Lecture6

红色标注的语句，为重点。
蓝色下划线标注的语句，说明给出了参考阅读链接，可依兴趣阅读。

写在Lecture6之前

转眼间，我们的Python普及分享会进展已经过半，在后半部分的分享中，我们主要关注使用Python进行机器学习、深度学习。

按照原计划，机器学习部分打算安排两个Lecture，一部分普及常见机器学习算法，另一部分关注sklearn模块的基本使用。但经过这一段时间的交流，发现大家对于机器学习的常见算法还是比较了解的（也有很多和我一样的调包侠）；而对于不是很了解的同学，这些知识在互联网上可以很方便地获得，而且互联网上的知识质量很多是显著高于我能提供的（笑）。

因此，我打算只用Lecture6将机器学习的基本概念、sklearn的使用方法做一个简单的扫盲，并提供进一步学习的一个路线图和资料。

对于Lecture7，我计划插入一部分网络爬虫的延伸内容，可能会考虑app端抓包爬虫，或者将爬虫部署到服务器等内容，具体安排未定，也要看大家的意见。

1. 机器学习解决什么样的问题

1.1 机器学习的主要类别

机器学习的划分方式不一，主要有有监督学习、无监督学习、半监督学习、强化学习等几个大类。

有监督学习指，已知数据和其一一对应的标签（例如每条数据代表一个猫或狗的各个特征值，并告诉了算法每条数据究竟是猫还是狗），训练一个算法，智能地将一个新的输入数据映射到一个标签的过程。

无监督学习，并不知道数据的标签，希望按照一定的指定偏好，训练一个算法，智能地将所有数据映射到不同的若干标签的过程。例如，仅仅提供猫或狗的数据，并不告诉程序哪一条数据是猫或狗，程序也许会根据耳朵、鼻子的不同来将原始数据划分为泾渭分明的若干类，每个类均为猫/狗占绝大部分。

半监督学习是前两种的综合，即有一部分数据标签已知，另一部分标签未知的学习任务。半监督学习关注的核心就是该如何利用标签未知的数据，尤其是关注标签未知数据的分布。

强化学习类似各位玩游戏的过程，通过设定规则、奖惩机制，程序会通过不断地试错，来提升任务性能。一个经典的例子是通过强化学习，来玩很久以前很经典的flappy bird游戏，程序通过大量的死亡后，会逐渐掌握该采用何种方式让小鸟顺利通过水管。

在金融领域，我们主要关注有监督学习和无监督学习。

1.2 分类、聚类、回归、降维

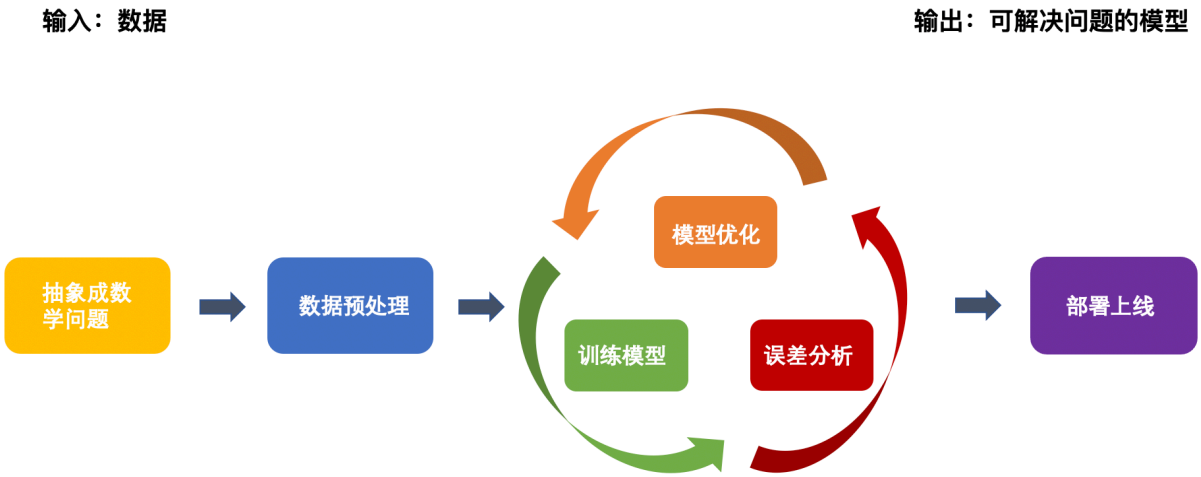
分类、聚类、回归、降维是有监督学习和无监督学习的主要表现形式

	分类	聚类	回归	降维
类别	有监督学习	无监督学习	有监督学习	无监督学习
含义	已知有若干类，将给定样本分入对应类别	不知道总体中每个样本的类别，将总体样本划分成若干组内相似，组外相异的类	用函数去拟合点集，这样对于一个新样本，能给出连续的估计值	减少高维数据维度，尽可能保留更多信息，但缩小体积
经典例子	客户流失预警（二分类），拍照识花（多分类）	用户画像细分，自动生成多个用户群体	预测股价	将高维数据可视化
常用算法	logistic回归 决策树-分类树 深度学习 支持向量机 朴素贝叶斯	K均值聚类 层次聚类 DBSCAN聚类	线性回归 决策树-回归树 深度学习	PCA主成分分解 特征选择 非负矩阵分解

结合第三行的例子，相信大家可以初步理解这四大机器学习任务的主要内容。

2. 一个机器学习项目的流程

对于一个机器学习项目，将其当作一个黑箱的话，其输入为数据，输出则为一个可以解决问题的模型（如分类模型）。因此，可以这样理解机器学习的过程：通过对一部分数据内部规律的探寻，将规律转化成计算机可以理解的模型，来外延解决类似数据的问题。



Lecture6

对于一个常见的机器学习过程，主要有以下四步：

抽象成数学问题：在一切开始前，我们需要将实际业务、实际学术中的问题进行数学抽象，询问自己，我们是想通过数据获得一个分类模型，抑或是回归等其他模型。

数据预处理：我们获取的原始数据，往往要经历以下分析/处理流程：

——数据清洗、整理：原始数据往往杂乱无章，缺失、异常值也很多。在进行下一步动作前，我们需要对其进行整理、查缺补漏、删除异常等操作。

——特征工程：清洗整理完毕后，我们的数据已经变得“好看”，但我们还要对数据做精加工处理，使其变得“好用”。这一步叫做特征工程，主要包括计算新的特征、筛选可用的特征等工作。

数据预处理这一步骤非常重要，在一个大型的机器学习项目中，花在这一步的时间往往超过40%。可见其对结果的影响之大。

Ps：

要记住的是，数据预处理并不能改变数据内在的固有质量，*Garbage in, garbage out*是一句著名的定律，对于原本就包含错误信息，或信息含量低的数据，数据预处理过程并不能使其变得能用。任何算法性能都是由数据的内在质量决定上限的，我们仅仅能通过数据预处理，使其逼近这个上限，但无法突破由垃圾数据决定的极低上限。

训练和调试：

这一步，是机器学习中看起来最“高大上”，但实际操作很简单，然而又很费事费神的一步。

我们需要针对数据的质量、类型、大小评估哪些算法在性能、运算开销上适合本次任务。这需要丰富的经验，例如，支持向量机在处理多特征的数据时会有一定的性能优势；而对很多看上去复杂的问题，其实logistic回归就已经能很好；XGBoost在各大竞赛中所向披靡，但其搜索过程是完全遍历，堪称内存黑洞……

在选取了模型后，只需要借助sklearn的几行代码，就可以使其在你的机器上以原生C的速度飞快地训练出一个可用的模型。训练的过程，本质上是从数据里挖掘信息，拟合出该模型最适合这批数据的表达形式。我们会从原有数据集中，挑出一部分（往往是一大部分）作为训练集（train），来让模型进行训练。有些模型的训练是迭代式，主要的训练过程就是不断减小误差值（损失函数），例如诸多使用随机梯度下降的算法；而部分模型则不需要这个迭代过程，如贝叶斯相关的各类模型，以及谱聚类等聚类算法。

模型训练完毕后，我们就可以用其来执行任务了，如进行一些分类等。但此时我们还不知道其效果究竟如何，为了解决这一问题，我们会从原始数据集中，不属于训练集（train）的部分中挑出一部分做验证集（validation），测试模型在这一部分没见过的数据集上的效果究竟如何。

通常情况下，第一次训练出来的模型往往效果不好，这时就需要我们进行复盘。分析误差的由来，究竟是来自数据还是来自我们的算法，如果是前者需要回滚到特征工程一步，如果是后者需要调试算法中的各项参数。这是一个痛苦的过程，在深度学习中，这一步往往被称为“炼丹”。

部署上线：

模型在经过若干轮迭代，正式确定后还要经历最后一步，即从训练集（train）和验证集（validation）之外再从数据集中选取剩下的最后一部分，做测试集（test），来评定模型的最终效果。

如果模型的最终效果是令人满意的，就可以进行固化、导出等操作了，这时候，模型已经成为了一个可以重复使用的“制成品”，可以投入到生产、学术应用中，供他人轻松使用。训练模型可能很耗时，但使用模型是很迅速的。

以上即为常见的机器学习过程。请注意区分在机器学习过程中，训练集（train）、验证集（validation）、测试集（test）等的作用。

3.常见的机器学习算法

这里列举几类常用的机器学习算法：线性模型类（线性回归、岭回归、Logistic回归）、决策树类（C4.5、CART）、神经网络类（CNN、RNN、LSTM）、支持向量机、贝叶斯类（朴素贝叶斯、贝叶斯网络）、集成学习（bagging与boosting）、聚类算法（k-means、层次聚类、DBSCAN）

指望一下子全理解这些算法显然不现实（显然，我也没有全理解，所以更不敢乱向大家讲这些算法），尤其是在某些算法完全理解还比较困难的基础上，如Logistic回归和LSTM。对于接触暂时较少的朋友，这里列两个层次的参考资料供大家学习：

Level1:（当作看小说）

对于仅仅想知道这些算法在干啥，优缺点何在，会用就行的朋友，我强烈推荐sklearn包的User Guide。虽然它是为了教用户如何使用这个包，但它考虑到了现在机器学习的低门槛（真实），从最实用最基础的角度介绍了各类算法，并且当场提供可供操作的代码，图文并茂，读起来非常易于理解，链接如下：

https://scikit-learn.org/stable/user_guide.html

Level2:（中期目标）

对于数学功底不错，想了解算法的来龙去脉、各种变种的推导，较深刻理解不同算法的朋友，南大周志华老师的西瓜书《机器学习》是大家都知道的教材。但我并不建议大家一上来就看西瓜书，可能会被比较多的数学公式弄昏（因为我数学比较菜）。

南大老师的书自然要买一本支持一下，就不贴链接了。建议学习第1-11章内容，即到“特征选择与稀疏学习”为止。

建议配合“南瓜书”来学习，“南瓜书”解释了一些周老师受限篇幅没有展开的推导过程：

https://datawhalechina.github.io/pumpkin-book/#/

综合了一下多方意见，我其实更推荐李航的《统计学习方法》，其内容编排更加细致，很有中国教材的风范。内容可能相对周老师的要陈旧些，但经典永流传。

Level3:（抢CS专业饭碗）

如果你想更进一步，我列出一些我尝试看过但看不懂（今后也不打算看），向上进阶必须要掌握的书籍：

Pattern Recognition and Machine Learning by Christopher Bishop
Machine Learning: A Probabilistic Perspective by Kevin P. Murphy

如果你已经掌握了这两本书，我愿称你为最强。

4. sklearn操作

这里列出两个学习的层次建议：

Level1:

了解sklearn的逻辑，建议使用官方的Getting Started：

https://scikit-learn.org/stable/getting_started.html

Level2:

没错，还是看User Guide，因为很长，所以建议等到要用的时候再看特定需要的算法。

https://scikit-learn.org/stable/user_guide.html

5.个人建议

关于机器学习部分，我还是建议大家根据日常实际需求，做有针对性的学习，没有必要“完全吃透”，那可能是计算机专业博士生的工作。因为如果日常工作用不到机器学习的朋友，将来估计也用不到，真要用上了也肯定是小应用，临时学习可能更能解决问题，并且在现在的大环境下，机器学习越来越被封装化，更像是一个工具，而不是“看门本领”；而对于日常工作经常用到机器学习的朋友，早就已经在各种业务场景里使用过各种工具了，相信也已经有了自己的理解，不需要这么简单入门的一份指引说明。

因此，Lecture6注定是一个矛盾的存在：了解的同学早就了解了，暂时没基础的同学也不可能一下子就补齐，这也是我前期规划时的漏洞，没有考虑到这一点，在这里向大家致歉～