《人工智能-总览、应用与前沿》

主讲老师：高彦杰

课程说明

- ✓ 内容简介：
  - ✓ 课程概览，技术栈，前沿技术趋势，知识体系
  - ✓ 机器学习流程，KNN算法
- ✓ 企业级应用领域：
  - ✓ 互联网，安防，金融，运营商，医疗，工业，智能制造等

课程推荐环境

```
# 例如 pip install numpy==1.13.3 安装numpy库
matplotlib == 1.5.3
numpy == 1.13.3
pandas == 0.18.1
scikit_learn == 0.19.1
```

课程大纲

1. 前置，课程概览(★★★)
2. 人工智能简介，过去现在与未来(★)
3. 人工智能应用场景(★★)
4. 人工智能技术概览(★★)
5. 机器学习算法(★★★)
6. 主流机器学习库和计算框架介绍(★★)
7. 如何进行机器学习(★★★)

前置与学习方法

课程概览

1）人工智能基础

2）机器学习算法

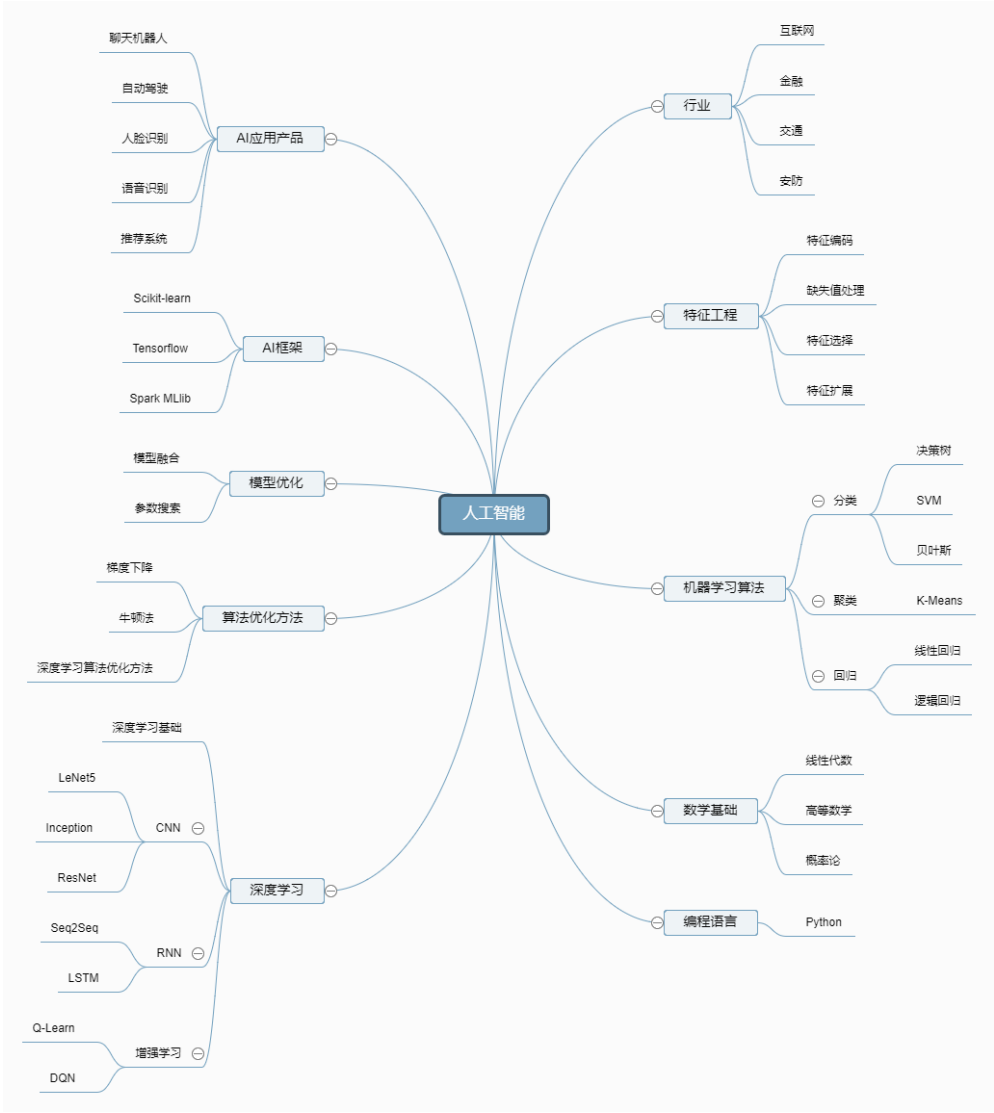
3）深度学习

4）人工智能大数据框架应用

5）企业级人工智能项目实战

6）毕设答辩与笔面试

知识点覆盖


人工智能
AI应用产品
聊天机器人
自动驾驶
人脸识别
语音识别
推荐系统
AI框架
Scikit-learn
Tensorflow
Spark MLlib
模型优化
模型融合
参数搜索
算法优化方法
梯度下降
牛顿法
深度学习算法优化方法
深度学习
深度学习基础
CNN
LeNet5
Inception
ResNet
RNN
Seq2Seq
LSTM
增强学习
Q-Learn
DQN
行业
互联网
金融
交通
安防
特征工程
特征编码
缺失值处理
特征选择
特征扩展
机器学习算法
分类
决策树
SVM
贝叶斯
聚类
K-Means
回归
线性回归
逻辑回归
数学基础
线性代数
高等数学
概率论
编程语言
Python

课程概览

1）人工智能基础

人工智能简介

数学与数学分析基础

特征工程

课程概览

2）机器学习算法

决策树与随机森林

分类算法

回归算法

聚类算法

课程概览

3）深度学习

深度学习-CNN

深度学习-RNN

课程概览

4）人工智能框架，大数据与应用实践

深度学习TensorFlow

大数据与机器学习-Spark MLlib

翻转课堂

课程概览

5）企业级人工智能项目实战

语音识别

端到端自动驾驶

人脸识别

对话机器人

物体检测

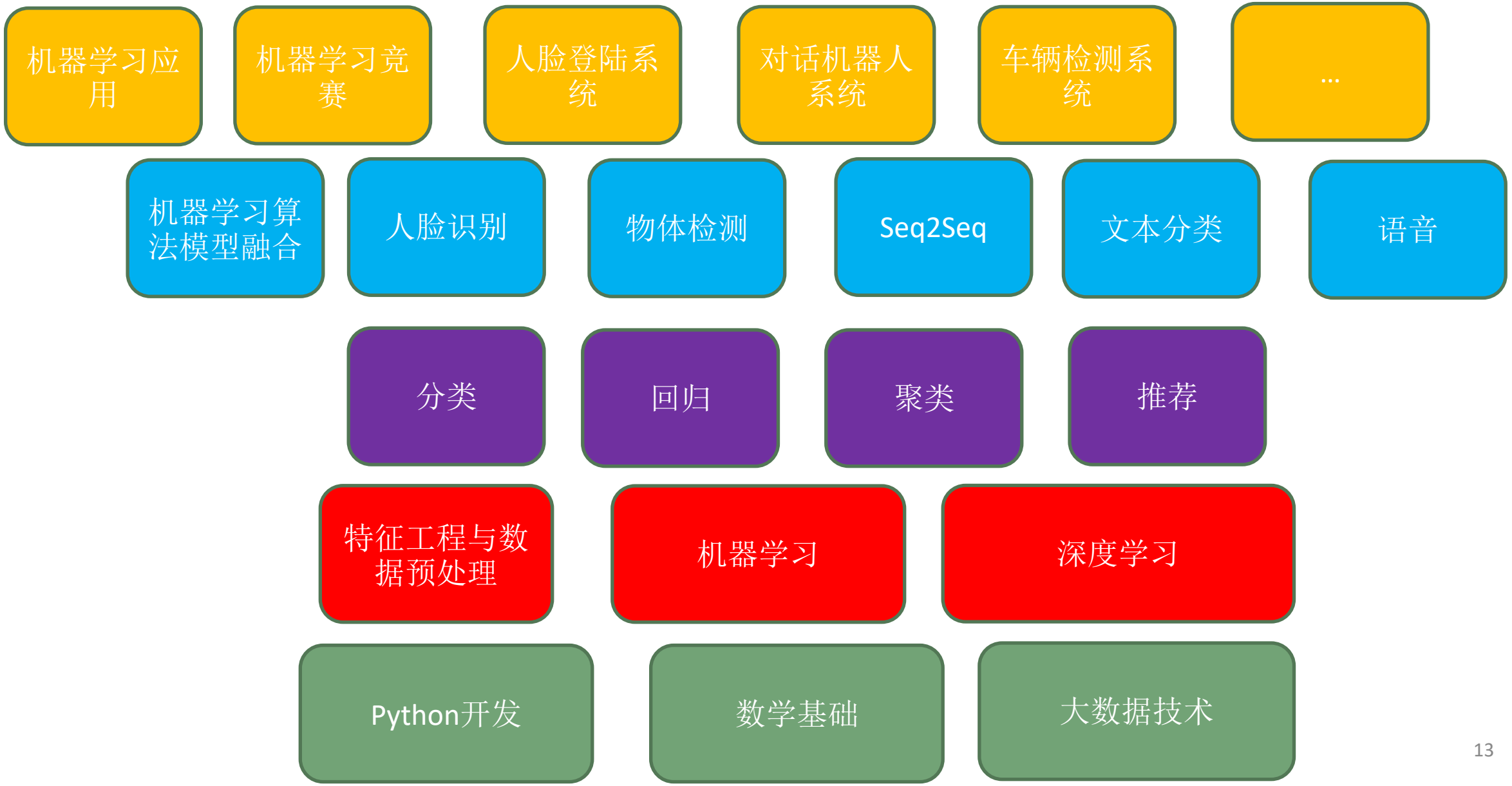
复杂深度学习模型设计

深度学习算法底层实现

人工智能工程化实践

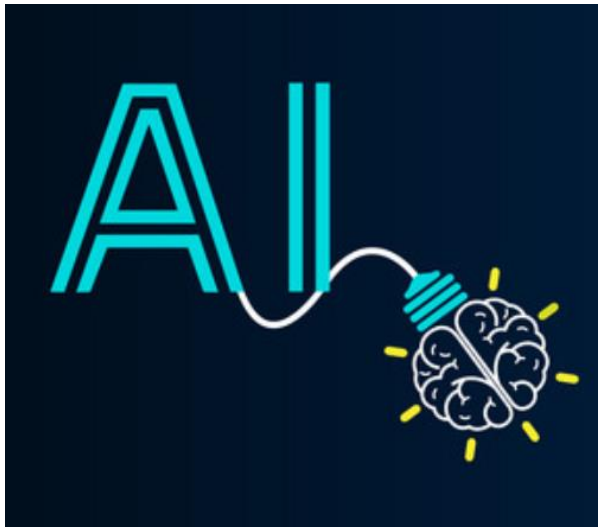
6）毕设答辩与笔面试题型流程分析

知识体系


机器学习应用
机器学习竞赛
人脸登陆系统
对话机器人系统
车辆检测系统
...
机器学习算法模型融合
人脸识别
物体检测
Seq2Seq
文本分类
语音
分类
回归
聚类
推荐
特征工程与数据预处理
机器学习
深度学习
Python开发
数学基础
大数据技术

学习方法

- 大纲为重点知识主线与脉络，形成知识体系
- 有余力情况下深度广度方向拓展学习
- 动手练习学习，加深理解
- 组内多交流与知识共享
- 打好通用技术基础，后期毕设和未来工作中在一个主要技术方向多沉淀
- 师傅领进门，成艺在自身（平时作业和毕设实践一定自我投入精力，只听课上讲解容易遗忘）
- 只要功夫深铁杵磨成针（量变产生质变）



人工智能简介

人工智能的加速发展


家电馆
0元安装
PLUS专享
风扇
冰箱
领券减1000
家电馆
HOME APPLIANCES
最高砍价1000元
清凉空调
PLUS会员用券减150
7月厨热季
大牌价到
海尔自清洁
三年免费换新机
电视家影
品牌大狂欢
家电可砍价
最高砍1000元
来试试啊
个护狂欢购
领券千元红包
立即抢购
品质国货
下单送好礼
强烈推荐
Haier
LG
Skyworth
SIEMENS
能率


Cortana

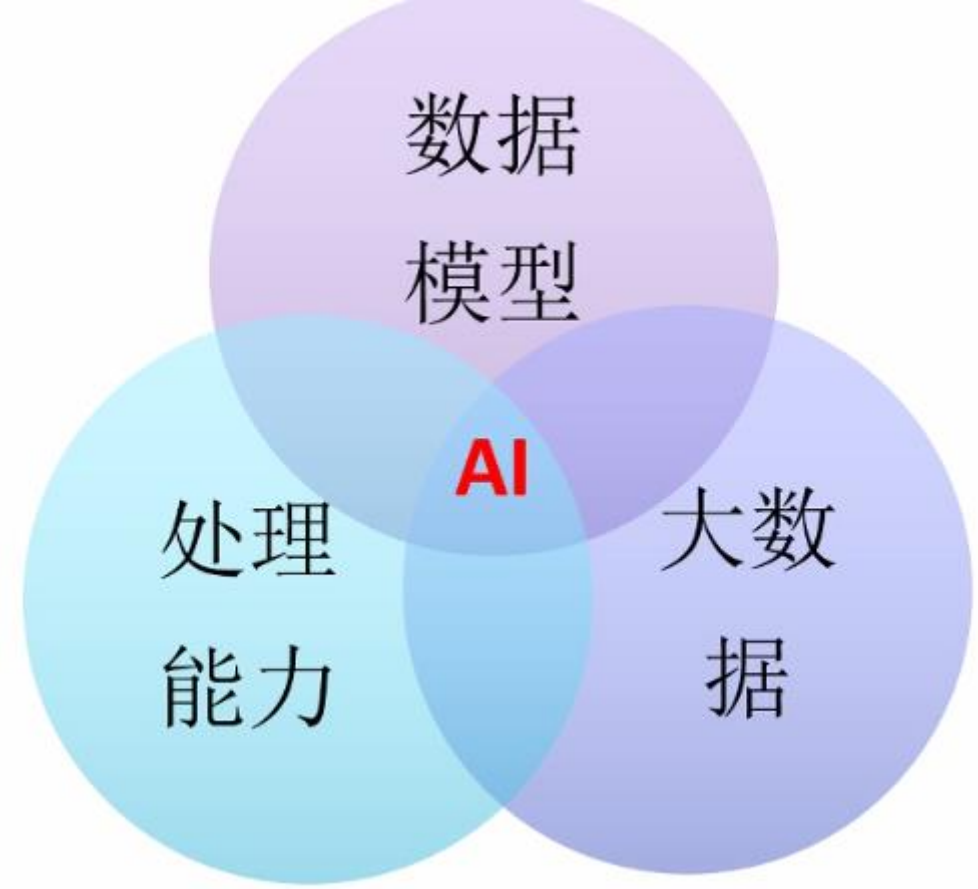

800
信用极好


LEE SEDOL
00:02:13
Google DeepMind
Challenge Match
ALPHAGO
00:27:20

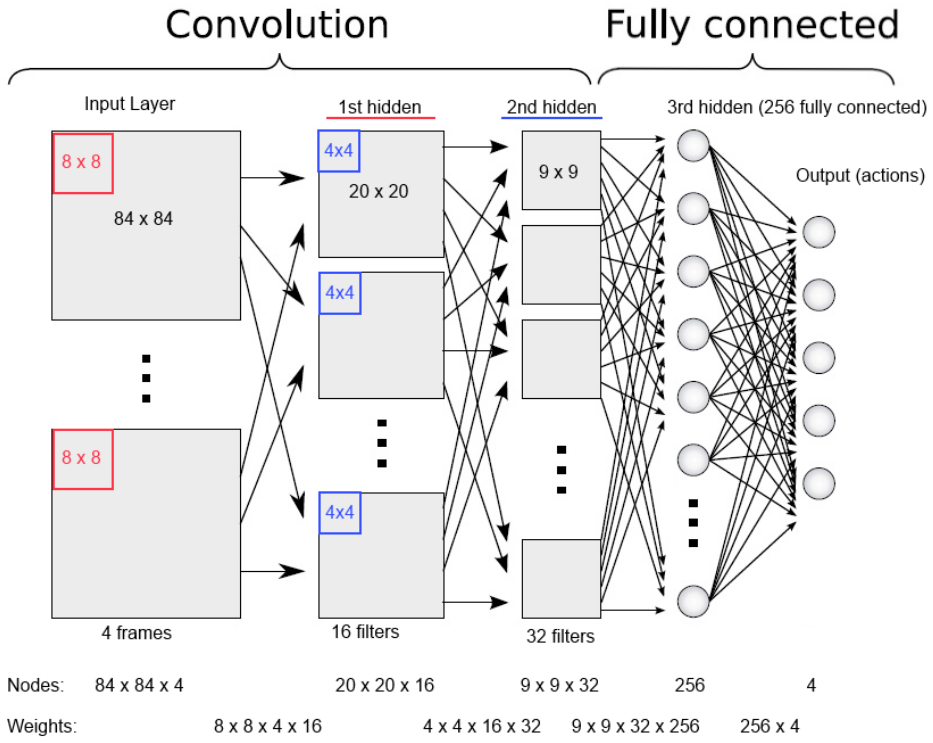

姓名 代用名
性别 男 民族 汉
出生 2013 年 05 月 06 日
住址 湖南省长沙市开福区巡道街幸福小区居民组
公民身份证号码 430512198908131367


小冰
她的主页
微博
个人资料
相册

人工智能核心点



大数据促进人工智能产生突破进展



大数据和深度学习使得人工智能取得突破进展

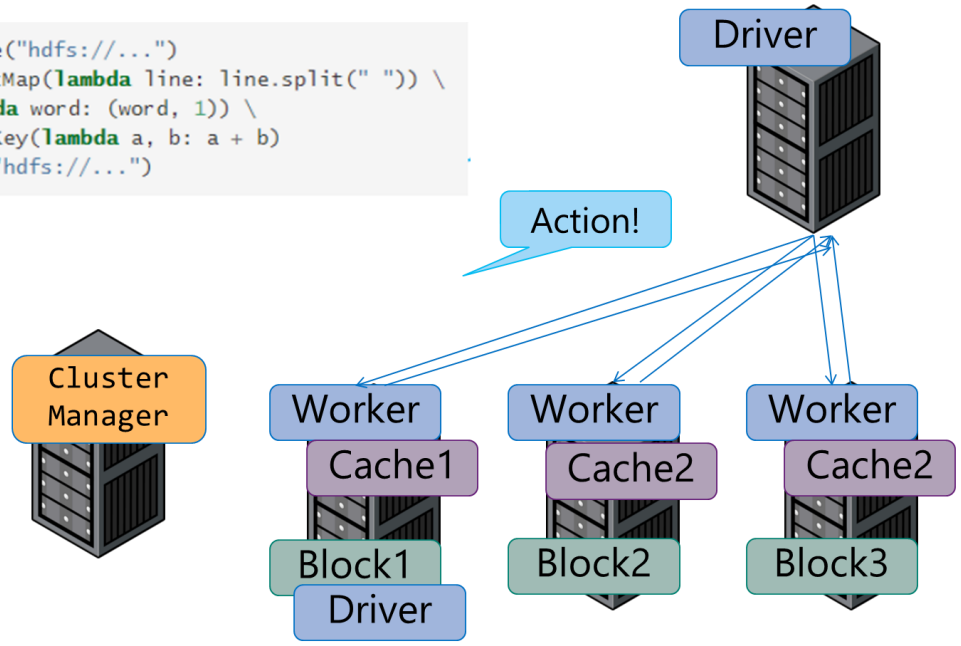
计算能力的提升加速人工智能的发展

Scale Up: GPU, ASIC

Scale Out: Distributed Computing

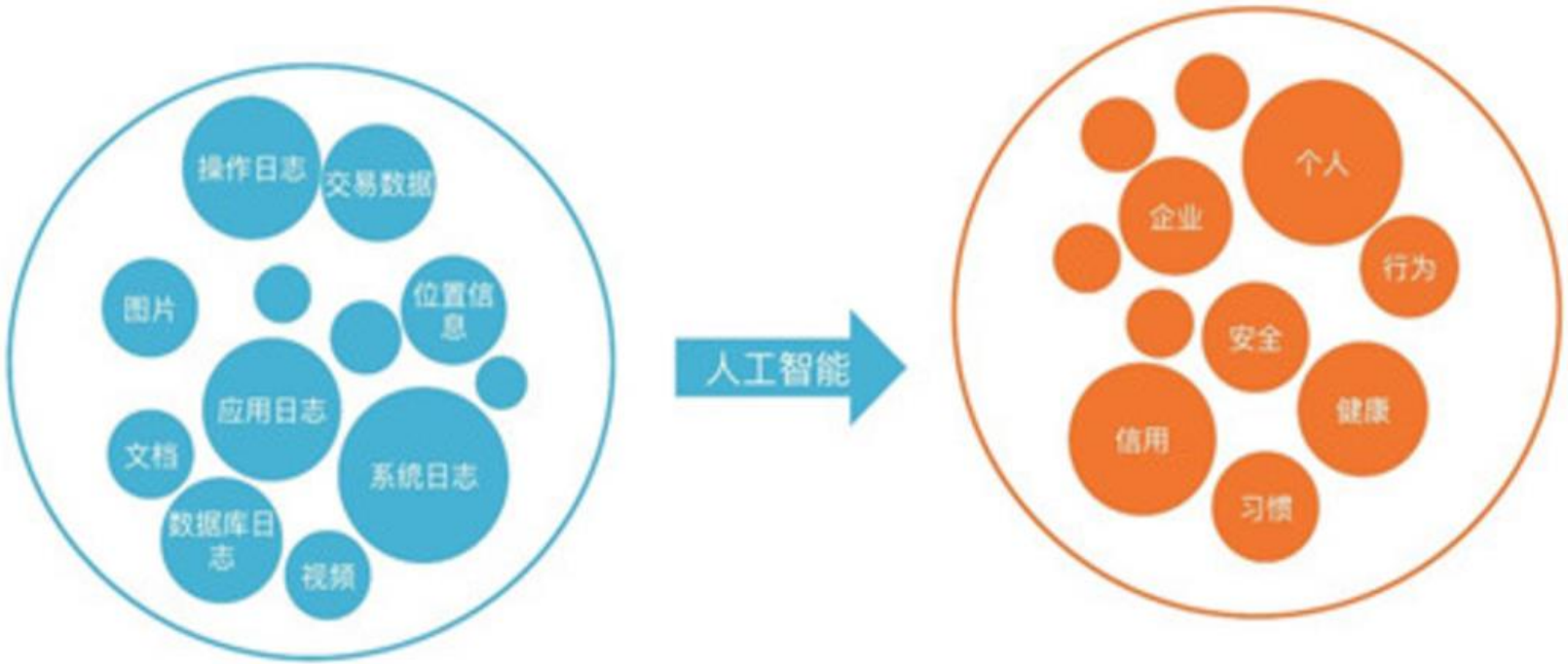
Spark架构

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```



单机数据分析向分布式计算发展

大数据的理解与分析需要AI



人工智能可学习大量大数据，高效分析挖掘出数据价值。

AI 历史

三次浪潮

- ✓ 1956~1976 专家系统
- ✓ 1976~2006 人工神经网络，深度学习尚未突破
- ✓ 2006~至今 基于互联网大数据的深度学习

现状

- ✓ 应用层：互联网，安防，医疗，金融，运营商
- ✓ 接口层：TensorFlow，Scikit-Learn，Spark MLlib等框架接口
- ✓ 算法层：深度学习发展迅速，一些关键场景其他AI算法仍有用武之地
- ✓ 框架层：TensorFlow等框架
- ✓ 基础设施层：CPU与GPU集群大范围使用，云平台蚕食传统Data Center市场

趋势

- ✓ 应用层：深入各个领域
- ✓ 接口层：Workflow，Model Sharing
- ✓ 算法层：AutoML，算法、系统发展
- ✓ 框架层：框架中间层编译优化，开放协议ONNX
- ✓ 基础设施层：GPU集群加速及管理，ASIC，FPGA

什么时候用AI？

- 利用数据来解决简单规则无法或者难以解决的问题，它被广泛应用在了搜索引擎、无人驾驶、机器翻译、医疗诊断、人脸识别、数据匹配、信用评级等任务中。
- 我们无法直接编程解决这些问题，但我们能够使用配合数据编程来解决。

如果给我们的机器学习系统提供足够多猫和狗的图片，我们可以"编写"一个喵星人辨别器：

喵

喵

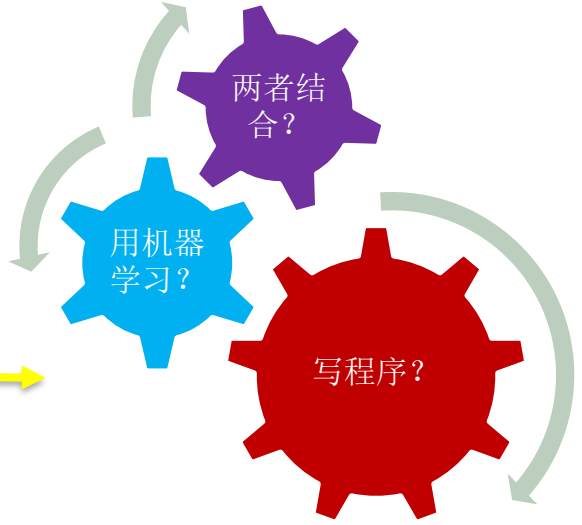
汪

汪

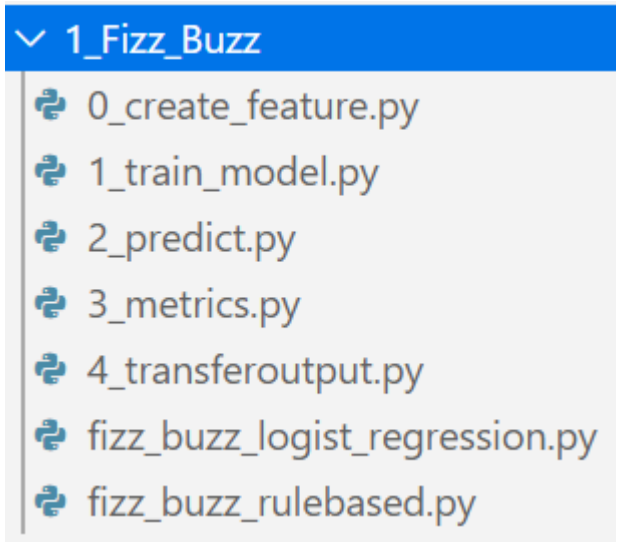
AI编程 VS 基于规则编程？

这个例子灵感来自 Joel Grus 的一次 应聘面试. 面试官让他写个程序来玩Fizz Buzz. 这是一个小孩子游戏。玩家从1数到100，如果数字被3整除，那么喊'fizz'，如果被5整除就喊'buzz'，如果两个都满足就喊'fizzbuzz'，不然就直接说数字。这个游戏玩起来就像是：

1 2 fizz 4 buzz fizz 7 8 fizz buzz 11 fizz 13 14 fizzbuzz 16 ...

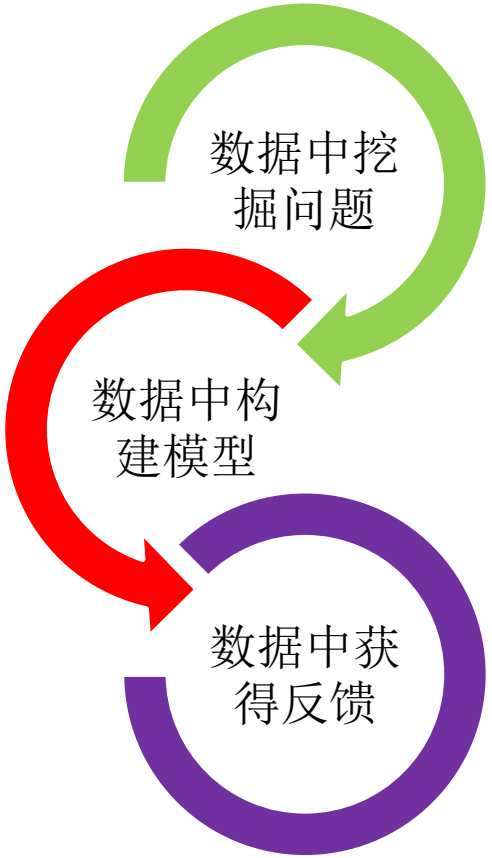


对应练习及文件


1_Fizz_Buzz
0_create_feature.py
1_train_model.py
2_predict.py
3_metrics.py
4_transferoutput.py
fizz_buzz_logist_regression.py
fizz_buzz_rulebased.py

传统思维 VS AI思维？

AI时代的思维方式

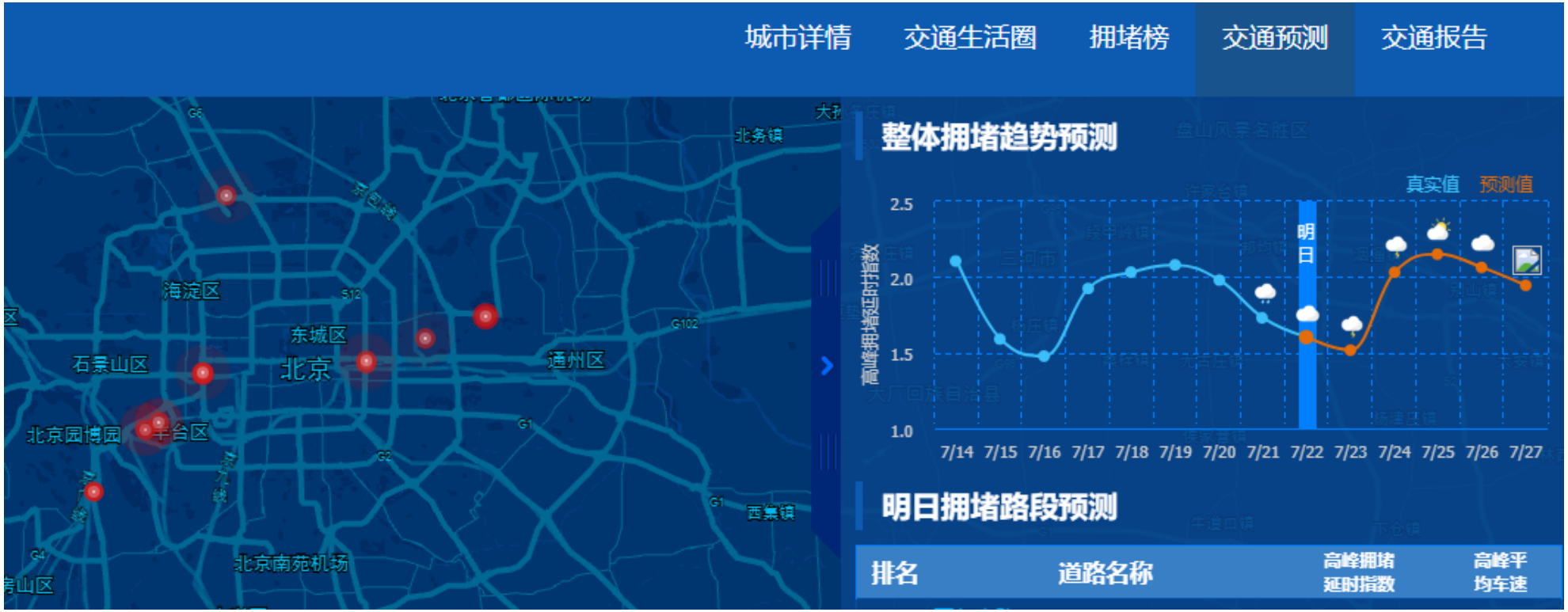


人工智能应用场景

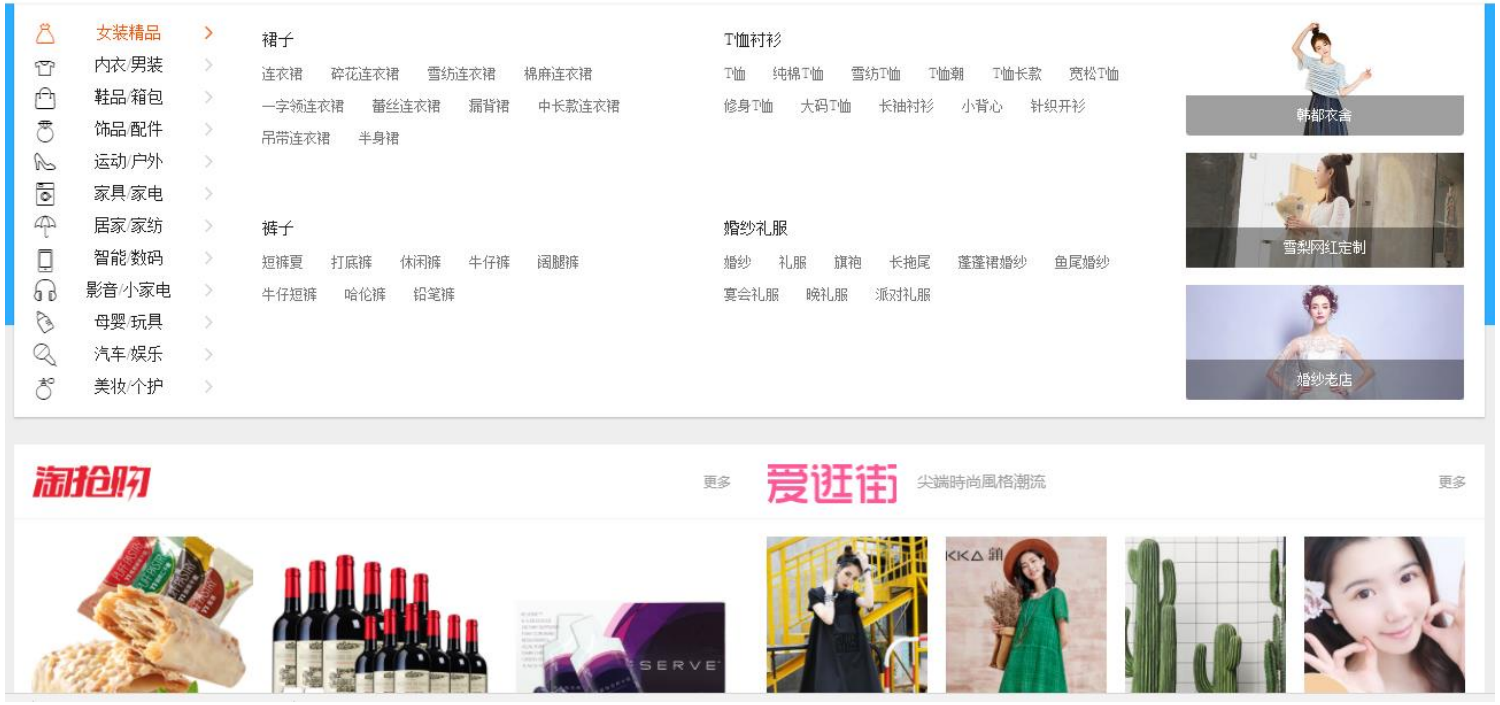
互联网应用-智能客服&助手

QA:
Q 你好吗
A 我很好

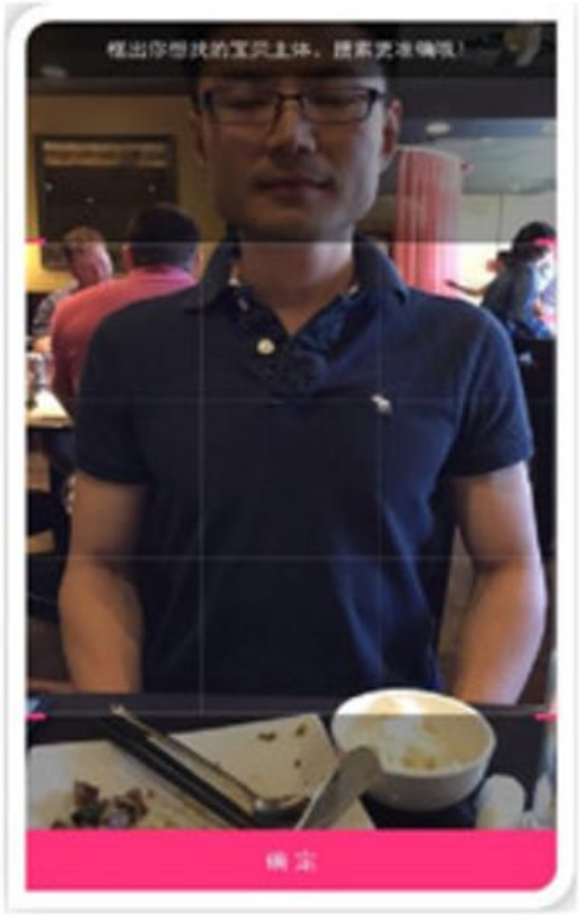
智能交通应用-实时视频分析：交通拥堵预测



电商个性化推荐：千人千面的电商



电商个性化推荐

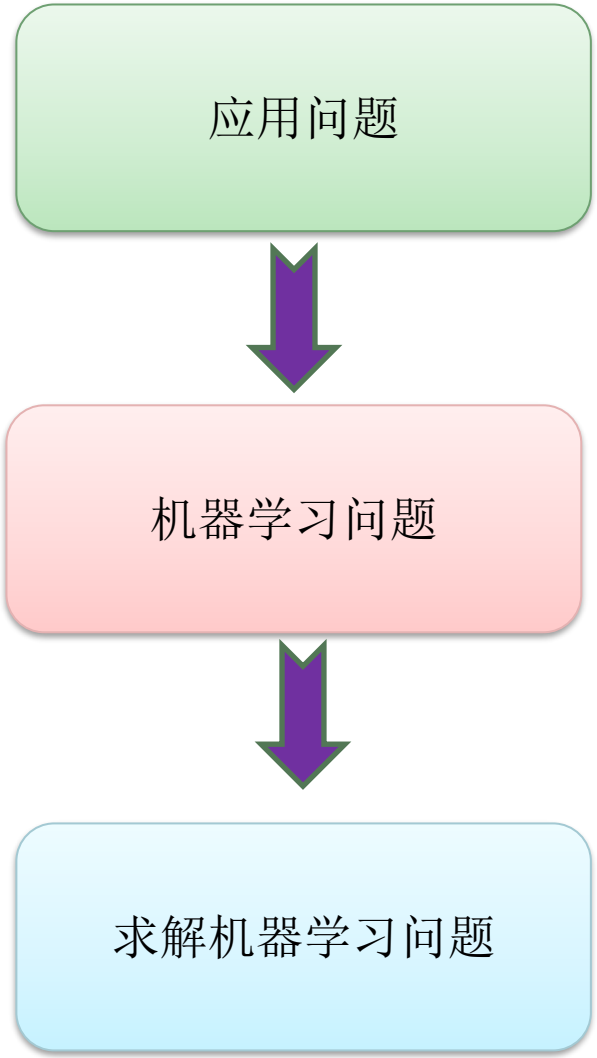


金融领域应用 征信

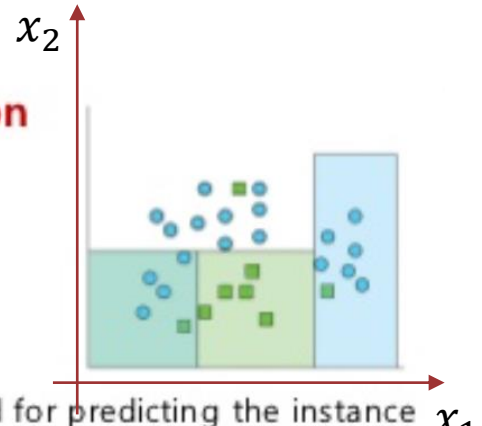

因为信用 所以简单
350
550
600
中等
良好
650
优秀
700
极好
950
较差
BETA
769
信用极好
评估时间:2016.07.06

应用到机器学习问题的映射

- 征信：
  - 用户特征 -> 征信评分
  - 回归问题
- 交通拥堵预测：
  - 历史和实时信息 -> 拥堵指数
  - 回归问题
- 对话机器人：
  - 句子 -> 句子
  - Seq2Seq + 分类问题
- 电商推荐 / 商品推荐：
  - 用户特征 -> 感兴趣的商品
  - 推荐 + 回归问题

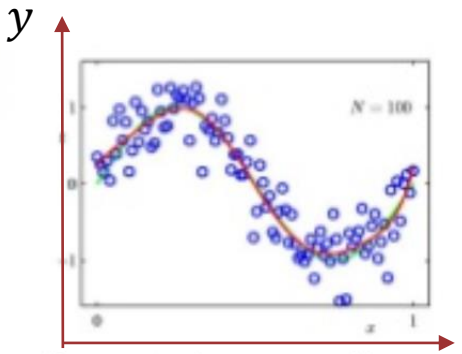


Classification



Learns a method for predicting the instance class from pre-labeled (classified) instances

Regression



An attempt to predict a continuous attribute

Clustering

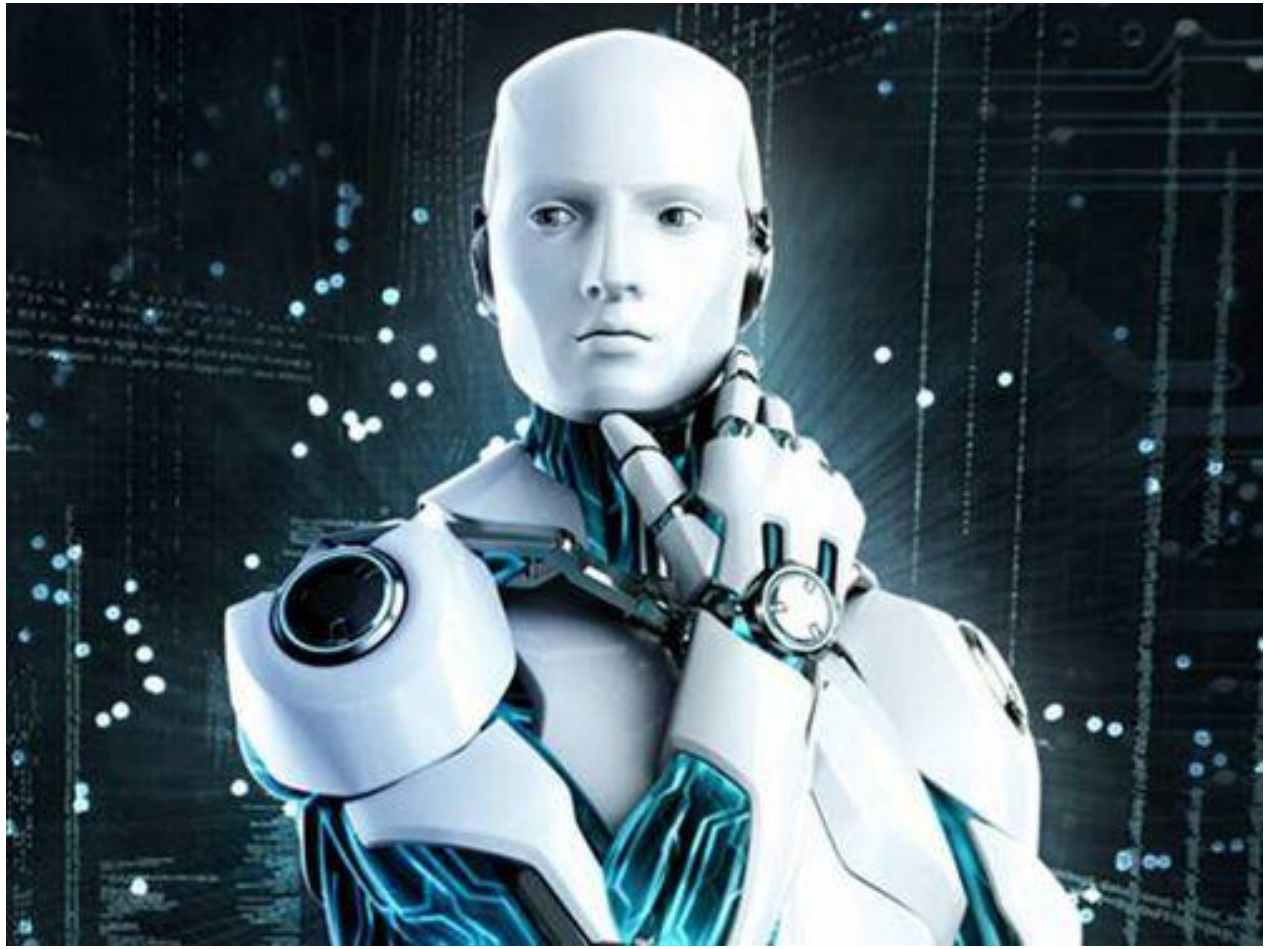
Finds "natural" grouping of instances given un-labeled data

几何视角看AI问题

人工智能技术概览

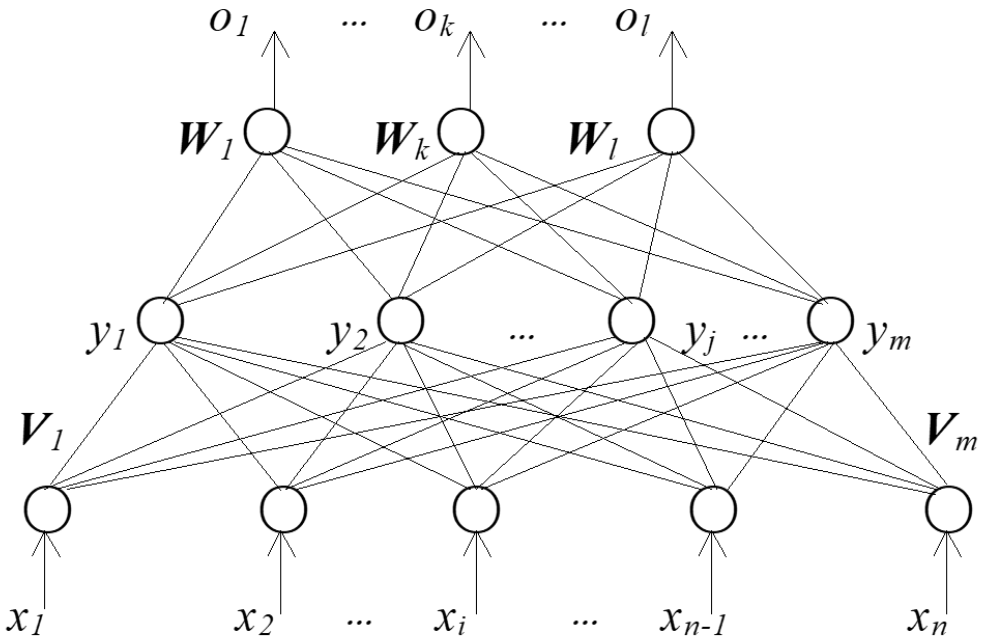
AI技术栈

- ✓解决方案
- ✓接口
- ✓算法
- ✓框架
- ✓平台



AI算法

- ✓ 分类
- ✓ 回归
- ✓ 聚类
- ✓ 深度学习



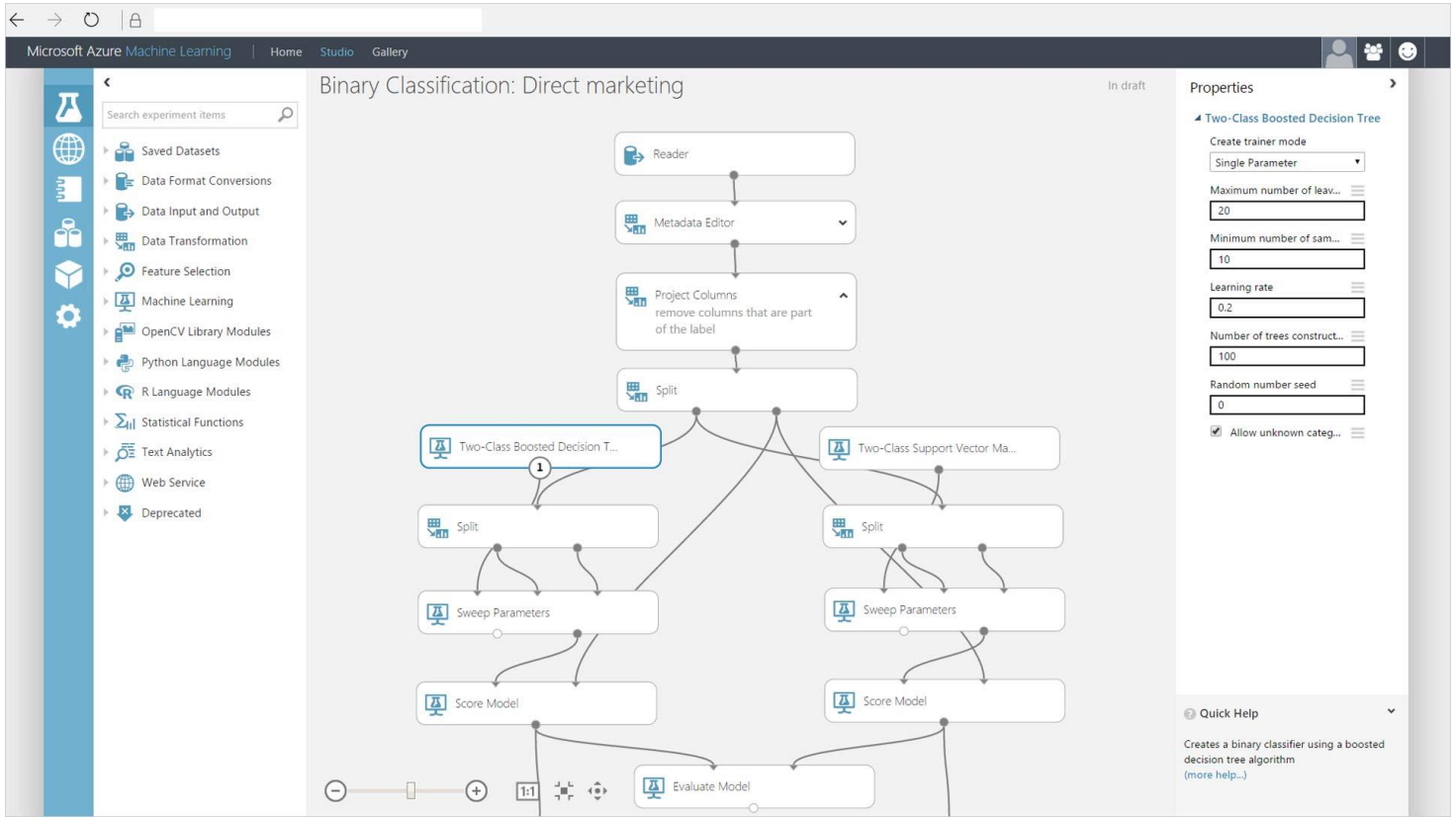
预习重点

- ✓ 重点：
  - ✓ 课程概览
  - ✓ 人工智能技术概览
  - ✓ FizzBuzz实例
  - ✓ 机器学习流程
  - ✓ KNN算法
- ✓ 难点：
  - ✓ AI与传统编程区别
  - ✓ 机器学习流程

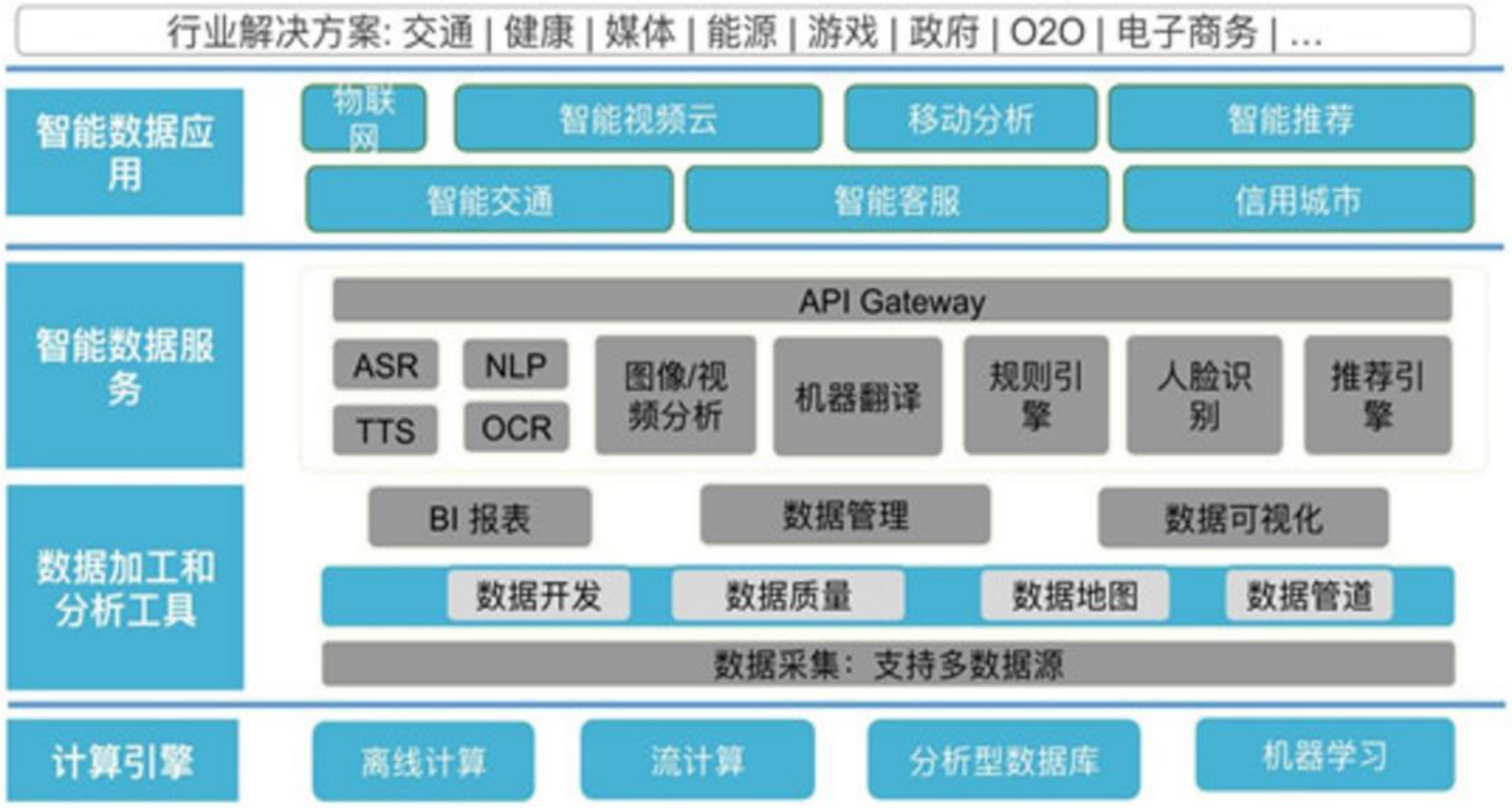
开源系统工具蓬勃发展

- ✓ 特征工程与数据预处理：
  - ✓ Numpy / Pandas / Spark
  - ✓ OpenCV
  - ✓ Jieba / NLTK / Scikit-Learn / gensim
  - ✓ …
- ✓ ML与DL框架：
  - ✓ TensorFlow / Keras / Slim
  - ✓ Scikit-Learn / XGBoost
  - ✓ Spark Mllib
  - ✓ …

人工智能分析平台


Microsoft Azure Machine Learning
Home
Studio
Gallery
Binary Classification: Direct marketing
In draft
Search experiment items
Saved Datasets
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Reader
Metadata Editor
Project Columns
remove columns that are part of the label
Split
Two-Class Boosted Decision T...
Two-Class Support Vector Ma...
Split
Split
Sweep Parameters
Sweep Parameters
Score Model
Score Model
Evaluate Model
Properties
Two-Class Boosted Decision Tree
Create trainer mode
Single Parameter
Maximum number of leav...
20
Minimum number of sam...
10
Learning rate
0.2
Number of trees construct...
100
Random number seed
0
Allow unknown categ...
Quick Help
Creates a binary classifier using a boosted decision tree algorithm
(more help...)

人工智能解决方案支撑技术栈



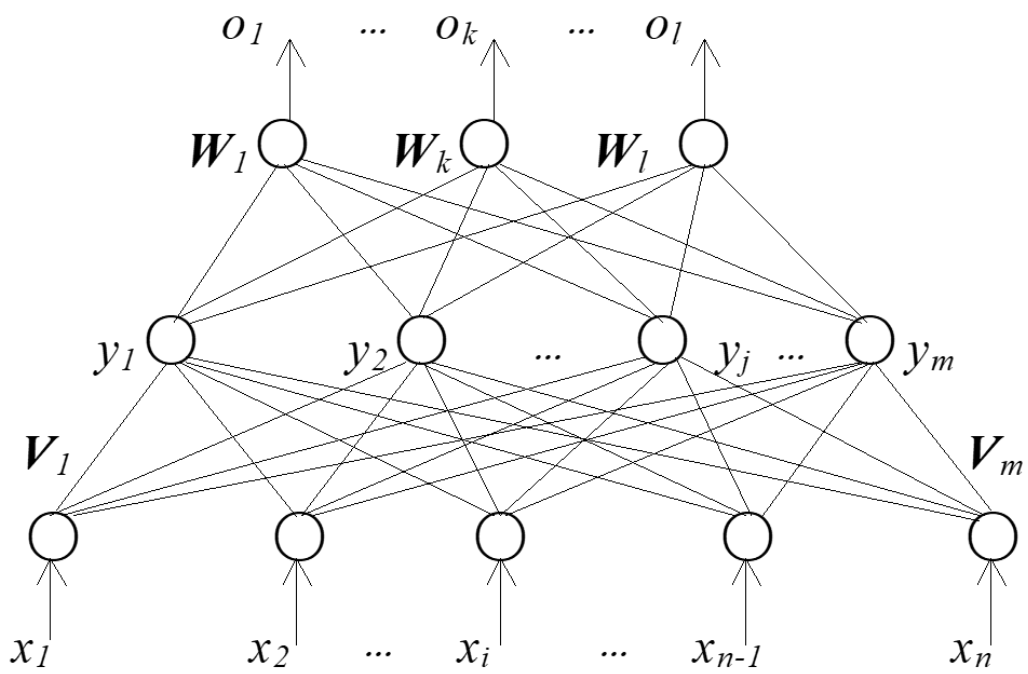
上午课程重点内容

- ✓ 重点：
  - ✓ 课程概览
  - ✓ 人工智能技术概览
  - ✓ FizzBuzz实例
- ✓ 难点：
  - ✓ AI与传统编程区别

机器学习算法

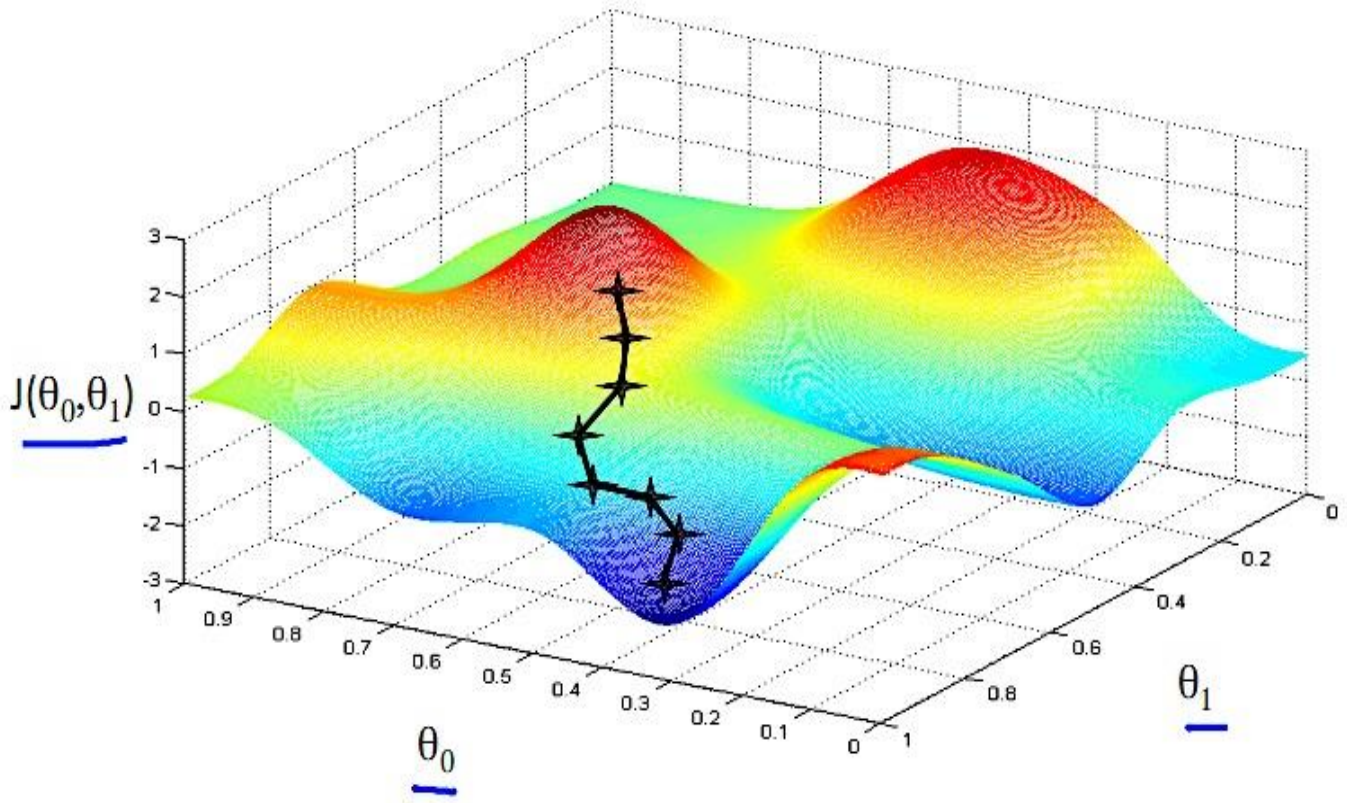
AI算法

- ✓分类
- ✓回归
- ✓聚类
- ✓深度学习
- ✓…



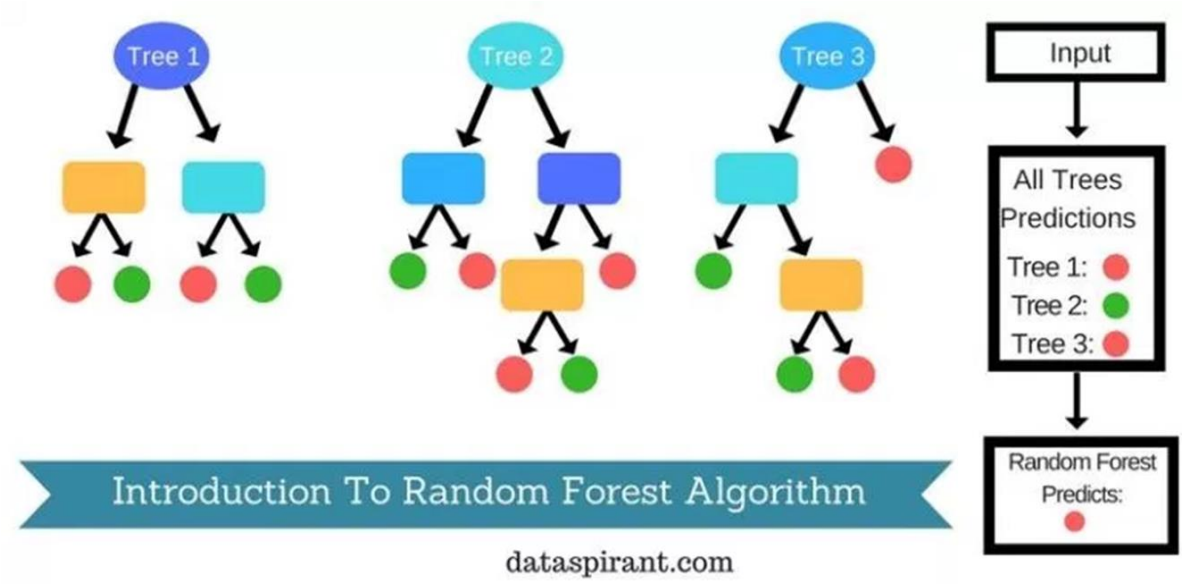
基于梯度下降训练的分类与回归算法

- ✓线性回归
- ✓逻辑回归
- ✓Lasso
- ✓Rige回归



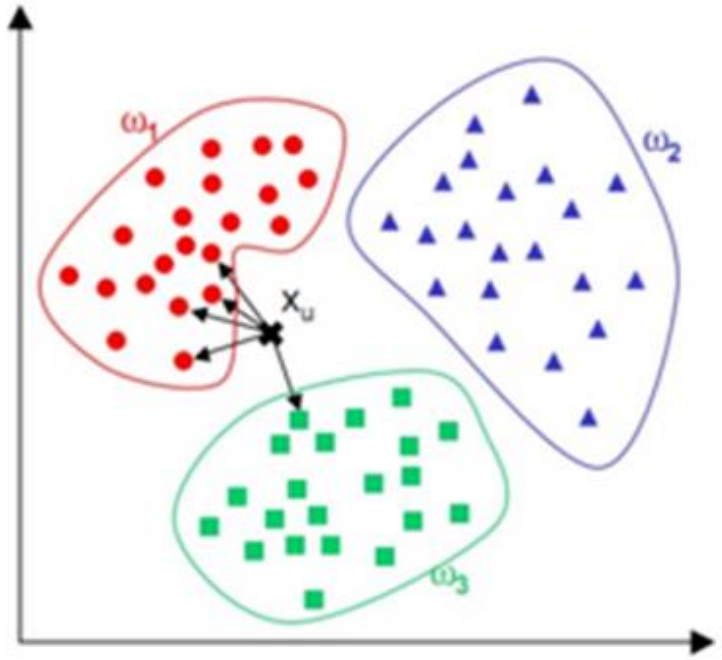
Tree算法

- ✓决策树
- ✓随机森林
- ✓GBDT



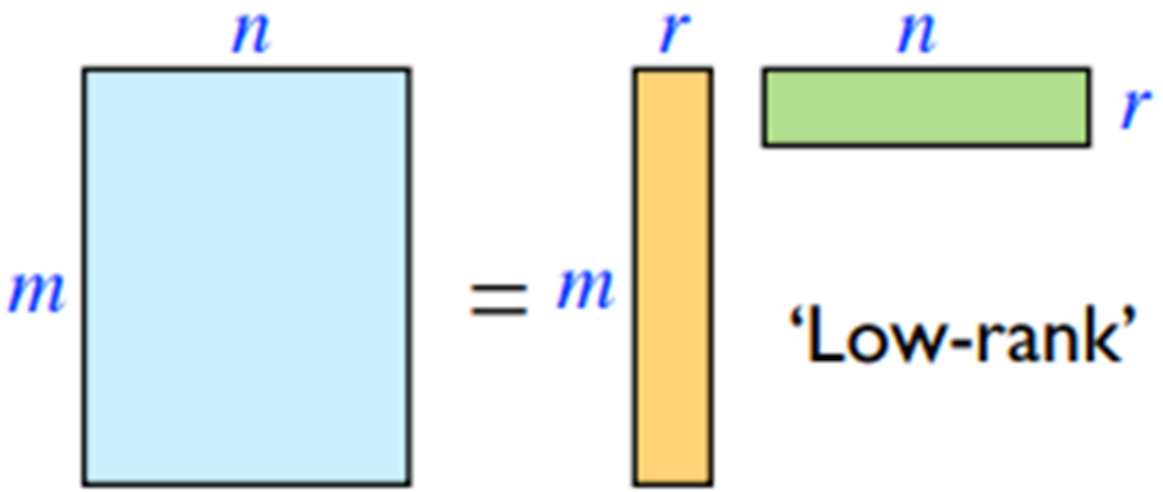
聚类算法

- ✓K-Means
- ✓K-Means ++
- ✓其他聚类算法



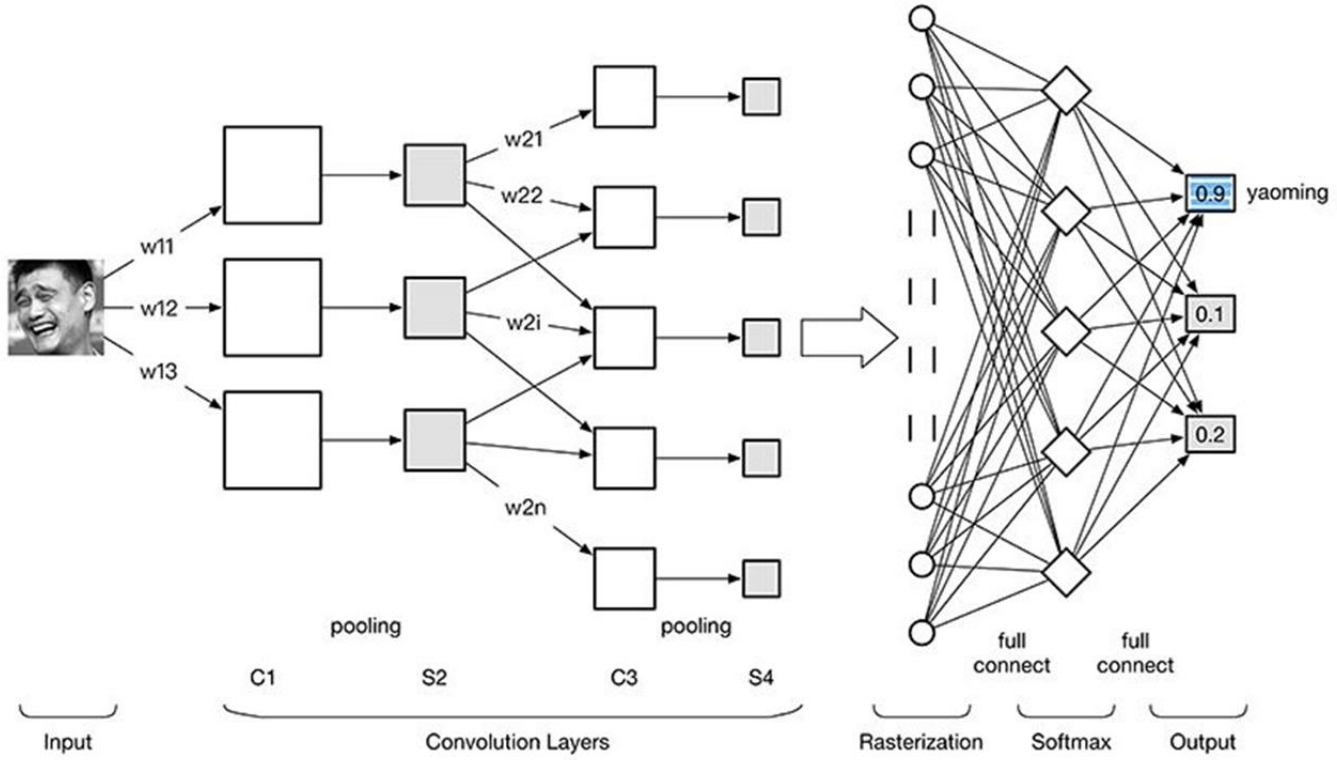
其他机器学习算法

- ✓KNN
- ✓推荐ALS
- ✓SVM等



深度学习

- ✓NN
- ✓CNN
- ✓RNN



主流机器学习库和计算框架介绍

开源系统工具蓬勃发展

- ✓ 特征工程与数据预处理：
  - ✓ Numpy / Pandas / Spark / …
  - ✓ OpenCV / …
  - ✓ Jieba / NLTK / Scikit-Learn / genism / …
  - ✓ Matplotlib
- ✓ ML与DL框架：
  - ✓ Tensorflow / Keras / Slim / …
  - ✓ Scikit-Learn / XGBoost / …
  - ✓ Spark Mllib / …
  - ✓ …

Scikit-Learn功能模块

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

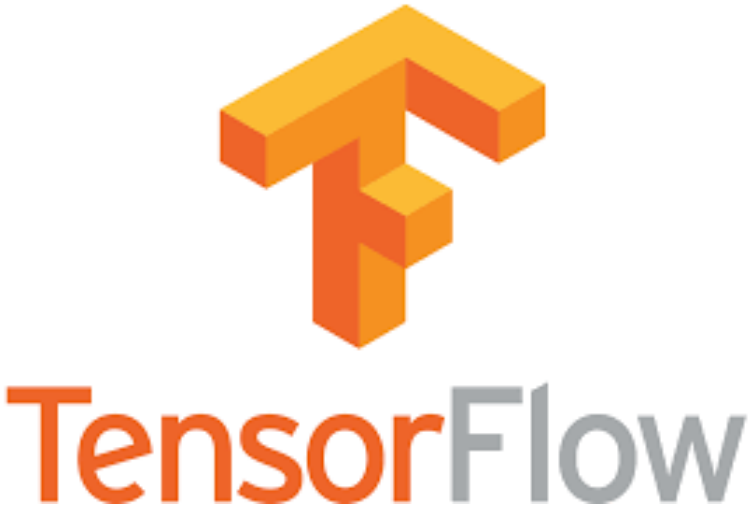
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Spark MLlib

- ✓MLlib 是Spark的可以扩展的机器学习库
- ✓包括分类，回归，聚类，协同过滤，降维
- ✓包括调优的部分

TensorFlow

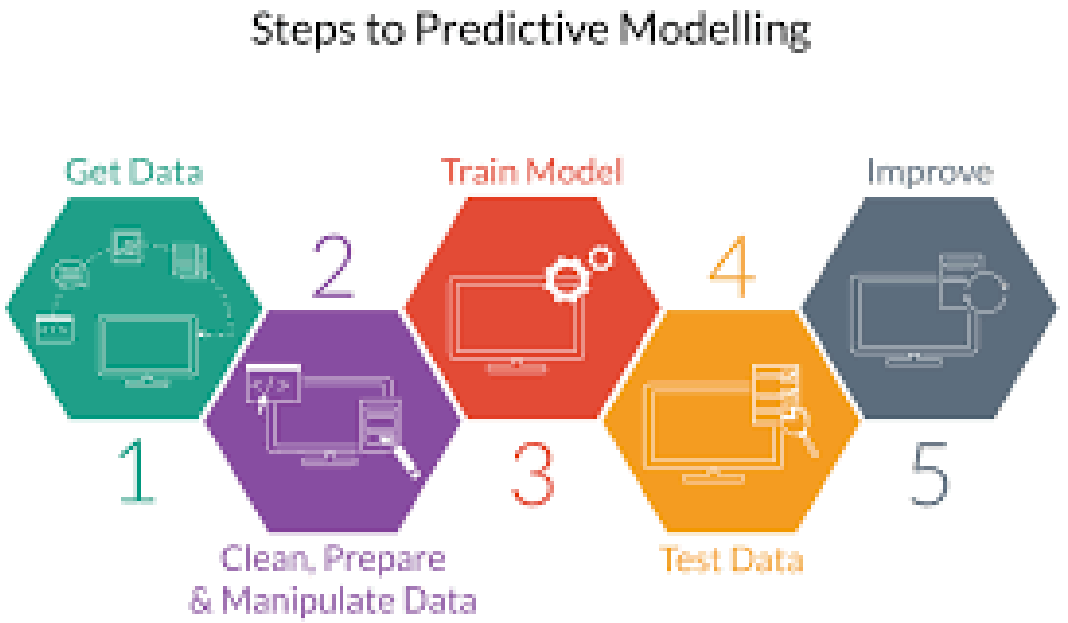
- ✓数据流图（data flow graphs）
- ✓用于数值计算的开源软件库
- ✓多种平台上展开计算
- ✓用于机器学习和深度神经网络



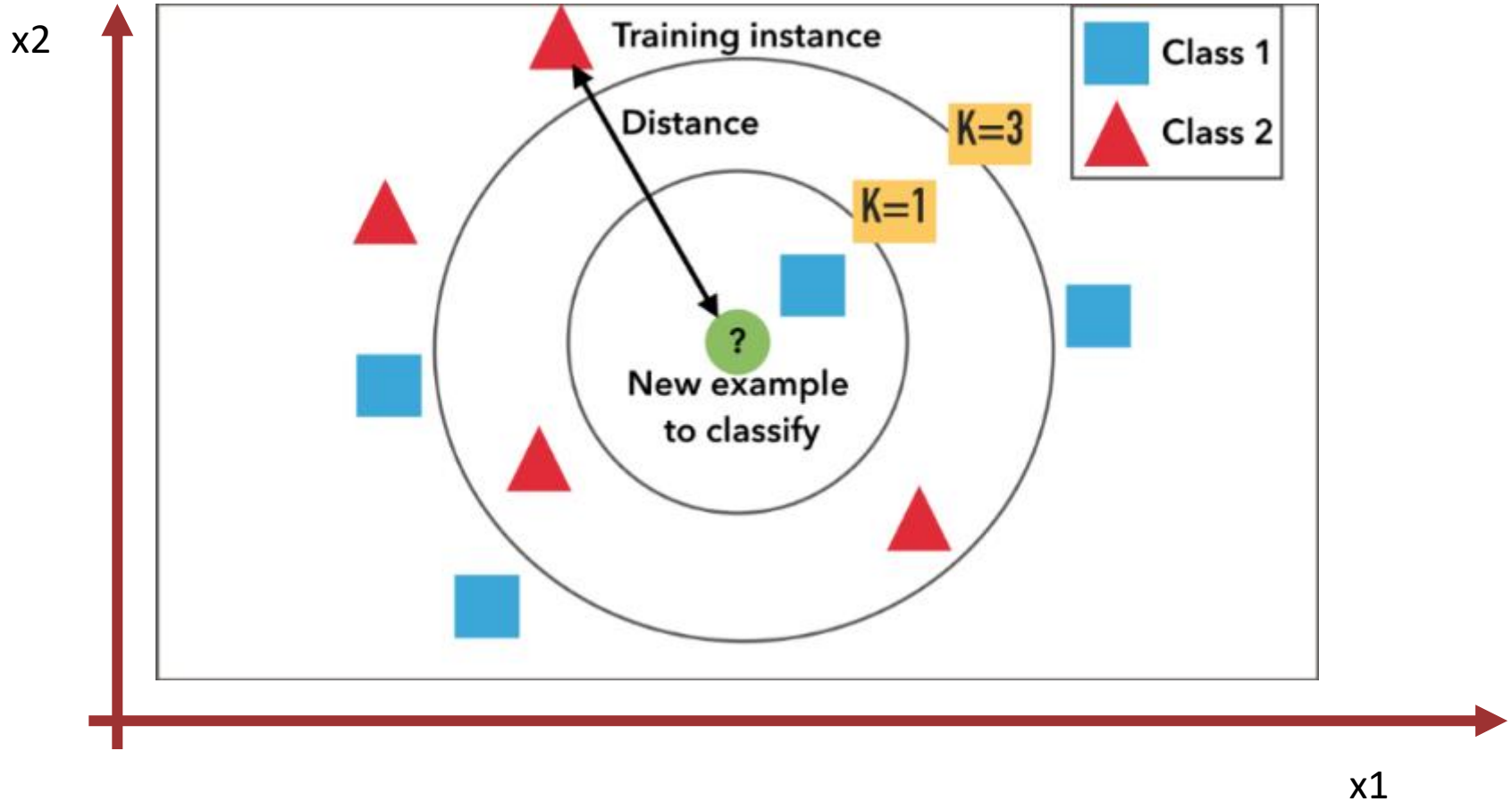
如何进行机器学习

机器学习基本工作流程

- ✓ 问题建模
- ✓ 获取数据
- ✓ 特征工程
- ✓ 模型训练与验证
- ✓ 模型诊断与调优
- ✓ 线上运行



KNN算法


x2
Training instance
Distance
K=3
K=1
Class 1
Class 2
?
New example
to classify
x1

k nearest neighbor classifier

TRAIN-KNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' \leftarrow$ PREPROCESS($\mathbb{D}$)
2 $k \leftarrow$ SELECT-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathbb{D}', k$

APPLY-KNN($\mathbb{C}, \mathbb{D}', k, d$)
1 $S_k \leftarrow$ COMPUTENEARESTNEIGHBORS($\mathbb{D}', k, d$)
2 **for each** $c_j \in \mathbb{C}$
3 **do** $p_j \leftarrow |S_k \cap c_j|/k$
4 **return** $\arg\max_j p_j$

▶ **Figure 14.4** kNN training (with preprocessing) and testing. p_j is an estimate for $P(c_j|S_k) = P(c_j|d)$. c_j denotes the set of all documents in the class c_j.

https://nlp.stanford.edu/IR-book/html/htmledition/k-nearest-neighbor-1.html

k nearest neighbor regressor

In *k*-NN regression, the *k*-NN algorithm is used for estimating continuous variables. One such algorithm uses a weighted average of the *k* nearest neighbors, weighted by the inverse of their distance.

This algorithm works as follows:

1.Compute the Euclidean from the query example to the labeled examples.
2.Order the labeled examples by increasing distance.
3.Find a heuristically optimal number k of nearest neighbors, based on RMSE. This is done using cross validation.
4.Calculate an inverse distance weighted average with the k-nearest multivariate neighbors.

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#k-NN_regression

均方根误差（或称方均根偏移、均方根差、方均根差等，英文：root-mean-square deviation、root-mean-square error、RMSD、RMSE）

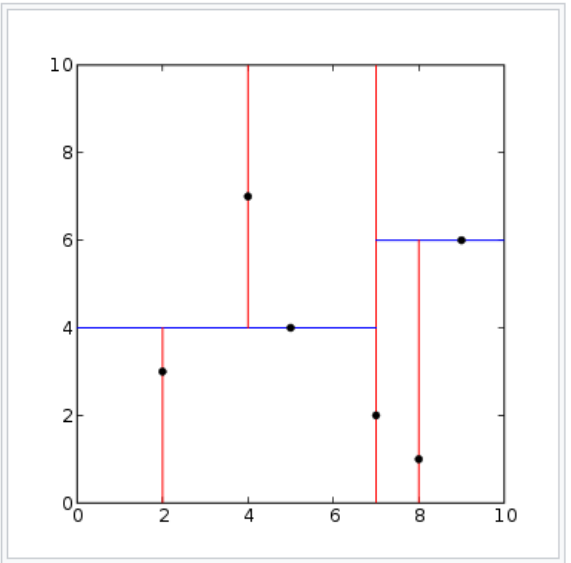
$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{T}}.$$

K-D tree construction

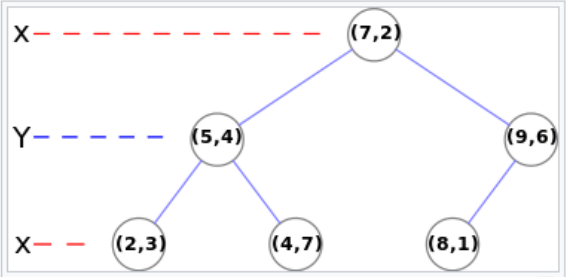
```
function kdtree (list of points pointList, int depth)
{
    // Select axis based on depth so that axis cycles through all valid values
    var int axis := depth mod k;

    // Sort point list and choose median as pivot element
    select median by axis from pointList;

    // Create node and construct subtree
    node.location := median;
    node.leftChild := kdtree(points in pointList before median, depth+1);
    node.rightChild := kdtree(points in pointList after median, depth+1);
    return node;
}
```



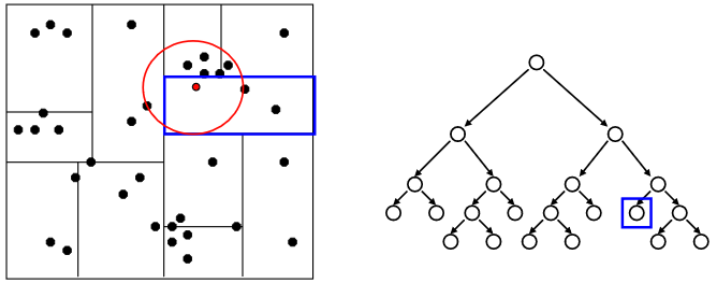
k-d tree decomposition for the point set (2,3), (5,4), (9,6), (4,7), (8,1), (7,2).



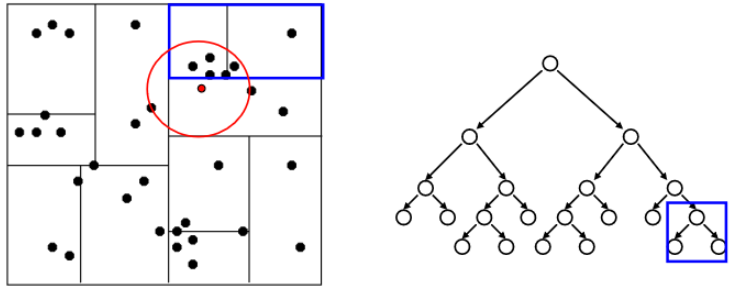
The resulting *k*-d tree.

https://en.wikipedia.org/

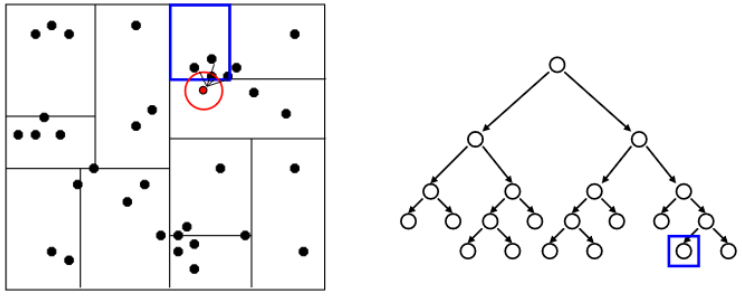
KNN search with K-D tree optimization



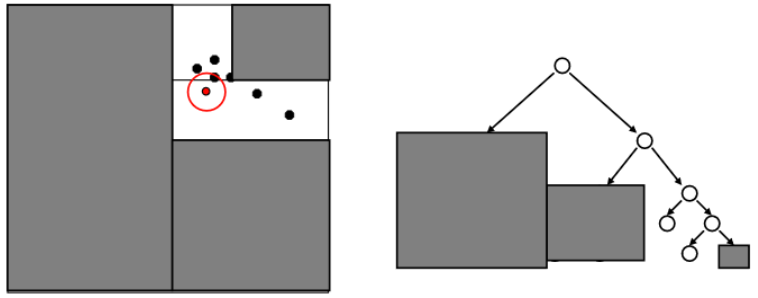
When we reach a leaf node: compute the distance to each point in the node.



Then we can backtrack and try the other branch at each node visited.



Each time a new closest node is found, we can update the distance bounds.



Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could NOT include the nearest neighbor.

https://slideplayer.com/slide/3273367/

KNN distance selection

There was no optimal distance metric that can be used for all types of datasets, as the results show that each dataset favors a specific distance metric, and this result complies with the no-free-lunch theorem.

1.1 Manhattan (MD): The Manhattan distance, also known as L_1 norm, Taxicab norm, Rectilinear distance or City block distance, which considered by Hermann Minkowski in 19th-century Germany. This distance represents the sum of the absolute differences between the opposite values in vectors.

$$MD(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

1.2 Chebyshev (CD): Chebyshev distance is also known as maximum value distance [26], Lagrange [72] and chessboard distance [61]. This distance is appropriate in cases when two objects are to be defined as different if they are different in any one dimension [75]. It is a metric defined on a vector space where distance between two vectors is the greatest of their difference along any coordinate dimension.

$$CD(x,y) = \max_i |x_i - y_i|$$

1.3 Euclidean (ED): Also known as L_2 norm or Ruler distance, which is an extension to the Pythagorean Theorem. This distance represents the root of the sum of the square of differences between the opposite values in vectors.

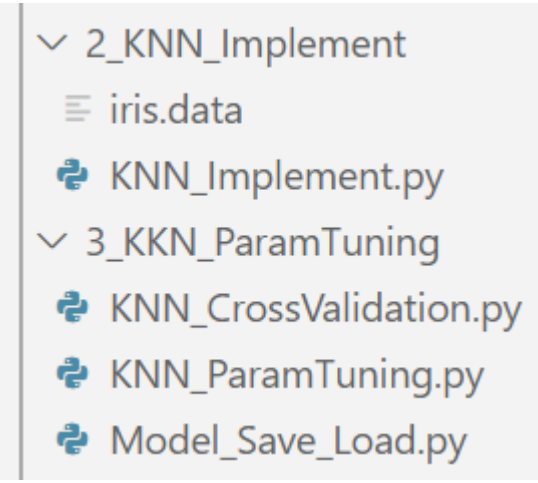
$$ED(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

3.2 Cosine distance (CosD): The Cosine distance, also called angular distance, is derived from the cosine similarity that measures the angle between two vectors, where Cosine distance is obtained by subtracting the cosine similarity from one.

$$CosD(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Effects of Distance Measure Choice on KNN Classifier Performance - A Review

对应练习及文件


2_KNN_Implement
iris.data
KNN_Implement.py
3_KKN_ParamTuning
KNN_CrossValidation.py
KNN_ParamTuning.py
Model_Save_Load.py

本节课程重点内容 （适用复习）

- ✓ 重点：
  - ✓ 课程概览
  - ✓ 人工智能技术概览
  - ✓ FizzBuzz实例
  - ✓ 机器学习流程
  - ✓ KNN算法
- ✓ 难点：
  - ✓ AI与传统编程区别
  - ✓ 机器学习流程

参考资料

- ✓ 统计学习方法
- ✓ 机器学习
- ✓ 深度学习
- ✓ 深度学习：核心技术、工具与案例解析
- ✓ 数据仓库与数据分析教程
- ✓ 数据挖掘：概念与技术
- ✓ Scikit-learn, TensorFlow, Keras, Spark MLlib相关文档

预习重点内容

- ✓ 重点：
  - ✓ 课程概览
  - ✓ 人工智能技术概览
  - ✓ FizzBuzz实例
  - ✓ 机器学习流程
  - ✓ KNN算法

学习建议与小助手

- 如何解决英文阅读困难？
  - 有道翻译
  - Google翻译
- 如何解决Bug和环境问题？
  - 百度
  - Bing
  - Google
- 如何生成漂亮作业与报告？
  - 可视化matploitlib
  - Jupyter-notebook
- 课上走神?
  - 回看视频
- 知识理解和交流?
  - 组内小伙伴
- 在线笔记?
  - Onenote
  - 有道笔记
  - evennote