告营销等等。

在这个步骤结束后，通常会生成一个项目章程，主要包括以下内容

* 明确的研究目标
* 项目人物和背景
* 如何进行分析
* 期望使用什么数据
* 可交付成果和确保成功的措施
* 时间表

该项目章程也不仅适用于数据科学项目，也适用于其他诸如游戏制作的项目。

(b) 检索数据

数据来源可以是从公司内部得到的数据，也可以是外部数据，虽然如今普遍认为数据是比石油更有价值的资产，但是已经有很多组织开始免费地提供数据了，比如 Data.gov 就是一个美国政府的开放数据库。

在这里有几个概念，数据库的主要目的是存储数据，数据仓库则是用来读取和分析数据的，数据集市是数据仓库的一个子集，为特定的部门提供服务，而数据湖则充满了自然或原始格式的数据。

(c) 数据准备

该步骤主要包括数据清洗、数据转换和数据整合。垃圾输入导致垃圾输出，这个步骤十分重要通常，且会反复进行，比如在数据建模阶段会发现错误，进而返回该步骤。

数据清洗的主要目的是消除数据中的错误，包括数据输入过程中的错误、冗余的空白、不可能的值、缺失值以及异常值等。

数据转换的主要目的是使数据的格式满足模型的要求，比如将两点间的距离转换为横纵坐标之间的关系，或者将变量转换为虚拟变量。虚拟变量用于表明观测值的分类结果，如果属于某种类别，就置“1”，否则置“0”。

数据整合主要指的是从不同的数据源整合数据，用到的一些手段有连接表、表添加、使用视图、丰富数据度等等。

(d) 数据探索

数据探索阶段主要用到的是数据可视化技术，因为以图片的方式来展示数据和变量之间的交互更容易理解。经常会用到的图形包括简单的线图或直方图、以及更为复杂的箱线图、桑基图等等。但有时将几个简单的图形组成一个复杂的图形更容易理解，或者多个图形关联更新等。当然也有其他的数据探索方法，比如制表、聚类或简单的建模。

(e) 数据建模

在该阶段主要步骤如下：

i. 特征工程和模型选择

ii. 模型训练

iii. 模型的验证与选择

iv. 经过训练的模型应用在未知数据上

该阶段用到的技术包括机器学习、数据挖掘和统计领域等。

(f) 展示和自动化

该阶段你将主要将自己的成果展示给其他人并反复解释结果的意义，所以这就要求一定的相关领域专业知识。而且，如今各大公司对人工智能算法工程师的需求大大降低，更多地是在寻找和业务需求相匹配的工程师，所以将相关的专业知识和机器学习相结合愈加重要。另外，成果模型可能会部署到实际的应用当中，所以可能会需要模型自动更新报告，即自动化。

4. 机器学习在数据科学中的应用

机器学习在数据科学中的应用，主要有三个方向：

（1）回归：即预测，比如基于现有的数据预测之后的股价从而辅助交易分析员做出决策

（2）分类：比如在文本中识别人名和地名、图像识别等等

（3）根本原因分析：当业务的目标是更深入地理解一个现象的基本过程时，常常建立一个模型来完成此任务，比如理解和优化业务流程。

机器学习的类型：

（1）有监督学习

通常应用于有标注的数据集，需要人机交互来对数据进行标注。比如识别图像中的数字。

（2）无监督学习

不依赖于有标注的数据集，尝试在没有人机交互的情况下标注数据。

（3）半监督学习

需要有标注的数据集，也需要人机交互来对数据进行标注，但是通常从标注很少的数据集开始。

接下来将详细介绍机器学习在建模阶段的应用

i. 特征工程和模型选择

进行特征工程，必须为模型选取并创造出可能的预测因子。模型最终将根据预测因子实现预测功能，在这里，介绍一下交互变量，任何一个单一的变量影响很小，但是如果两个变量同时存在，那么它们的影响就会很大。就像化学上的醋和漂白剂，单独用没什么影响，但混合的话，就会产生有毒的氯气。

除此之外，还有易得性偏差：如果选取的特征只是那些容易获得的特征，那么构建的模型就会呈现出片面的倾向。在这一阶段，通常用到的方法有 PCA 主成分分析法等等。

ii. 模型训练

在这一阶段，将清洗好的数据数据输入模型。

iii. 模型的验证

机器学习中有两种常见的误差衡量：

分类错误率（针对分类问题）；

均方误差（针对回归问题）；

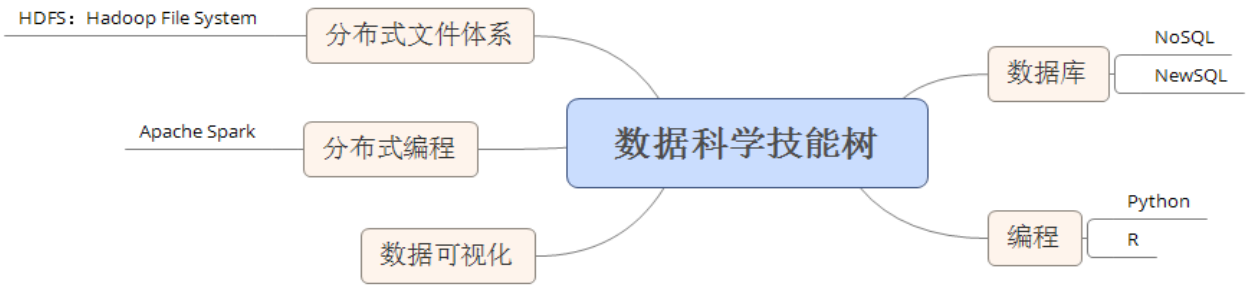
在这一阶段有很多验证策略，比如：

将观测值的 X% 划分为训练数据集，其余的部分作为测试数据。

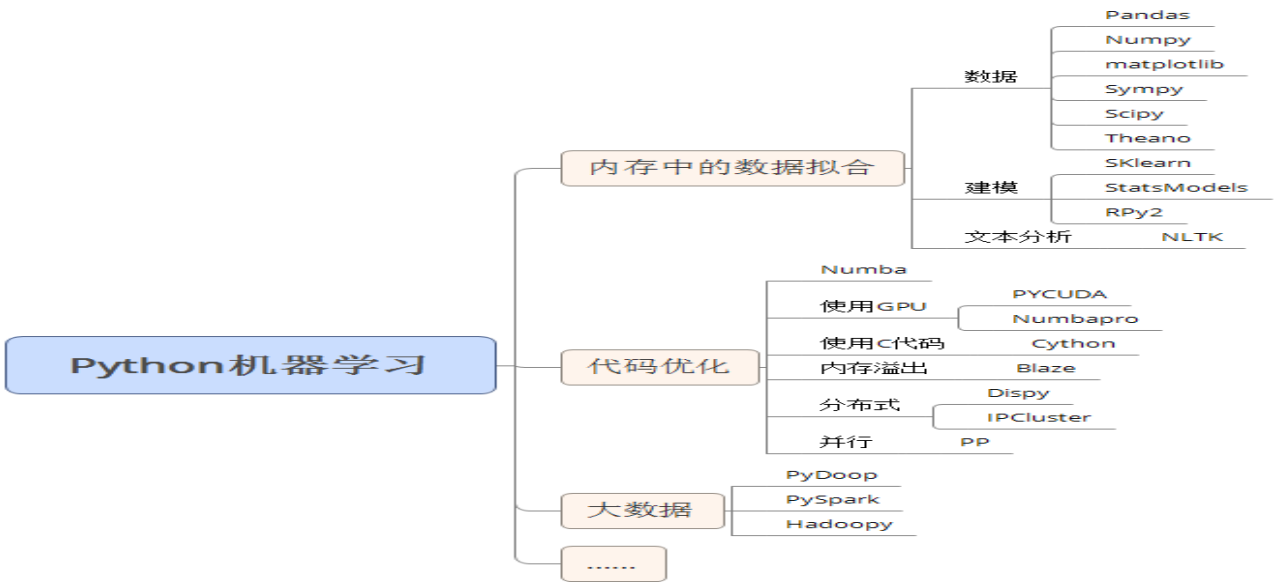
K 交叉验证——将数据集分为 K 个部分，每一部分都作为测试集，同时其他的部分作为训练数据集。这种方法的优势是数据集中所有的数据都是可用的。

iv. 经过训练的模型应用在未知数据上

5. 数据科学技能树



（P 3-6-1）数据科学技能树



（P 3-6-2）python 在数据科学中的常用库

科技入门一本通之游戏开发

一、主流游戏引擎

1.Unity3D

Unity3D 对于游戏开发者们来说是一个真正可以负担的起的引擎，具有其他引擎难以匹敌的用户量。更为重要的是，你只需要付费一次，而且，不管你的游戏如何成功，都不用担心 Unity 会分走你的收入。这对于很多开发商来说当然是非常具有吸引力，尤其是初创公司和新入行的开发者们。

(P 3-7-1) unity3D

优点：业内最具竞争力的授权条款；易于使用而且兼容所有游戏平台；开发者社区支持强大；学习门槛非常第；开发商使用率最高。

缺点：工具数量有限，所以开发商必须给自己创作工具；做复杂和多样化的效果比较耗时。

2. 虚幻引擎

数年以来，虚幻引擎一直是做高端 EA 游戏最受欢迎的引擎。《战争机器》、《蝙蝠侠：阿卡汉姆疯人院》(Batman: Arkham Asylum)、《质量效应》以及很多大作都是出自该引擎之手。

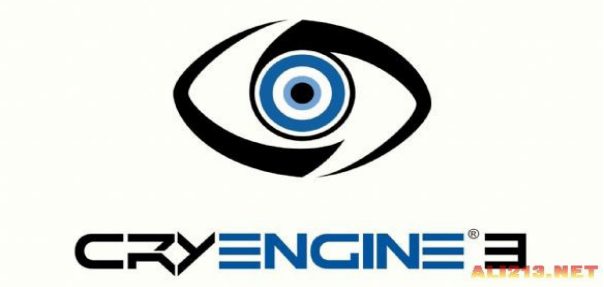
(P 3-7-2) 虚幻引擎

优点：开发商使用率较高，开发商社区支持支持强大，有视频教程和大量资源。最佳的引擎支持并且随时更新其他引擎平台的功能，每次更新都会增加新工具，而且管理相对容易，有些工具甚至小学生都会使用。兼容大多数平台，比如 iOS、Android、Linux、Mac、Windows 和大多数游戏主机。

缺点：授权条款只适合大作，商业授权价格为 99 美元，在游戏收入超过 5 万美元之后，必须支付 25% 的分成。也有一些开发者抱怨有些工具不好用，学习门槛较高。

3.CryEngine 3

该游戏引擎以优质的画面输出获得了大量开发者认可，如果你要做视觉出色的游戏，这款引擎绝对是最理想的选择。不过，该引擎也有自己的问题。



(P 3-7-3) CryEngine 3

4.HeroEngine

该引擎在 MMO 和在线游戏领域获得了非常高的人气，代表作《星球大战：旧共和国》。对于新入行的开发者以及初创公司来说，授权费用较高，不过，如果你有一个非常具备潜力的项目，该引擎还是非常值得考虑的。

(P 3-7-4) HeroEngine

优点：提供多个开放世界地图，而且可以实现无缝转换；提供相对完善的 AI；地图工具简单易用，并且集成了多个工具；脚本强大，足够帮助开发者研发复杂的项目、获得需要的资源；可以通过 HeroCloud 支持客户服务器。

缺点：脚本引擎强大但不够直观；HeroEngine 和 HeroCloud 对于初创公司来说成本较高；新开发者学习门槛较高。

5. Rage Engine

该引擎的用途非常多，比较知名的游戏包括 GTA Ⅲ、GTA: Vice City、GTA: San Andreas 和很多知名大作。

(P 3-7-5) Rage Engine

优点：兼容与处理较大世界观和天气特效方面非常出色；复杂的 AI 设计方面领先其他引擎；非常适合多种玩法的游戏；网络编程速度非常快；非常具有吸引力的画质水平。

缺点：和其他顶级引擎相比界面比较差；对于键盘和鼠标控制优化做的不足。

6. Project Anarchy

该引擎是一套完整的端到端游戏引擎和尖端移动设备工具组，受到很多游戏开发者赞誉，但同样具有一些缺点。

(P 3-7-6) Project Anarchy

优点：提供免费的手游研发工具（主要平台，比如 ios、android 和 Tizen）；拥有非常强悍的程序调试工具；非常活跃的开发者社区与论坛；编辑器非常强大；音频输出能力非常好；优秀的 Havok AI；vForge 为开发者提供大量的定制化选择。

缺点：不支持 Mac 和 linux 开发环境；没有新手教学；对于初创公司来说成本较高

7. GameSalad

这款引擎据说是不用写代码就能做游戏。这并不是个噱头，该引擎的确支持这样的功能。不过 GameSalad 一些功能也并不好用。

(P 3-7-7) GameSalad

优点：如果你想独立开发一款 iPhone 游戏，该引擎非常适合你；对于快速实现游戏想法来说，GameSalad 是个不错的引擎；兼容流行的手游研发平台，比如 Cocona 和 Moai。

缺点：研发工具限制性比较强；缺乏大量的 iOS 功能；不兼容所有的平台。

8. GameMaker: Studio

作为开发者，如果你想要一款简单而又快速直接的游戏引擎开始项目，那么 GameMaker: Studio 绝对是理想之选；尽管授权费有些贵，但能够获得的功能还是物有所值的。

(P 3-7-8) GameMaker: Studio

优点：对于所有开发者来说都非常简单和直接；加入了编程语言（GML）；不需要处理内存管理或者多线程等方面的任务；独立于任何平台。

缺点：在内存问题方面的程序调试比较麻烦；授权费相对昂贵。

9. App Game Kit

该引擎是真正的跨平台研发工具，非常易用、简单，而且比较灵活。