Chapter2.End-to-End Machine Learning Project 第二章.端对端的机器学习项目

接下来，我们要进行一个端对端的项目项目实战，假设你已经被一家房地产公司雇用成为一个数据专家。以下是主要的步骤：

1. Look at the big picture. 了解大概
2. Get the data. 获取数据
3. Discover and visualize the data to gain insights. 发掘和可视化数据，增加理解
4. Prepare the data for Machine Learning algorithms.准备数据
5. Select a model and train it.选择一个模型，开始训练
6. Fine-tune your model.调试模型
7. Present your solution.展示解决方案
8. Launch,monitor,and maintain your system.发布，管理，维护系统

Working with Real Data 处理真实数据

学习ML的时候，最好使用的是真实数据。以下是可以获取真实数据的网站：

最受欢迎的数据仓库：

<http://archive.ics.uci.edu/ml/index.php>

<https://www.kaggle.com/datasets>

<https://registry.opendata.aws/>

数据仓库门户网站：

<http://dataportals.org/>

[https://opendatamonitor.eu/frontend/web/index.php?r=dashboard%2Findex](https://opendatamonitor.eu/frontend/web/index.php?r=dashboard/index)

<https://www.quandl.com/>

<https://en.wikipedia.org/wiki/List_of_datasets_for_machine_learning_research>

<https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public>

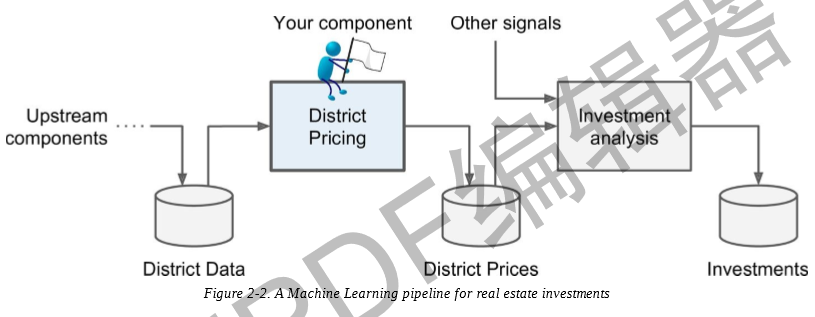
<https://www.reddit.com/r/datasets>

正式开始

Frame the Problem 框架问题

首要问题是问你的老板具体的商业目标；建立一个模型通常不是最终的目的。公司期望从模型中获得哪些收益？这些决定了你怎么框架问题，选择什么算法，你将使用什么性能评估工具评价你的模型，你将会花费多少精力来调试。

你的老板回复你的模型的输出（房价的中位数）会和其他一些信息输入给另一个ML系统。用来判断一个区域的是否值得投资。这一点非常重要，直接关系到收益。



*Pipelines 流水线*

*一系列的数据处理组件，被称为数据流水线。流水线在ML系统里面非常常见，因为有很多数据需要处理、转换后才能使用。*

接下来的问题是目前的解决方案是怎么样的。这通常会给你一个参考，来解决这个问题。老板会说目前是通过专家评估的，一帮人聚在一起通过最新消息，通过一些复杂的规则来得出一个评估结果。这既昂贵又耗时，而且他们的评估并不准确，通常会有15%的错误率。

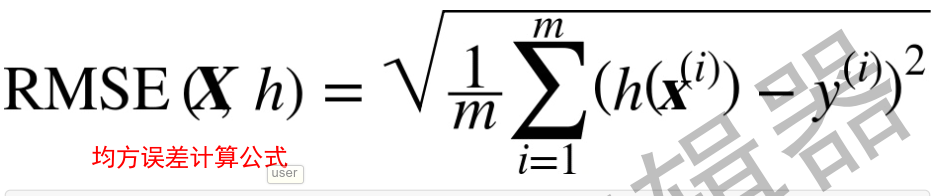
现在，有了这些信息，你准备好开始设计你的系统了。首先，你需要确定问题框架：监督，非监督，增强学习？分类任务，回归任务，或者其他的？批量学习还是在线学习？在你继续之前，暂停一下，尝试自己回答这些问题。

你找到答案了吗？我们一起看看：这明显是一个典型的监督学习任务，因为提供了有标签的训练例子（每一个实例都有一个期望输出，这块区域房价的中位数）。更进一步，这也是一个典型的回归任务，因为你被要求作出一个预测。并且，这是一个多元回归的问题，因为需要使用多个特征去做预测（通过人口，收入中位数等）。最后，没有持续输入的数据流到系统，没有特别需要快速适应变化的数据，而且数据足够小，内存可以满足，因此，批量学习是比较合适的。

提示：如果数据很大，你要么需要把批量学习任务分割在多个服务器上运行，或者使用线上学习技术。

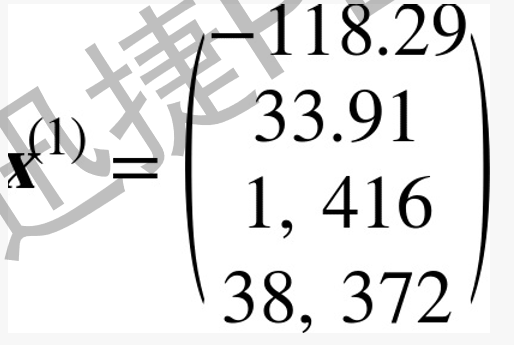
**Select a Performance Measure**

下一步任务就是选择一个性能评估方法。对于回归问题，典型的性能工具是Root Mean Square Error(RMSE).通过测量系统的错误标准差。例如，rmse等于50,000意味着系统中68%的预测落在实际值50,000美元以内，95%的预测落在实际值100,000美元以内。

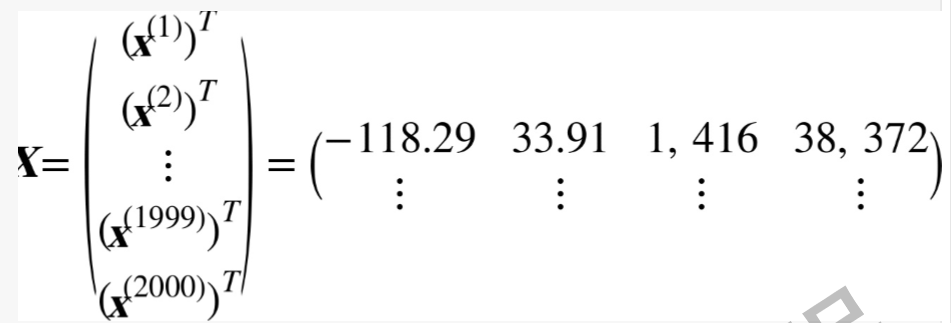


* m 是要测量实例的数量
* x(i)是数据集所有特征的向量，y(i)是带标签的实例

例如，如果区域坐标是东经-118.29，纬度33.91，当地有1416户居民，中位数收入是38372美元，中等房价是156400美元，那么：

和

* X是所有实例特征矩阵（包括标签），每一行是一个实例。

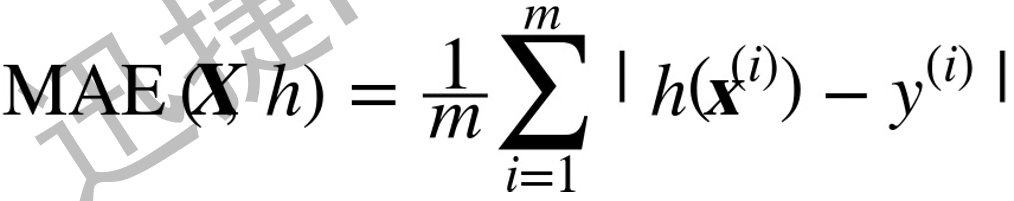


* h是系统预测功能，也被称作假设。当你的系统被提供一个实例的特征向量x(i),它输出的预测值ŷ(i) =h(x(i)).

例如，如果系统预测第一个区域的中位数房价是158400， ŷ(1) =h(x(1)) =158,400。那么这块区域的预测错误值是ŷ(1) –y(1)=2,000.

* RMSE(X,h)是实例集测量的损失函数。

即使RMSE通常被用于测量回归任务的性能，在某些情景下你更喜欢用另外一个函数。例如，假设，有很多异常地区（情况，结合上面房地产的例子）。那种情况下，你可能考虑用Mean Absolute Error(也被称作Average Absolute Deviation 平均绝对偏差)



RMSE和MAE都是用来测试两个向量之间的距离：预测向量和目标向量。可能是各种各样的距离测量或者是范数：

* 计算RMSE相当于计算欧几里得范数：就是计算距离。记为 ℓ2，||\*||2
* 计算MAE相当于计算 ℓ1范数。||\*||1，有时候也被称作曼哈顿范数
* 范数公式：



* 范数级数越高，越会聚焦大数字而忽略小数字，这就是为什么RSME相比较MAE更加敏感。当异常值很少的时候，例如钟形曲线，RMSE就表现更好。

**搭建环境**

**Python, NumPy,Pandas,Matplotlib,and Scikit-Learn.**

**开始编码**