品A和物品C比较类似，基于这个判断，用户C也可能喜欢物品C。



# 其它算法

- 时间序列算法:
  - 时序特征生成：均值，方差，偏度，峰度，极差等
  - 自相关函数：ACF，PACF
  - 差分，平稳性检验
  - ARIMA，EWMA，GARCH，HoltWinter
- 图算法:
  - 社区发现
  - 节点重要性
  - 链路预测
  - 图表示学习

# 深度学习算法

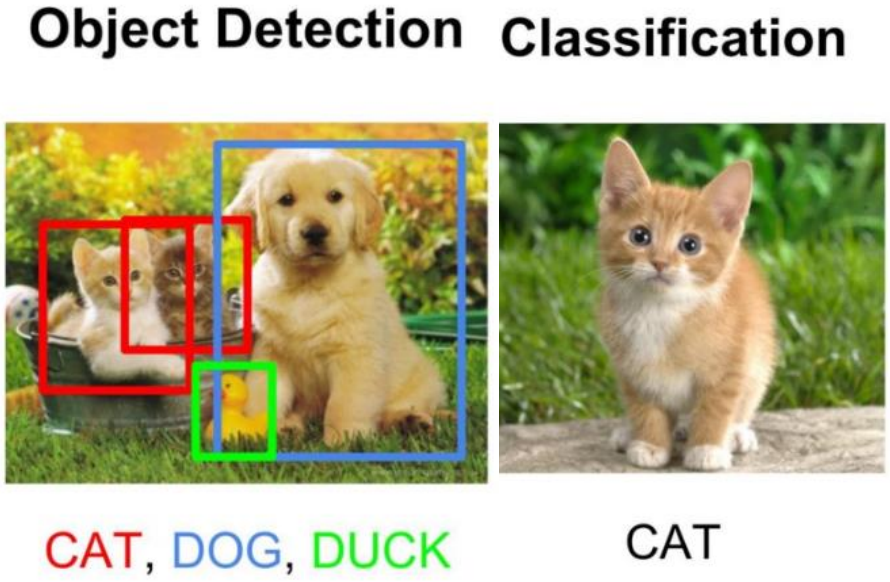
# 图片分类算法

- AlexNet
- VGG
- ResNet
- Inception
- MobileNet
- ShuffleNet

# 目标检测算法

- 相比于图片分类，目标检测算法不仅预测图片中的物体的类别，还能预测物体在图片中的位置。

- FasterRCNN
- SSD
- RFCN



# 文本分类算法

- 算法的输入是一个句子或者一段文本，输出是输入对应的类别，如情感是否正面，用户意图类别等。
- 平台集成了四种不同的文本分类算法，它们用不同的方式构建输入句子的向量表示，然后用全连接层进行分类：
  - FastText
  - TextCNN
  - LSTM
  - BERT

# 序列标注算法

- 序列标注算法对输入句子中的每个词语进行标注，可以进行词性标注，命名实体识别等任务。
- 平台集成了BiLSTM-CRF算法：
  - 用双向LSTM提取句子中各个词语的向量表示，并计算每个词语属于各个标签的概率，最后用CRF建模标签之间的转移概率，最终得到整体概率最大的标签序列。

| 迈向 | 充满 | 希望 | 的 | 新 | 世纪 |
|---|---|---|---|---|---|
| V | V | N | U | A | N |

# 其它算法和工具

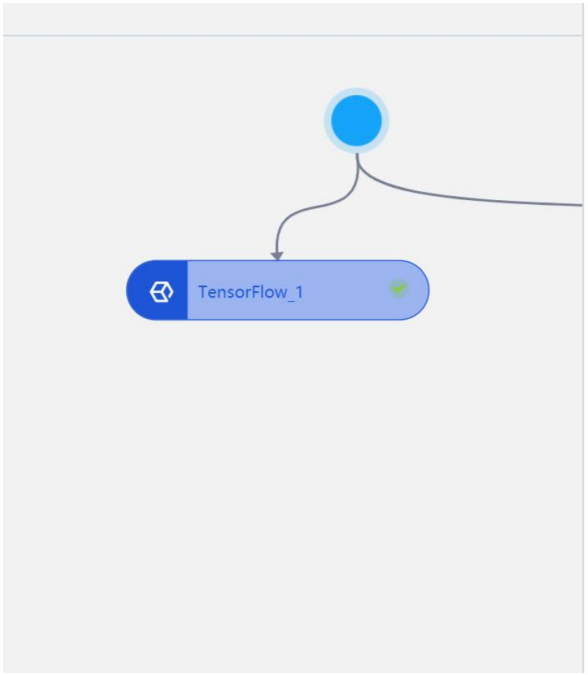
- 词向量训练算法：
  - Word2Vec
  - GloVe
- 分词工具
- 关键词提取工具
- 文本摘要工具
- 词频统计工具

# 内容安排

- 智能钛机器学习平台概述
- 智能钛平台内置算法
- **智能钛平台组件和notebook**
- 智能钛平台实践

# 组件功能

- 组件功能使得用户可以上传自己的程序代码，并在智能钛平台上运行。

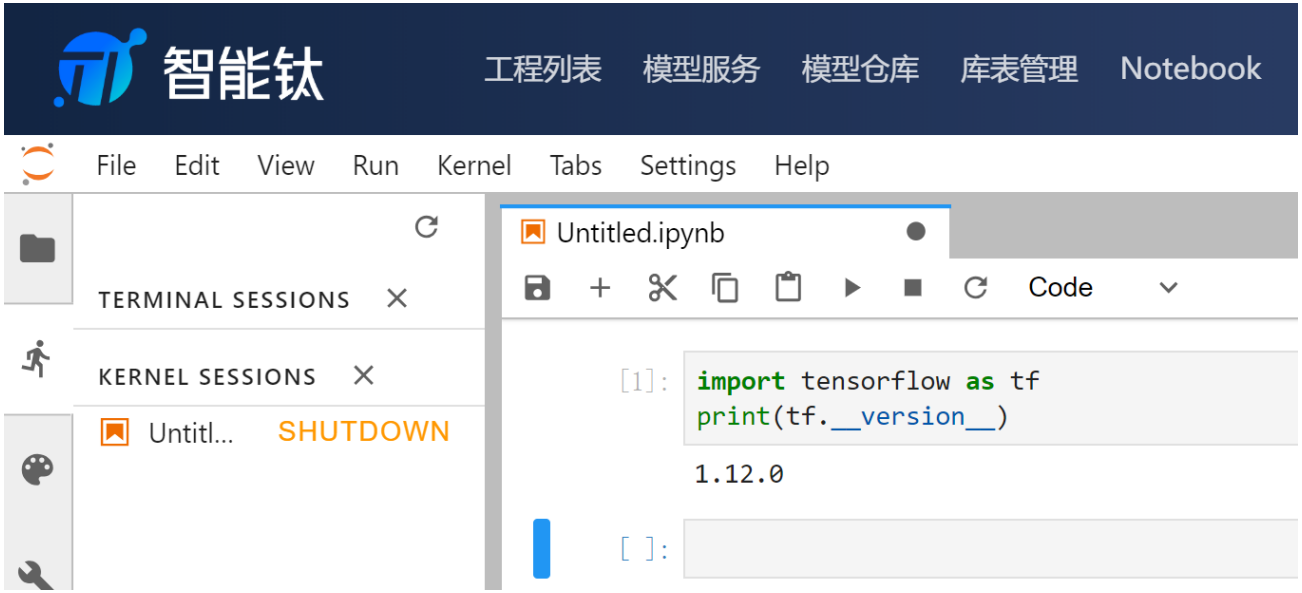


# 依赖包文件

- 如程序入口为main.py，需要调用模块a.py的foo()函数和模块b.py的bar()函数
  - 将a.py，b.py和__init__.py放在同一个目录下，并压缩成一个zip包。
  - 新建一个组件，依赖包文件为上一步创建的zip包。不妨设zip包文件名为dependencies.zip
  - 在main.py中执行from dependencies import a, b，即可调用a.foo()和b.bar()

# Notebook功能

- 智能钛平台提供了Notebook功能，用户可以在智能钛的界面上使用Notebook进行交互式编程。



# 内容安排

- 智能钛机器学习平台概述
- 智能钛平台内置算法
- 智能钛平台组件和notebook
- **智能钛平台实践**

# 组件功能实践

- 新建一个tensorflow组件，上传自己的代码，体验智能钛平台的组件功能，包括程序脚本，依赖包和启动参数功能。

# Notebook实践

- 新建一个自己的Notebook，查看tensorflow版本，并运行一个tensorflow demo程序。

# 内置算法实践

- 参考【典型工作流】中的【花朵图片分类】工作流及其文档，新建一个自己的图片分类工作流，完成模型的训练，在线部署和在线预测。
- 文档链接：https://tio.cloud.tencent.com/gitbook/doc/tione/%E6%9C%80%E4%BD%B3%E5%AE%9E%E8%B7%B5/%E8%8A%B1%E6%9C%B5%E5%88%86%E7%B1%BB.html
- 文档也可以在右上角——帮助文档——最佳实践中找到