

Cursor数据可视化与洞察



>> 今天的学习目标

Cursor数据可视化与洞察

- Python基础语法与AI
- 机器学习七步法
- CASE：客户续保预测

数据可视化

模型洞察

Python基础语法与AI

Python是数据分析的首选语言

Thinking: 如何选择数据分析语言？

- Python是首选的数据分析语言
- 在数据分析/数据科学领域中占有率70%
- 有强大的生态（社区+工具）

科学计算: Sklearn, Numpy, Pandas

人工智能: Tensorflow, PyTorch

网络爬虫: Scrapy, Request, BeautifulSoup

运筹优化: ortools, pulp

Python生态强大，代码简洁

相对其他语言Python更好上手，浙江高考将Python列为可选科目之一

Worldwide, Jun 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	31.6 %	+4.3 %
2		Java	17.67 %	-2.4 %
3		Javascript	8.02 %	-0.2 %
4		C#	6.87 %	-0.4 %
5		PHP	6.02 %	-0.9 %
6		C/C++	5.69 %	-0.2 %
7		R	3.86 %	-0.1 %
8		Objective-C	2.5 %	-0.3 %
9		Swift	2.24 %	-0.1 %
10	↑	TypeScript	1.86 %	+0.2 %

Python基础语法

学习Python可以从以下3个维度掌握

- 基础语法

输入，输出，条件判断，循环语句，注释，引用包，函数定义

- 数据结构

列表、元组、字典、集合

- 常用分析工具

Numpy, Pandas

Python基础语法

- 输入输出

```
name = input("What's your name?")
```

```
sum = 100+100
```

```
print ('hello', name)
```

```
print ('sum', sum)
```

Python基础语法

- 条件判断 if ... else ...

```
score = 95
```

```
if score >= 90:
```

```
    print('Excellent')
```

```
else:
```

```
    if score < 60:
```

```
        print('Fail')
```

```
    else:
```

```
        print('Good Job')
```

Python基础语法

- 循环语句 for ... in

```
sum = 0
```

```
for number in range(11):
```

```
    sum = sum + number
```

```
print(sum)
```

- 循环语句 while

```
sum = 0
```

```
number = 1
```

```
while number < 11:
```

```
    sum = sum + number
```

```
    number = number + 1
```

```
print(sum)
```


Python基础语法

- 注释

```
# -*- coding: utf-8 -*-
```

```
'''
```

这是多行注释，用三个单引号

这是多行注释，用三个单引号

这是多行注释，用三个单引号

```
'''
```

- 引用模块/包: import

引用一个或多个包

```
import module_name1,module_name2
```

导入包中指定模块

```
from package_name import module_name
```

- 函数定义 def

```
def addone(score):
```

```
    return score + 1
```

```
print(addone(99))
```

Python数据结构

数据类型：列表、元组、字典、集合

- 列表： []

```
lists = ['a','b','c']
```

- 元组 (tuple)

```
tuples = ('tupleA','tupleB')
```

- 字典 {dictionary}

```
score = {'guanyu':95,'zhangfei':96}
```

- 集合： set

```
s = set(['a', 'b', 'c'])
```

Python数据结构（列表）

列表

```
lists = ['zhangfei','guanyu','liubei']
```

列表中添加元素

```
lists.append('dianwei')
```

```
print(lists)
```

```
print(len(lists))
```

在指定位置添加元素

```
lists.insert(0,'diaochan')
```

删除末尾元素

```
lists.pop()
```

```
print(lists)
```

Thinking : 什么是人工智能?

The theory and development of computer systems able to perform tasks normally requiring human intelligence.

— — — *Oxford Dictionary*

Using data to solve problems.

— — *cy*

Using data to solve problems



AI的本质

AI就是利用数据，解决问题

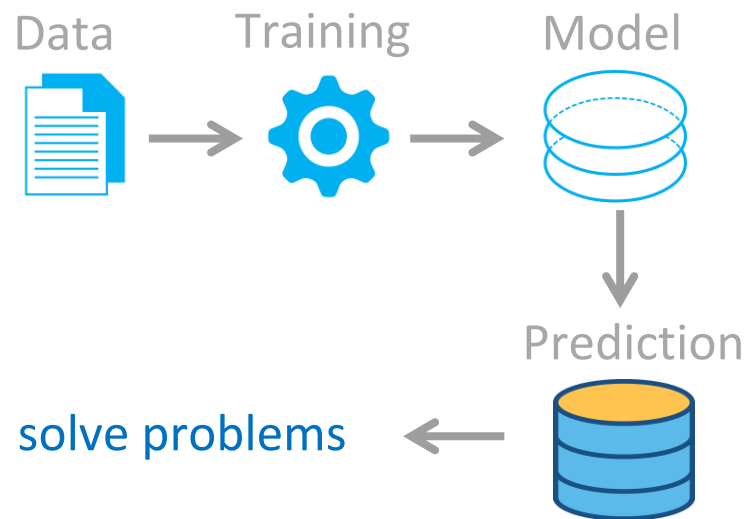
Using data to solve problems

Training

Prediction

Using data

solve problems



训练阶段：通过对数据的训练，创建一个预测模型并对其进行微调。

模型生成：预测模型可以从这些数据背后找出答案来，帮我们解决某个问题。

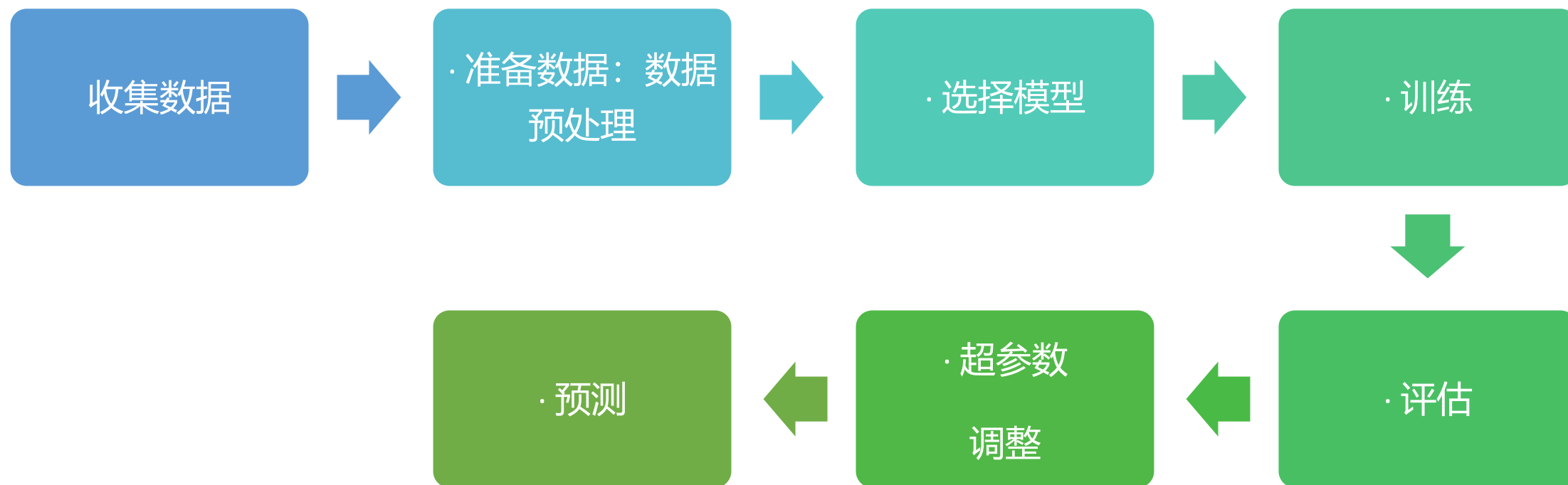
预测阶段：通过测试集完成模型评估，从而了解模型在测试集中的有效性。

过程中，预测模型会被不断改进和使用。

机器学习的步骤

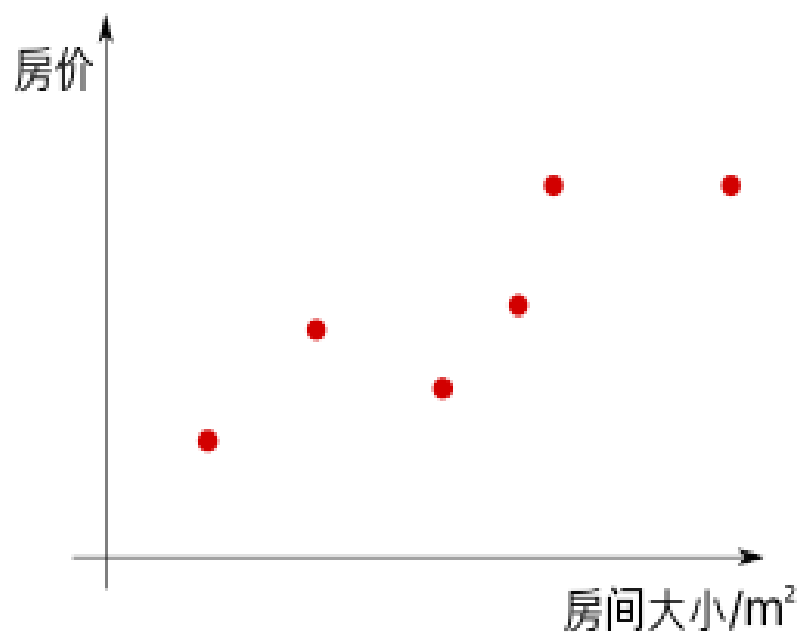
- Thinking: 如何预测房价?

机器学习的7个步骤



机器学习的步骤

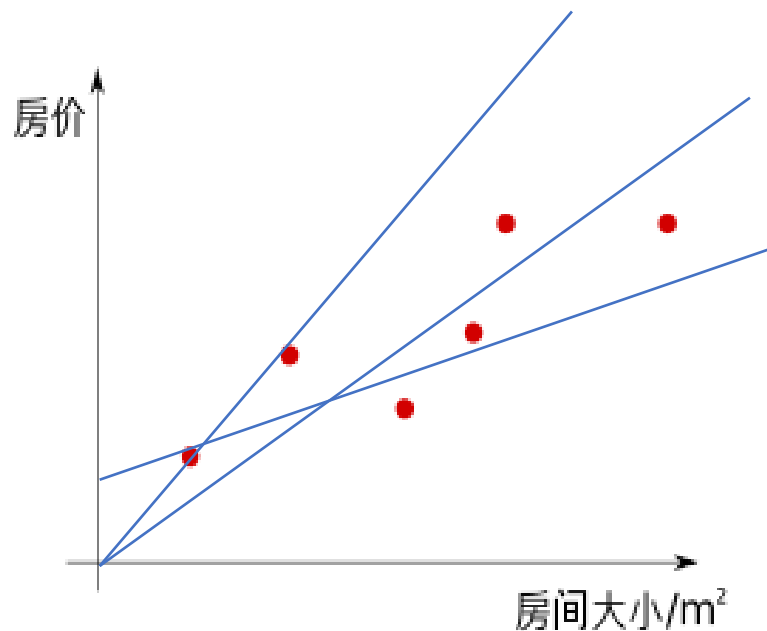
- Thinking: 如何预测房价?



房间大小x	房价y
50	82
80	118
100	172
200	302
.....

机器学习的训练过程

- 训练是机器学习的主要步骤
- 针对预测房价这个例子，我们可以用简单的线性模型： $y = w * x + b$



机器学习的训练过程

- 在机器学习中，我们有很多特征，基于这些特征，我们需要训练在Model中的权重 w
- 这些特征值构成的矩阵，称之为权重矩阵 weights
- 同时，还存在偏差，称之为 biases

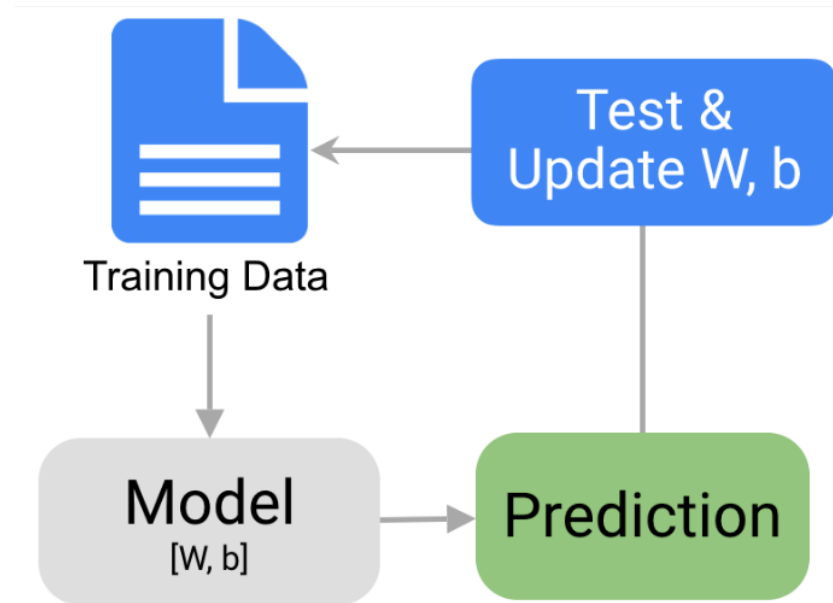
房间大小	区域	周围绿化	周边配套	房型	房价y
50	海淀	A	A	style1	82
80	通州	B	A	style1	118
100	朝阳	C	B	style2	172
200	海淀	C	C	style3	302
.....

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

$$f(x) = \mathbf{w}^T \mathbf{x} + b$$

机器学习的训练过程

- 机器学习的过程，就是在搜索空间中对 W 和 b 进行搜索的过程，使得模型的准确率达到某个标准
- 一个训练步骤(training step)，称之为一次迭代。目的在于更新权重和变量
- 通过多次迭代，模型中的参数不断进行更新。就好像是在数据中进行线性拟合
- 当完成训练时，可以使用模型对房价进行预测



机器学习的模型选择

- Thinking: 什么是回归问题, 什么是分类问题?
- Thinking: 什么是线性回归, 什么是逻辑回归?

机器学习的模型选择

- 判断一个问题是分类，还是回归：
输出的数据类型：离散 or 连续

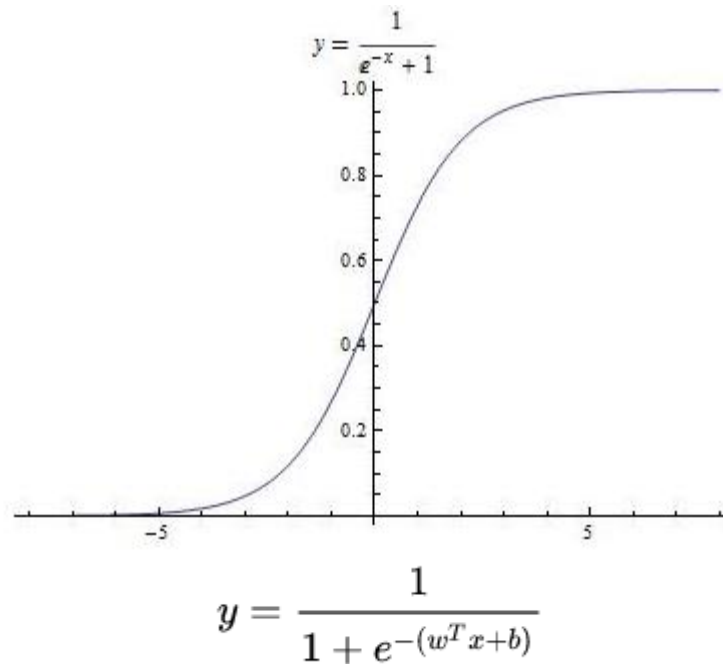
- 线性回归：

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

$$f(x) = \mathbf{w}^T \mathbf{x} + b$$

- 逻辑回归：

使用sigmoid函数，实际上是分类算法



机器学习的模型选择

- Thinking: 如何判断杯子里盛的是水, 还是饮料?

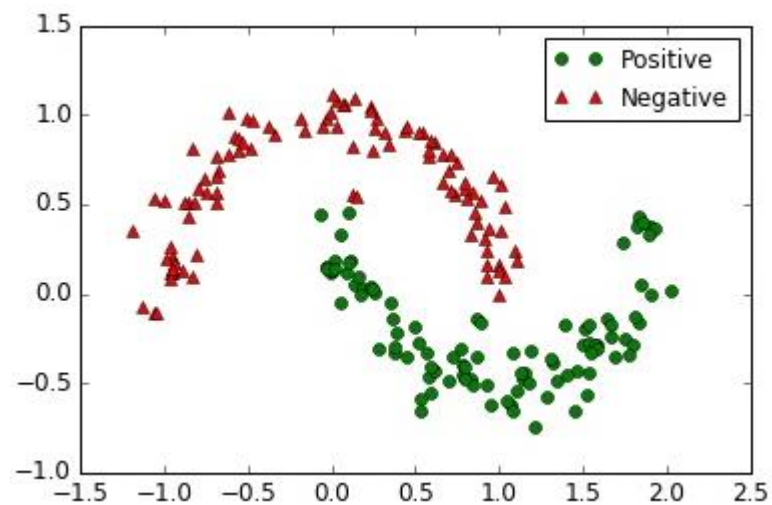
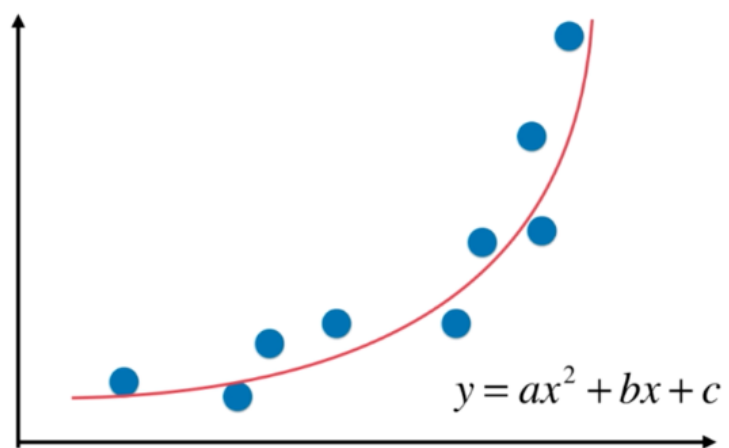


机器学习的模型选择

Color	Sugar	Classification
252	0.1%	water
210	4%	beverage
150	8%	beverage
250	0.5%	water
.....

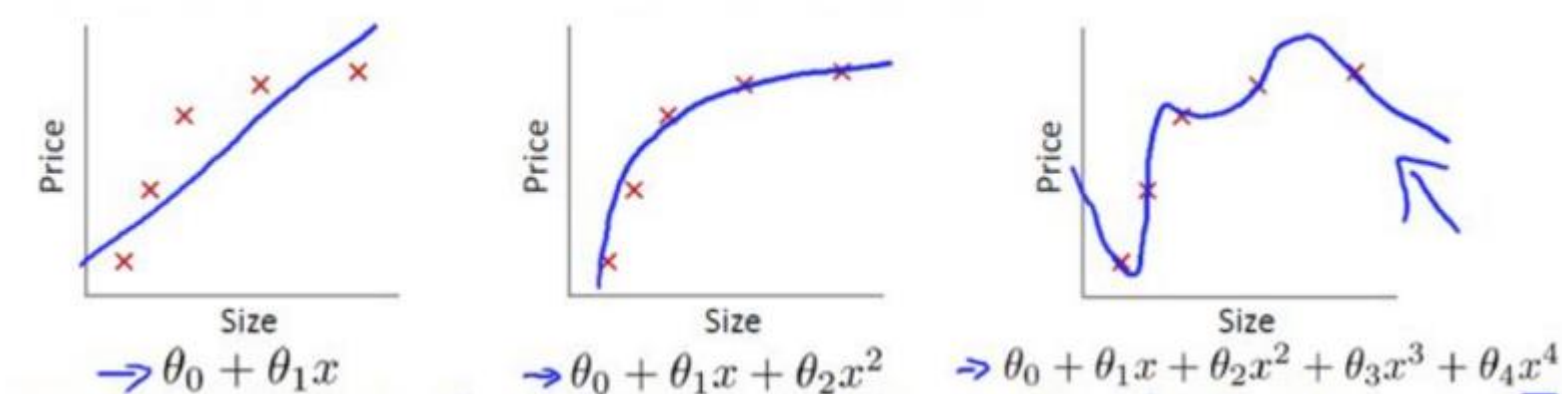
可见光的波长为400 ~ 760nm，白色是包含光谱中所有颜色的集合
因此采用Color这里采用颜色值

机器学习的模型选择



机器学习的特征构造

如何用线性回归模型拟合非线性关系



机器学习的评估

- 对数据的评估有多种方式：
- 我们会选择一部分数据作为测试集，比如20%或者10%



Training
80%



Evaluation
20%

超参数调整

- 我们还可以对模型中的参数进行调整，比如epoch的次数，学习率等
- 这些参数通常被称为超参数。调整超参数的过程比起科学更像是艺术。这是实验性的过程，并很大程度上取决于具体的数据集、模型和训练过程

数据分析模型

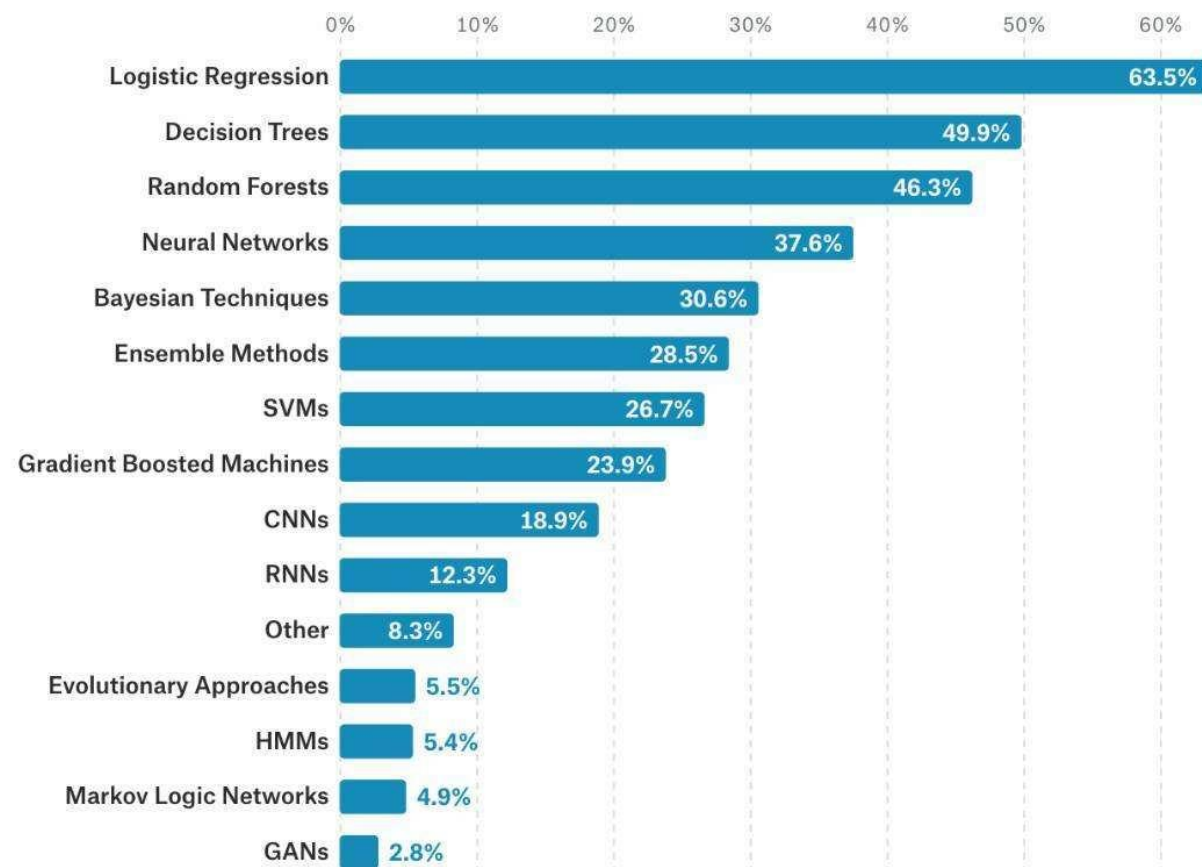
——10大经典模型

- 分类算法：C4.5, 朴素贝叶斯 (Naive Bayes) , SVM, KNN, Adaboost, CART
- 聚类算法：K-Means, EM
- 关联分析：Apriori
- 连接分析：PageRank

数据分析模型

主流模型

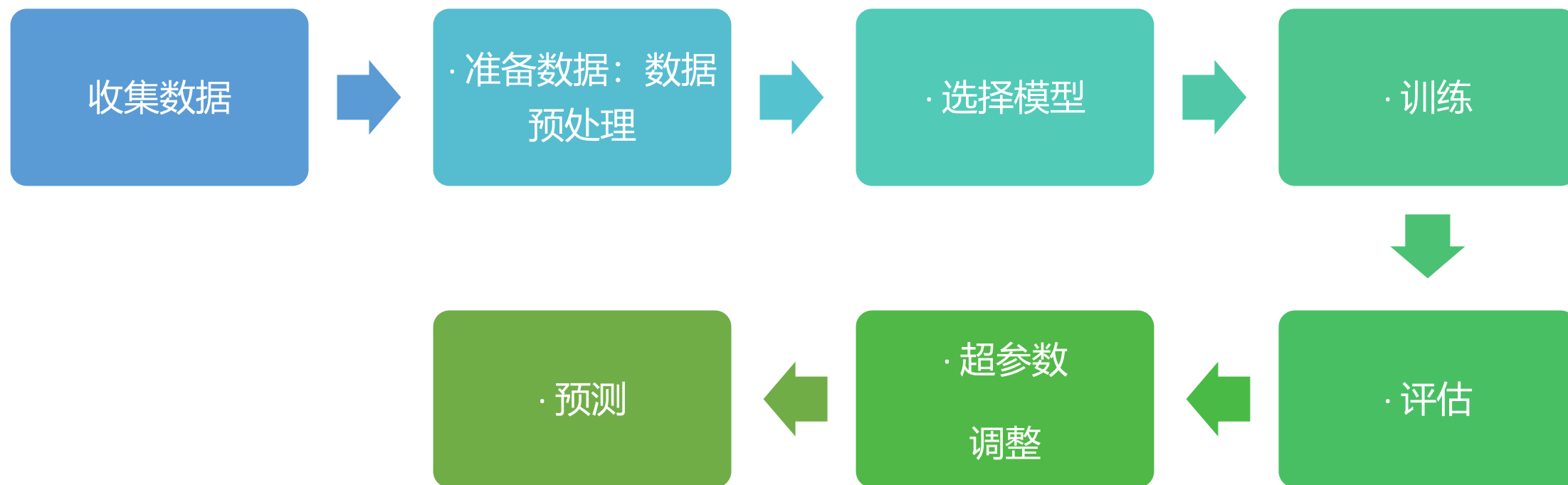
- Logistic Regression, Decision Trees, Random Forests在业界依然是主流



7,301 responses

[View code in Kaggle Kernels](#)

Summary



Thinking & Action

Action: 寿险客户续保预测

寿险行业是一个高度依赖于客户忠诚度和持续支付保费的领域。准确预测客户是否续保可以帮助保险公司提前采取措施，减少流失率，比如提供更加个性化的服务或者优化产品设计。

训练集: train.csv 1000条

测试集: test.csv 200条

<https://www.kaggle.com/t/467370365f1747868eadbd65eeb970c5>

英文名称	含义描述
policy_id	每个保单的唯一标识符
age	客户的年龄，范围从18岁到70岁。
gender	客户的性别，分为“男”和“女”。
birth_region	客户的出生地
insurance_region	客户投保时所在的地区
income_level	客户的收入水平，分为“低”、“中”和“高”。
education_level	客户的最高教育程度，分为“高中”、“本科”、“硕士”和“博士”。
occupation	客户的职业，例如“销售”、“经理”、“设计师”、“工程师”、“医生”等。
marital_status	客户的婚姻状况，分为“单身”、“已婚”和“离异”。
family_members	客户的家庭成员数量，单身客户通常有1-2人，已婚客户通常有3-6人。
policy_type	保单的类型，例如“平安六福保”、“盛世福尊悦版”、“优悦版”等。
policy_term	保单的有效期限，分为“1年”、“5年”、“10年”和“20年”。
premium_amount	客户每年需要支付的保费金额
policy_start_date	保单开始生效的日期
policy_end_date	保单到期的日期，根据保单生效日期和保单期限计算。
claim_history	保单是否有过理赔记录，分为“是”和“否”。
renewal	是否续保，Yes 或者 No

决策树与随机森林

决策树：

- 决策树基本上就是把我们的经验总结出来
- 常见的决策树算法有C4.5、ID3和CART
- Thinking：如何构造一个判断是否去打篮球的决策树

将哪个属性（天气、温度、湿度、刮风）作为根节点是个关键问题

天气	温度	湿度	刮风	是否打球
晴天	高	中	否	否
晴天	高	中	是	否
阴天	高	高	否	是
小雨	高	高	否	是
小雨	低	高	否	否
晴天	中	中	是	是
阴天	中	高	是	否

决策树与随机森林

信息、熵以及信息增益：

- 引用香农的话来说，信息是用来消除随机不确定性的东西
- 对于机器学习中的决策树而言，如果带分类的事物集合可以划分为多个类别当中，则某个类（ x_i ）的信息可以定义为

$$I(X = x_i) = -\log_2 p(x_i)$$

随机变量的信息

当 x_i 发生时的概率

- 熵是约翰.冯.诺依曼建议使用的命名，熵=信息的期望值

$$H(X) = \sum_{i=1}^n p(x_i) I(x_i) = -\sum_{i=1}^n p(x_i) \log p(x_i)$$

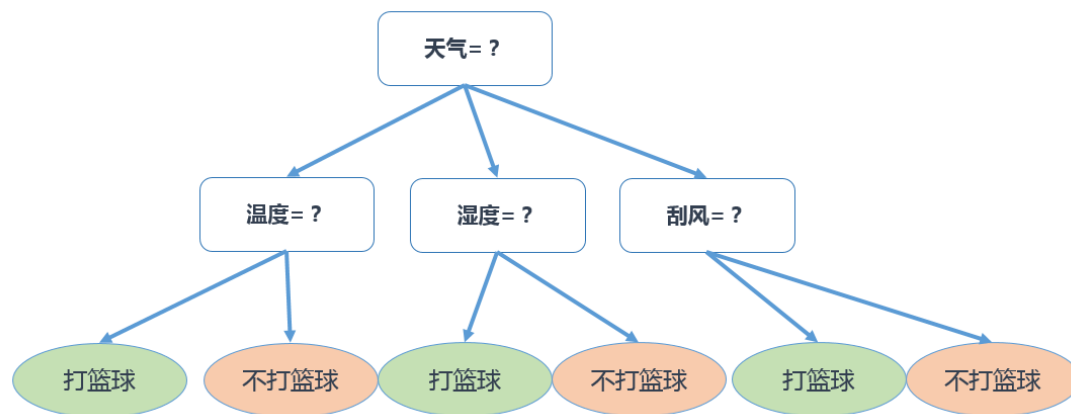
熵用来度量不确定性的，当熵越大， $X=x_i$ 的不确定性越大

对于机器学习中的分类问题，熵越大 => 这个类别的不确定性大

- 信息增益在决策树算法中是用来选择特征的指标，信息增益越大，则这个特征的选择性越好

信息增益 $g(D, A) = H(D) - H(D | A)$

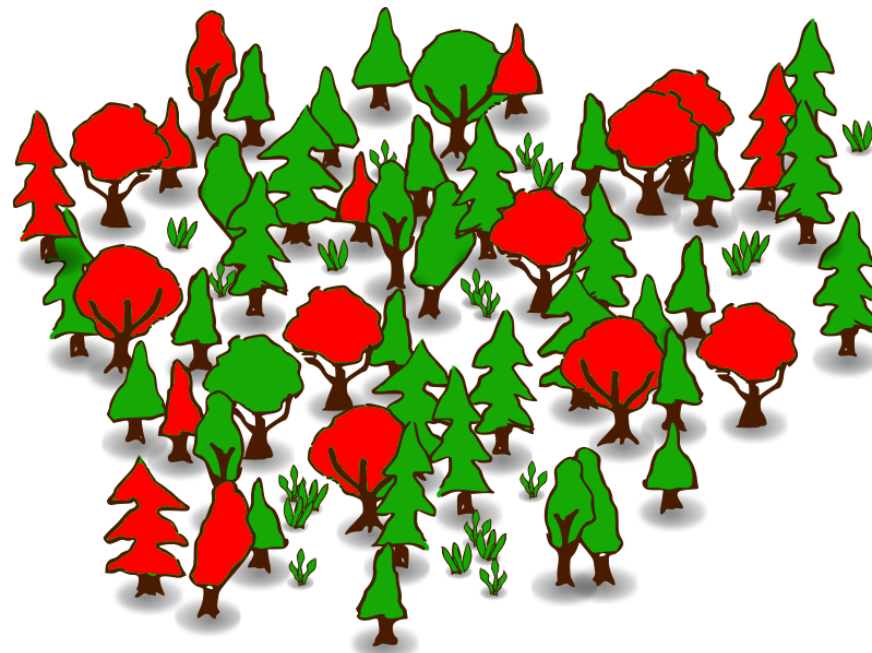
原有树的熵 $H(D)$ 增加了一个分裂节点，使得熵变成了 $H(D | A)$






决策树与随机森林

随机森林的生成：

- 森林中的每棵树都是独立的
- bagging思想，将若干个弱分类器的分类结果进行投票选择，从而组成一个强分类器
- bagging不用单棵决策树来做预测，增加了预测准确率，即不容易过拟合，减少了预测方差



少数服从多数，获得票数最多的类别
就是森林的分类结果



Thank You
Using data to solve problems