# ML workflow

## 挑战：

Formulate problem：关注在重要领域，如无人驾驶，无人商店

Data：高质量的数据很难获取，关于隐私的问题突出

Train models：模型越来越复杂，昂贵

Deploy models：在现实部署时高计算成本的模型可能并不适合

Monitor：数据分布改变，依据数据会出现不公平问题

## 人的作用：

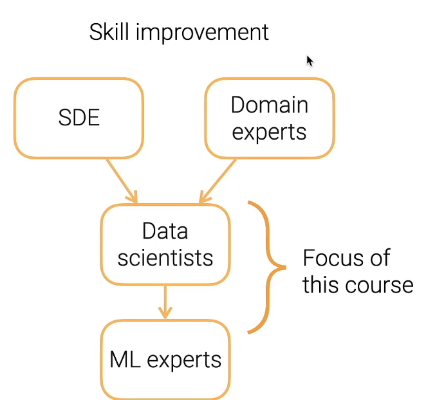
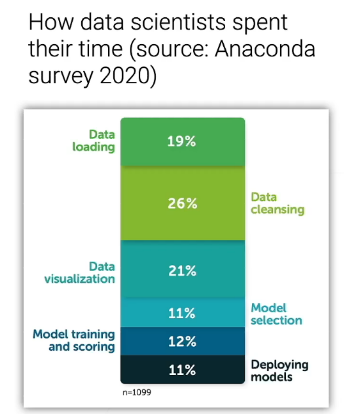
Domian expert：对产品和商业有很好理解，把模型部署到关键的地方

Data scientist：数据挖掘、训练模型、上线模型

ML expert：对SOTA模型做定制化

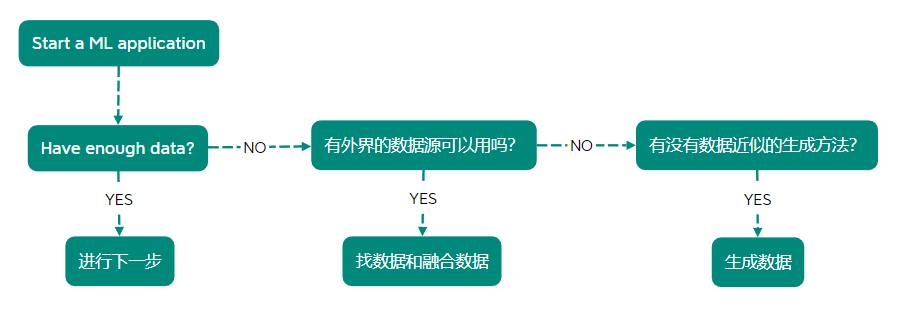
SDE：软件开发工程师，对数据应用、模型训练、模型部署全流程进行维护开发

## 如果我要持续提升自己的技能，需要做什么样的规划？



SDE和Domain expert对模型不需要很好的理解，但经过开发数据处理调参后，慢慢成为数据科学家，能够做整理数据、模型训练、上线。对于数据科学家，对模型进行大量改动后，慢慢有了自己理解创新，成为机器学习专家

## 关于数据获取：

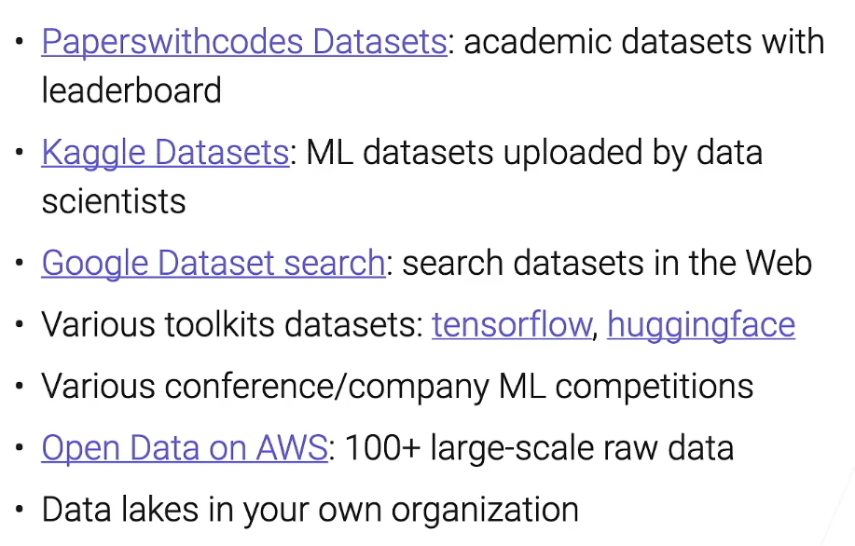


# 在哪里找数据集？

Paperswithcodes Datasets

Kaggle Datasets

Google Datasets search



## 数据融合：

把不同来源的数据合并成一个数据集

生产数据通常存储在不同的表内

Join table by keys（数据库函数），各种合并表的方法

## 生成合成的数据

-用GAN

-数据增强

-图片增广

-反翻译

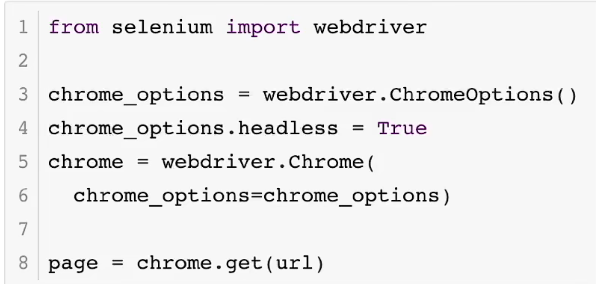
## 网页数据抓取：

数据科学家感兴趣的是在某个网页里的某项数据，数据抓取的结果可能是一张表

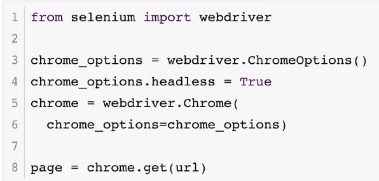
爬网页是把网页整个保存下来，匹配关键字，返回索引

Curl基本不可用

用headless browser：没有GUI的网页



可能需要不同的ip来做这些事

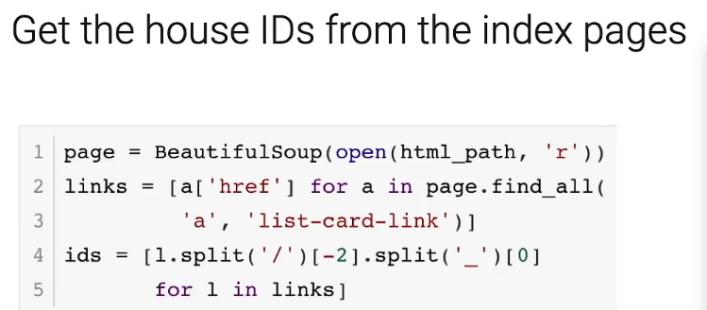


## 例如：房子交易信息

1.把网页全部放在刚刚代码内，保存为HTML

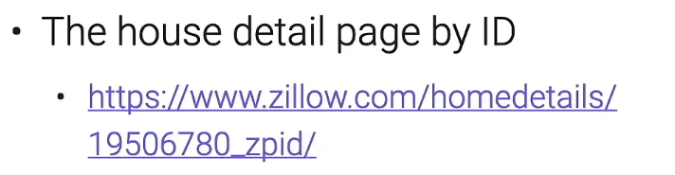


2.拿到具体房子的URL



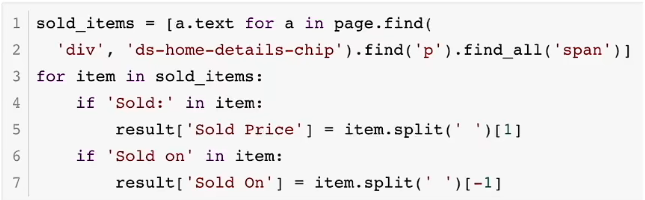
倒数第二个下划线和第一个\_切开

拿到id后，套入模板内



Id：19506780

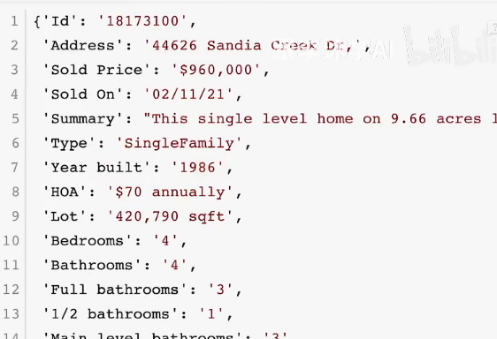
3.将网页上具体信息抓取下来



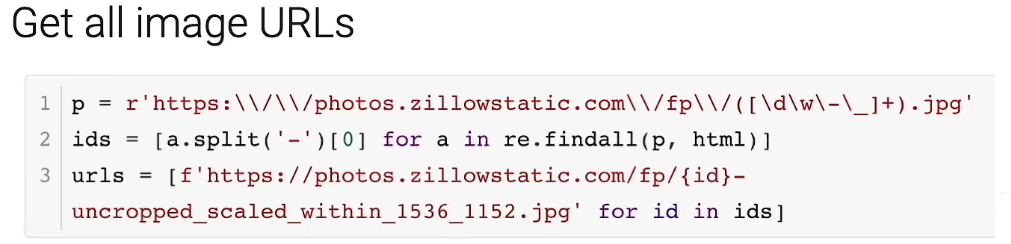
售价

成交价

把提取的信息存储成一个表格



## 如何把图片抓取



成本主要是在图片存储的开销

## 总结

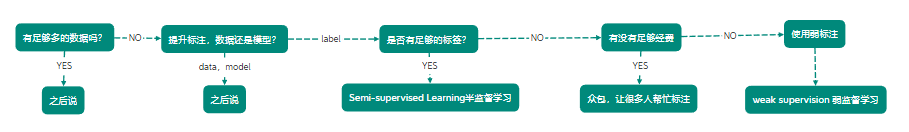
网页数据抓取是一个非常有用的手段去获取数据，当网页不给API的时候

如果使用云的话，成本比较低

使用浏览器工具查看HTML的数据存放在哪些地方

小心使用，避免法律问题

# 数据标注：



Semi-supervised Learning(SSL)

很小的一部分数据有标注（10%左右），其余数据没标注，如何把他们一起利用起来

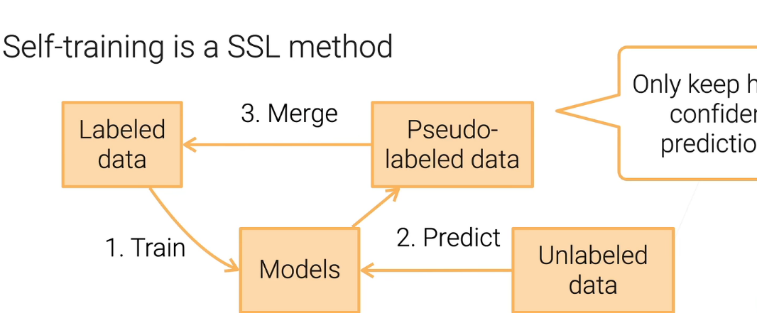
对未标注的数据和已标注的数据作以下假设：

-连续性假设(Contiunity assumption)：有相似特征的样本们很可能有相同标注

-聚类假设(Cluster assumption)：数据有内在的类结构，在一个类里面的数据倾向于有相同标注

-流形假设（Manifold assumption）:数据是在一个低维的流形上分布的，可以通过降维来获取一个干净的数据

Self-training



只有高置信的预测能留下来

标数据时，可以使用比较贵的模型

# 成本：

用人来标数据

设计好的方法让成本变低

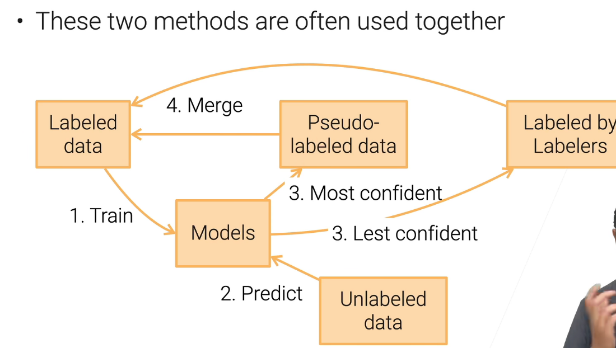
-使用交互页面（比如判断是或者不是）

-Active Learning 主动学习（和SSL很像但是有人为干预：将“有意思”的图片送给标注工去标注）

-选择特别不确信的图片出来（最高分类预测是1/n）

-quary by committee 多种模型投票觉得此图片较难

-主动学习+自训练



## 标注质量控制

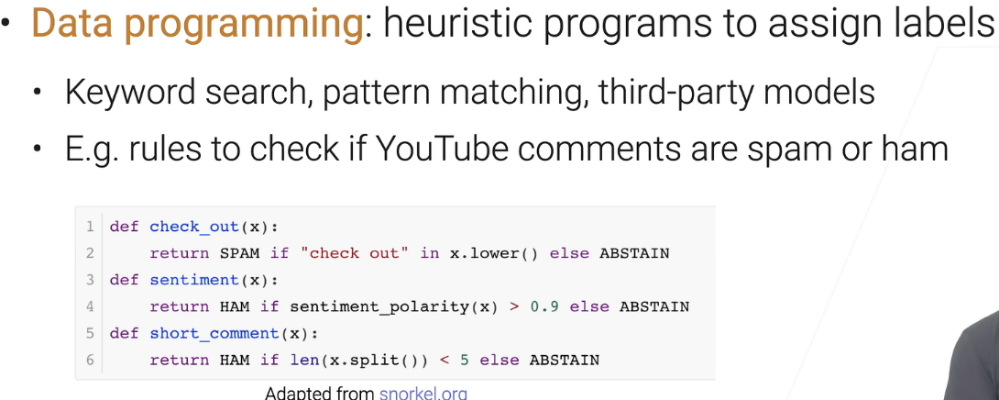
-标注工犯错或者没理解指令

-将任务发给不同的人，进行投票

## weak supervision 弱监督学习

半监督地生成标号

数据编程：



关键字搜索，模式匹配，第三方模型

## 总结

-拿到标号

-自训练

-众包

-数据编程

-无监督或自监督算法

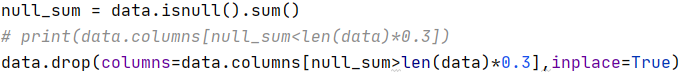
## 数据处理实例：

先读入爬取的数据





接下来，简化数据，将30%以上都是None的列丢掉





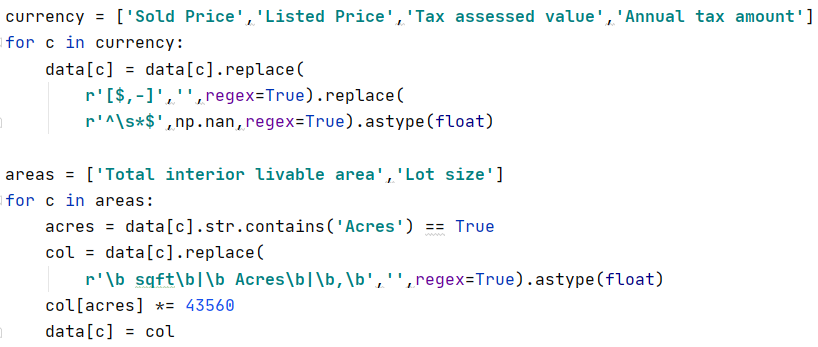
接下来查看数据类型



大部分为错误的数据类型

pandas不认识，需要进行数据转换，是数据清洗的一部分

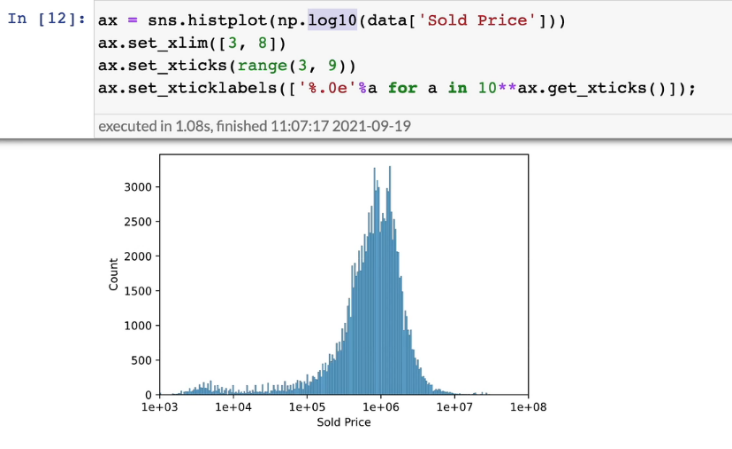
转化数据

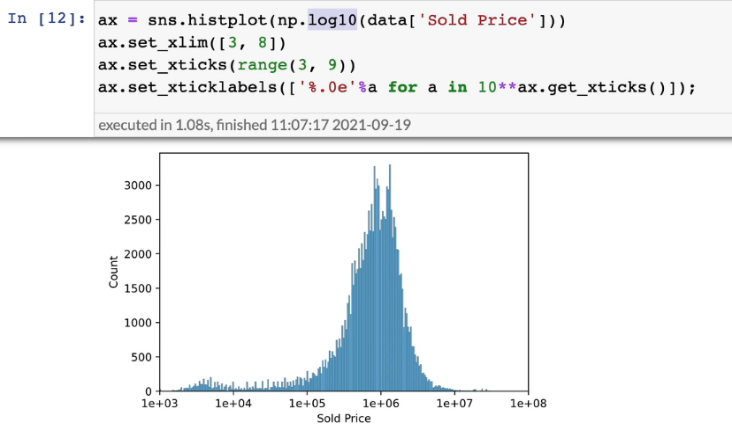


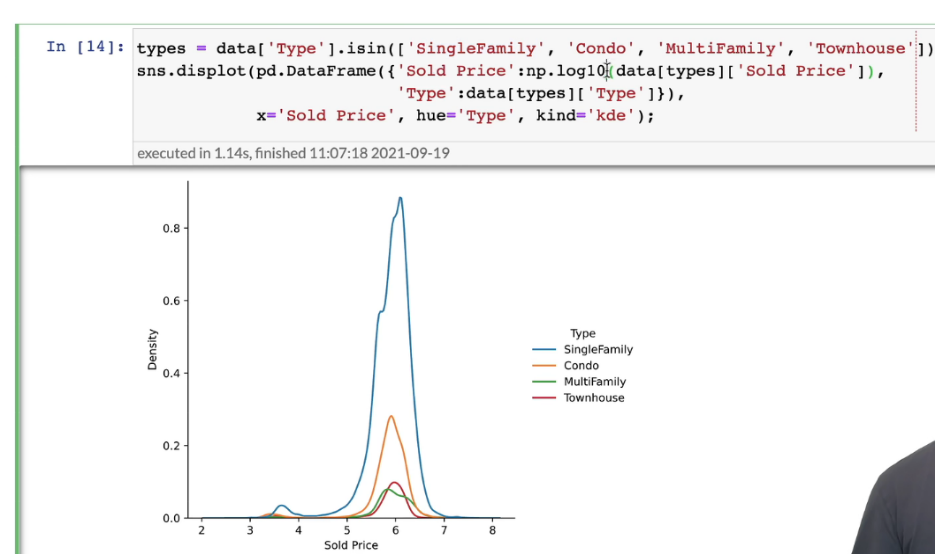
数据清洗，留下正常的数据

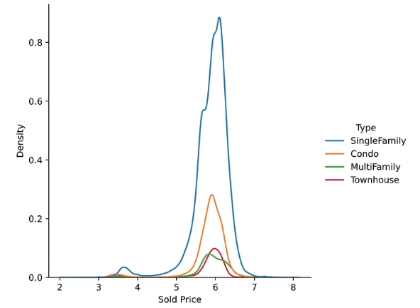


数据分析：

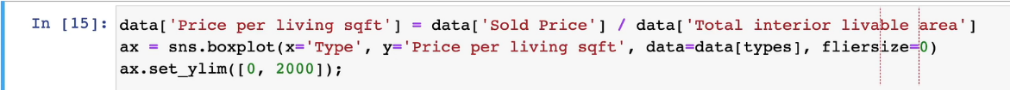


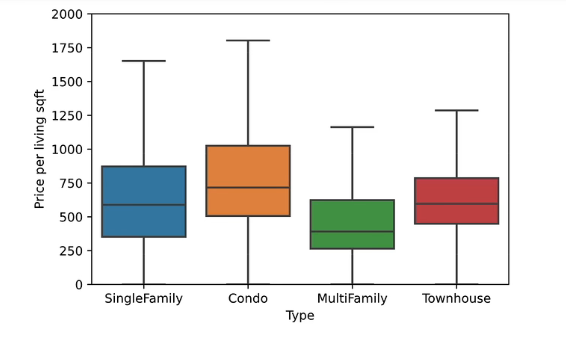
房子售价的分布



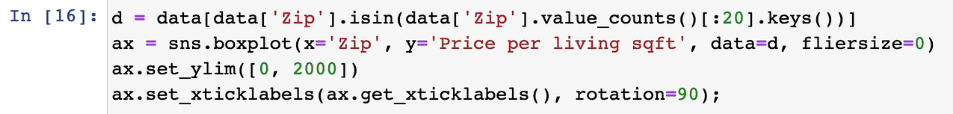


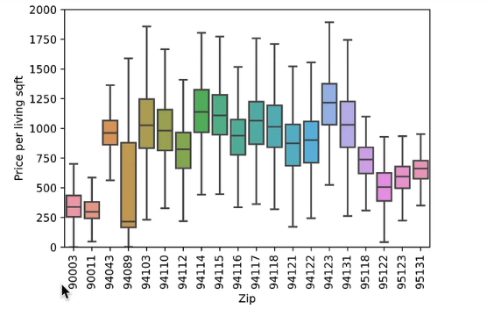
不同类型的房子售价的分布



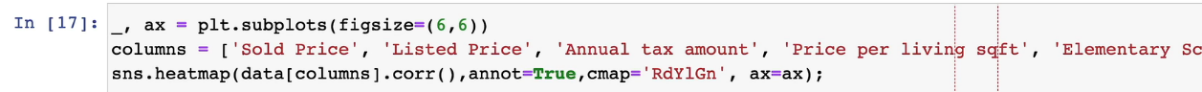


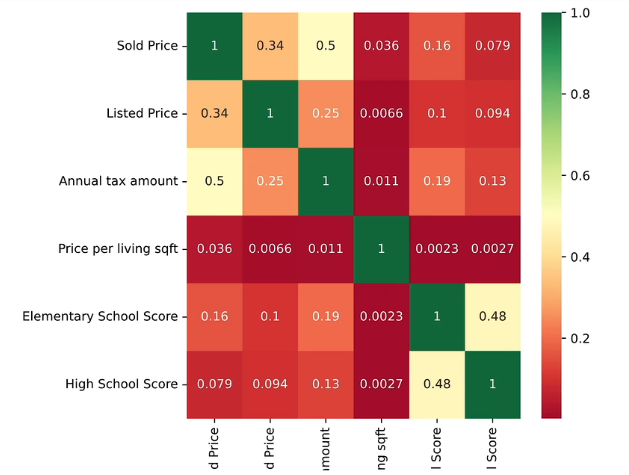
不同类型房子售价的箱图





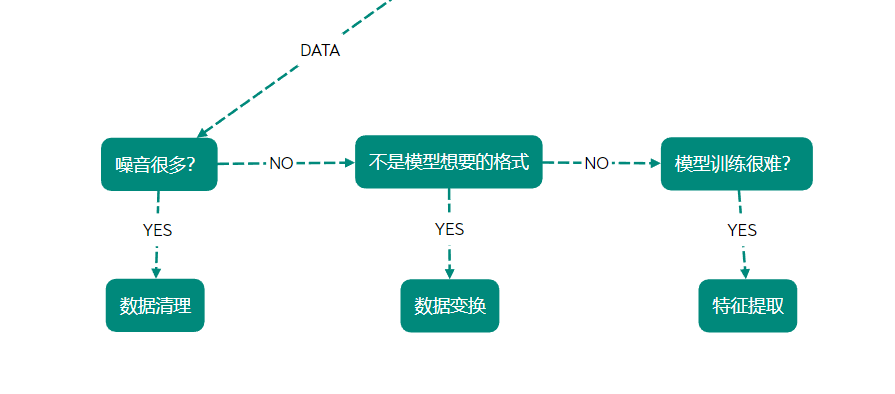
邮政编码对于房子价格的影响（地域）





不同标签之间的协方差（相互关系）

# 数据清理：



## 数据错误

-数据总会有错误，例如和事实不符

-好的机器学习模型对错误容忍度很高

-危害很高，实施错误的模型会影响新收集的数据

## 错误类型：

-Outliers（异常值）：偏离分布的一些值

-Rule violations（规则冲突）：不能有NAN或者必须是独一无二的

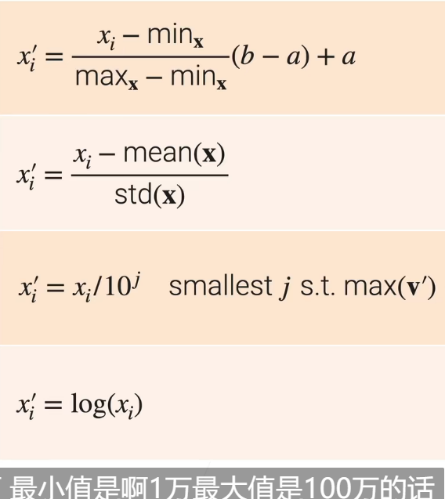
-Pattern violentions（模式冲突）：格式语义语法上有冲突

# 数据变换：

-数据被转化为适合ML算法的形式（包括了大部分数据预处理步骤，例：数据清理、特征抽取）

-集中注意于不同种类的数据的变换

对实值列的标准化手段：



均值为0方差为1

对数变换（基于相对值计算）

## 图片转换：

例如之前的房价预测，若是把美国每年卖的房子图片全部爬下来，估计有15TB，存储、传输、读取都是问题

所以要对图片进行下采样，转成jpg

-减少图片的大小节省存储和更快的加载

-ML对低分辨率的图片表现较好

-选择80%-90%的图片精度会有1%的精度损失

## 视频转换

平均视频长度

-电影2h、YouTube11分、tiktok15秒

我们通常会用10s左右的视频

-切成一段一段的小片段给机器学习模型

解码视频，采样成一系列的帧

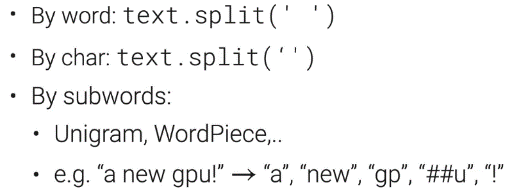
-计算比存储便宜（用GPU进行解码）

-权衡存储和解码的速度（若是存储成本高，就用好的压缩算法和好的GPU解码，反之亦然）

## 文本转换：

词根化和语法化-变成更平常的形式

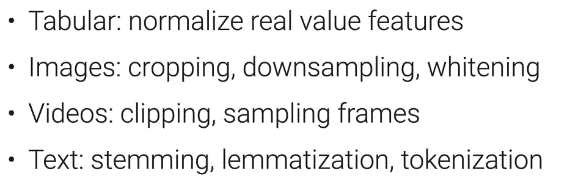
tokenization词元化：n-gram



## 总结

数据转换只是说把数据转换成适合ML算法的形式，不包括数据清理和特征提取，平衡数据大小、数据质量、读取速度

不同的领域有不同的方法

表格

# 特征工程：

-ML算法更喜欢固定长度的输入/输出

-在DL之前，FE是ML的关键

-在CV，人们尝试不同的FE然后训练SVM

-DL用机器去做FE

## Tabular Data Features

-int/float 直接用或者把最大值和最小值中划分n个区间，然后值落在区间内就赋值1，展开成长为n的一个向量

-类别数据：独热编码（虽然有n列，但是只有某一列元素为1，其余为0）

-将不常用的类别变成unknown

-Date-time：展开成一系列特征列（year、day、month、day\_of\_week .etc）

-特征结合：n列\*m列=n\*m列

## Text Features

-将文本转换成token features

-使用预训练的语言模型（GPT、BERT）

Image/Video Features

-手动抽取特征（SIFT）

-使用预训练的深度神经网络：将最后倒数第二层的输出（分类器之前）拿到特征向量（迁移学习）

-ResNet：在ImageNet上训练好的模型

-I3D：在kinetics上训练好的模型

# ML models overview

-Supervised Learning（监督学习）：

在有标号的数据上训练一个模型，去预测标号

Self-supervised Learning（自监督学习）：标号来自数据本身（word2vec、BERT、Transformer）

-Semi-supervised(半监督学习)：部分标好的数据训练模型，将没有标号的数据的标号预测出来

-Unsuoervised Learning（无监督学习）：在未标注的数据上进行训练（GAN）

-Reinforcement（强化学习）：用模型和环境交流得到的数据进行学习（机器人），不断和环境交互，最大化奖励

## Components in Supervised Training

-Model 输出预测值

-Loss 衡量模型预测值和真实值之间的差别

-Objective 优化目标（解决的问题）

-Optimization 学习参数

Type of Supervised Models

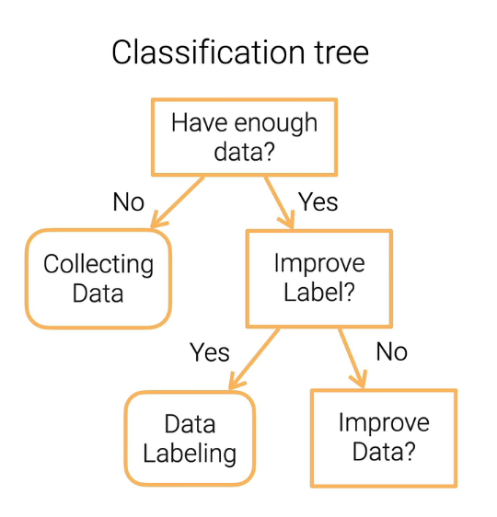
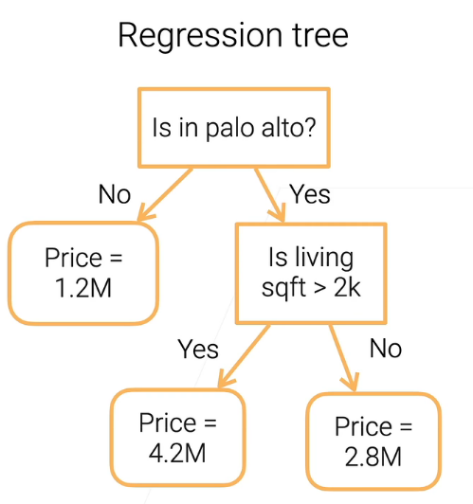
-Decision trees

-Linear methods:决策是输入的线性组合

-Kernel machines：用一个核函数来衡量两个样本之间的相似度

-Neural Networks：使用神经网络学习参数

# Decision Trees（决策树）



-好处

可解释

可以处理数值类和类别特征

-坏处

非常不稳定，对数据噪音敏感，没有鲁棒性（集成学习ensemble to help）

复杂的数据导致过拟合（剪枝解决）

不好并行处理

-树模型大量应用于工业：简单、好调参、通常给予满意的结果

## Random Forst随机森林（将树变得稳定）

-训练多个决策树来提升自己的稳定性

-每棵树独立地进行训练

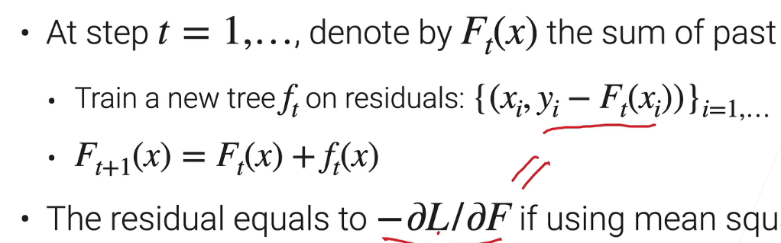
-共同决定（投票、取平均）

-随机来自于哪里？

-Bagging：随机采样训练样本

-随机选择特征列

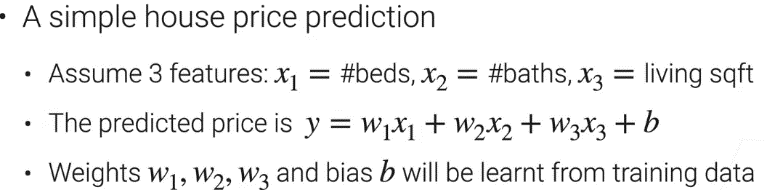
Gradient Boosting Decision Trees

-训练多棵树（顺序地完成）

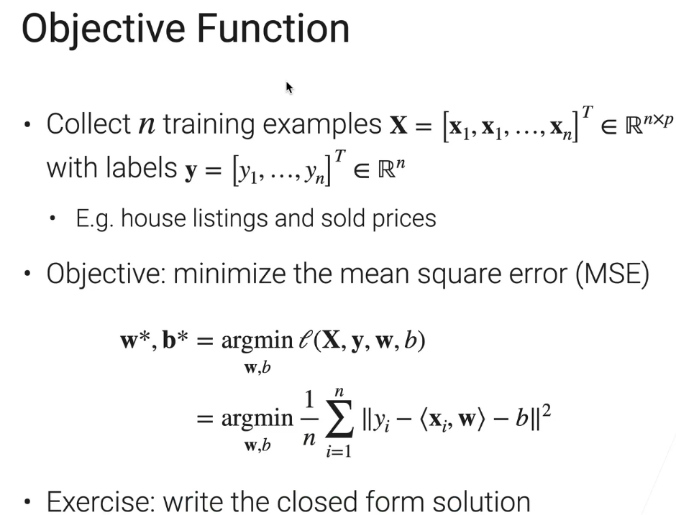
-将之前的树结果加起来，和真实值作差，用这个差再训练新的树（残差Residual）

-这个残差等于预测值和真实值的均方误差作梯度(训练一个新的树去拟合梯度的负数)

# Linear Regression



## Object Function



优化平均均方误差

## 用线性模型做分类：

回归的输出是一段连续的实数，而分类是输出对样本类别的预测；

在这个部分，我们所关心的是多类的分类问题

可以使用向量来输出（不是输出1个元素而是m个元素 m为类别数）；

使用线性模型预测出样本数据类别的置信度，最大置信度的类别为样本数据所对于的类别并用onehot（独热）编码输出。

这里的目标函数是MSE（均方误差）

使用均方误差（MSE）作为目标函数，使得预测值趋近真实值，但是作为分类关心的是数据对应类别的置信度

让模型更加专注到把正确的类别弄出来：

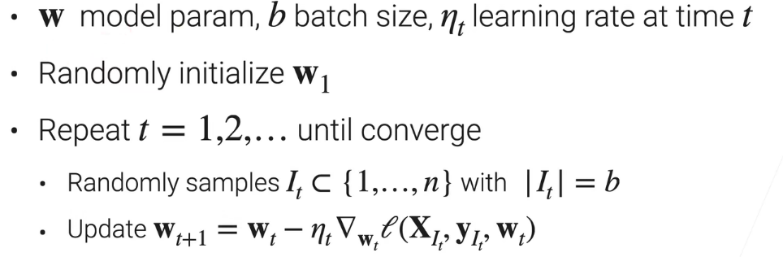
具体来说：把预测的分数换成概率的形式（Softmax函数）；

衡量真实值概率与预测值概率的区别，用Cross-entropy(交叉熵)

用Softmax回归就好

# Mini-batch Stochastic gradient descent（SGD）

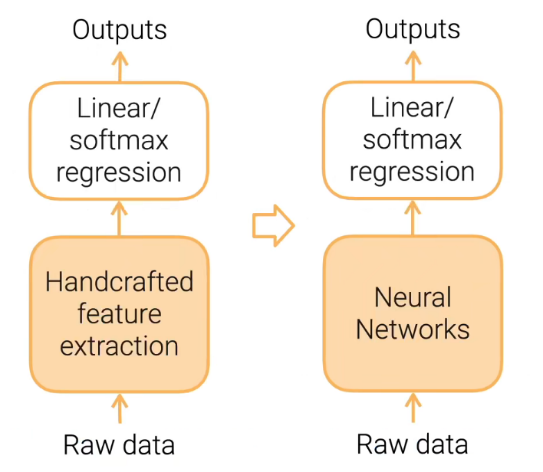
-通过Mini-batch SGD求解

把偏移写进模型内（在输入向量内加入全1的特征）

-好处：可以解决大部分机器学习问题

-坏处：对超参数敏感，需要调参

# Neural Networks



NN通常需要更多的数据和计算

NN结构对模型数据结构

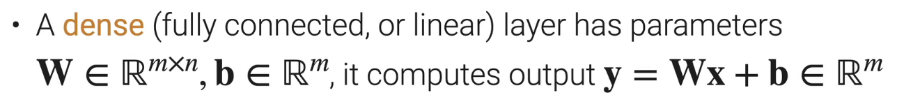
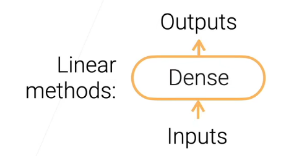
-多层感知机

-卷积神经网络

-循环神经网络

--Transformers

## Linear Method -> Multilayer Perception(MLP)



-线性回归：全连接层输出维度为1

-softmax 回归：全连接层+softmax

## MLP

-需要增加激活函数（Sigmoid，ReLU），避免变成线性的叠加

-堆叠多个隐藏层（中间添加激活函数）

-超参数：隐藏层大小，个数