一、 课程计划

目录

二、 数据分析

1. 数据分析定义

数据分析离不开数据，计量和记录一起促成了数据的诞生。伴随着数据记录的发展（尤其是技术），人类受益也越来越多，计算机出现带来的数字测量，更加大大的提高了数据化的效率。人们的重点也逐渐移向了记录下来的庞大数据，对这些数据进行研究、分析，以期获取更大的利益。

数据分析是指用适当的统计分析方法对收集来的数据进行分析，将它们加以汇总和理解并消化，以求最大化地开发数据的功能，发挥数据的作用。数据分析的目的是把隐藏在一大批看似杂乱无章的数据背后的信息集中和提炼出来，总结出所研究对象的内在规律。

商业领域中，数据分析能够给帮助企业进行判断和决策，以便采取相应的策略与行动。例如，企业高层希望通过市场分析和研究，把握当前产品的市场动向，从而指定合理的产品研发和销售计划，这就必须依赖数据分析才能完成。生活中最著名的例子便是天气专家通过对气象数据进行分析，并且制作出天气预报，根据预报，我们会做出相应的策略，是带伞还是加件毛衣。

数据分析可划分为：**描述性数据分析、探索性数据分析、验证性数据分析**。描述性数据分析属于初级数据分析，另两个属于高级数据分析。其中探索性分析侧重于在数据之中发现新的特征，而验证性数据分析则侧重于验证已有假设的真伪证明。我们日常学习和工作中所涉及的数据分析主要是描述性数据分析。

2. 数据分析作用

在商业领域中，数据分析的目的是把隐藏在数据背后的信息集中和提炼出来，总结出所研究对象的内在规律，帮助管理者进行有效的判断和决策。数据分析在企业日常经营分析中主要有三大作用：

2.1. 现状分析

简单来说就是告诉你当前的状况。具体体现在:

第一，告诉你企业现阶段的整体运营情况，通过各个指标的完成情况来衡量企业的运营状态，以说明企业整天运营是好了还是坏了，好的程度如何，坏的程度又到哪里。

第二，告诉你企业各项业务的构成，让你了解企业各项业务的发展以及变动情况，对企业运营状况有更深入的了解。

2.2. 原因分析

简单来说就是告诉你某一现状为什么发生。

经过现状分析，我们对企业的运营情况有了基本了解，但不知道运营情况具体好在哪里，差在哪里，是什么原因引起的。这时就需要开展原因分析，以进一步确定业务变动的具体原因。例如 2016 年 2 月运营收入下降 5%，是什么原因导致的呢，是各项业务收入都出现下降，还是个别业务收入下降引起的，是各个地区业务收入都出现下降，还是个别地区业务收入下降引起的。这就需要我们开展原因分析，进一步确定收入下降的具体原因，对运营策略做出调整与优化。

2.3. 预测分析

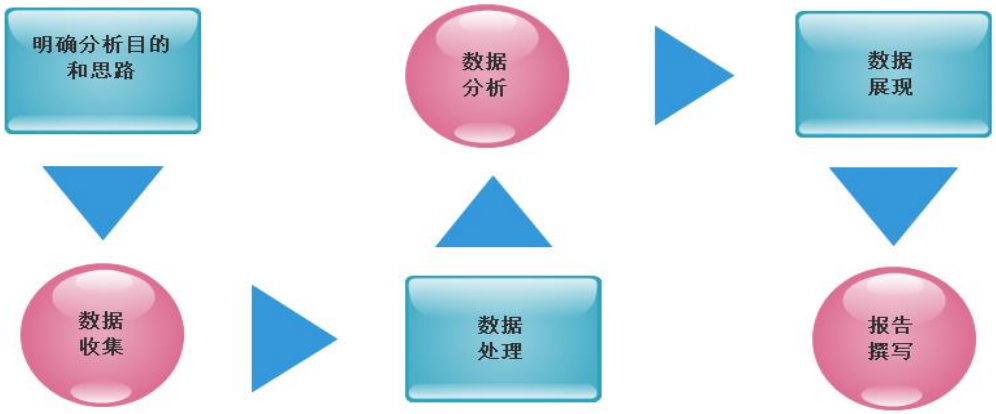
简单来说就是告诉你将来会发生什么。

在了解企业运营现状后，有时还需要对企业未来发展趋势做出预测，为制订企业运营目标及策略提供有效的参考与决策依据，以保证企业的可持续健康发展。

预测分析一般通过专题分析来完成，通常在制订企业季度、年度等计划时进行，其开展的频率没有现状分析及原因分析高。

3. 数据分析基本步骤

典型的数据分析包含以下几个步骤：



图：数据分析典型流程图

3.1. 明确分析目的和思路

明确数据分析目的以及确定分析思路，是确保数据分析过程有效进行的先决条件，它可以为数据的收集、处理及分析提供清晰的指引方向。

目的是整个分析流程的起点。目的不明确则会导致方向性的错误。即思考：为什么要开展数据分析，通过这次数据分析要解决什么问题？

当明确目的后，就要校理分析思路，并搭建分析框架，把分析目的分解成若干个不同的分析要点，即如何具体开展数据分析，需要从哪几个角度进行分析，采用哪些分析指标。只有明确了分析目的，分析框架才能跟着确定下来，最后还要确保分析框架的体系化，使分析更具有说服力。

体系化也就是逻辑化，简单来说就是先分析什么，后分析什么，使得各个分析点之间具有逻辑联系。避免不知从哪方面入手以及分析的内容和指标被质疑是否合理、完整。所以体系化就是为了让你的分析框架具有说服力。

要想使分析框架体系化，就需要一些营销、管理等理论为指导，结合着实际的业务情况进行构建，这样才能保证分析维度的完整性，分析结果的有效性以及正确性。比如以用户行为理论为指导，搭建的互联网网站分析指标框架如下：

把跟数据分析相关的营销、管理等理论统称为数据分析方法论。比如用户行为理论、PEST 分析法、5W2H 分析法等等，详细请查阅附件资料。

3.2. 数据收集

数据收集是按照确定的数据分析框架，收集相关数据的过程，它为数据分析提供了素材和依据。这里所说的数据包括第一手数据与第二手数据，第一手数据主要指可直接获取的数据，第二手数据主要指经过加工整理后得到的数据。一般数据来源主要有以下几种方式：

数据库：每个公司都有自己的业务数据库，存放从公司成立以来产生的相关业务数据。这个业务数据库就是一个庞大的数据资源，需要有效地利用起来。

公开出版物：可以用于收集数据的公开出版物包括《中国统计年鉴》《中国社会统计年鉴》《中国人口统计年鉴》《世界经济年鉴》《世界发展报告》等统计年鉴或报告。

互联网：随着互联网的发展，网络上发布的数据越来越多，特别是搜索引擎可以帮助我们快速找到所需要的数据，例如国家及地方统计局网站、行业组织网站、政府机构网站、传播媒体网站、大型综合门户网站等上面都可能有我们需要的数据。

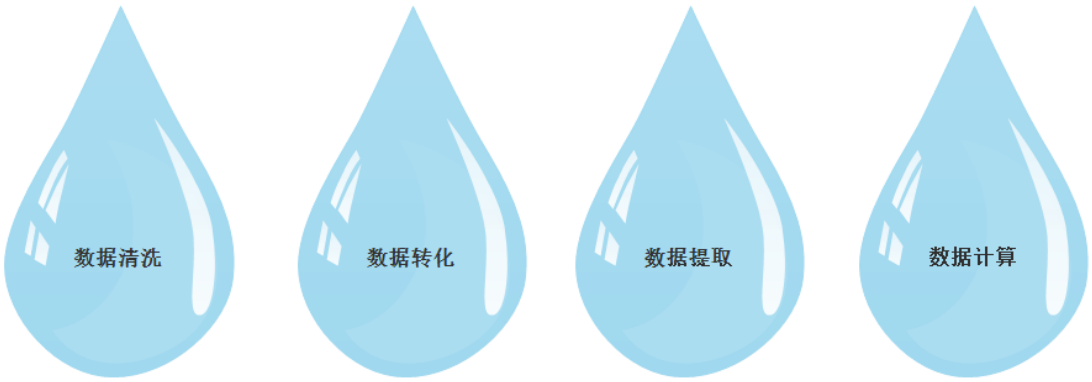
市场调查：进行数据分析时，需要了解用户的想法与需求，但是通过以上三种方式获得此类数据会比较困难，因此可以尝试使用市场调查的方法收集用户的想法和需求数据。市场调查就是指运用科学的方法，有目的、有系统地收集、记录、整理有关市场营销的信息和资料，分析市场情况，了解市场现状及其发展趋势，为市场预测和营销决策提供客观、正确的数据资料。市场调查可以弥补其他数据收集方式的不足，但进行市场调查所需的费用较高，而且会存在一定的误差，故仅作参考之用。

3.3. 数据处理

数据处理是指对收集到的数据进行加工整理，形成适合数据分析的样式，它是数据分析前必不可少的阶段。数据处理的基本目的是从大量的、杂乱无章、难以理解的数据中，抽取并推导出对解决问题有价值、有意义的数据。

数据处理主要包括**数据清洗、数据转化、数据提取、数据计算**等处理方法。一般拿到手的数据都需要进行一定的处理才能用于后续的数据分析工作，即使再“干净”’的原始数据也需要先进行一定的处理才能使用。

数据处理是数据分析的基础。通过数据处理，将收集到的原始数据转换为可以分析的形式，并且保证数据的一致性和有效性。



3.4. 数据分析

数据分析是指用适当的分析方法及工具，对处理过的数据进行分析，提取有价值的信息，形成有效结论的过程。由于数据分析多是通过软件来完成的，这就要求数据分析师不仅要掌握各种数据分析方法，还要熟悉数据分析软件的操作。

数据挖掘其实是一种高级的数据分析方法，就是从大量的数据中挖掘出有用的信息，它是根据用户的特定要求，从浩如烟海的数据中找出所需的信息，以满足用户的特定需求。数据挖掘技术是人们长期对数据库技术进行研究和开发的结果。一般来说，数据挖掘侧重解决四类数据分析问题:分类、聚类、关联和预测，重点在寻找模式和规律。数据分析与数据挖掘的本质是一样的，都是从数据里面发现关于业务的知识。

3.5. 数据展现

一般情况下，数据是通过表格和图形的方式来呈现的，我们常说用图表说话就是这个意思。常用的数据图表包括饼图、柱形图、条形图、折线图、散点图、雷达图等，当然可以对这些图表进一步整理加工，使之变为我们所需要的图形，例如金字塔图、矩阵图、漏斗图等。

大多数情况下，人们更愿意接受图形这种数据展现方式，因为它能更加有效、直观地传递出分析所要表达的观点。记位，一般情况不，能用图说明问题的就不用表格，能用表格说明问题的就不要用文字。

3.6. 报告撰写

数据分析报告其实是对整个数据分析过程的一个总结与呈现。通过报告，把数据分析的起因、过程、结果及建议完整地呈现出来，供决策者参考。

一份好的数据分析报告，首先需要有一个好的分析框架，并且图文并茂，层次明晰，能够让阅读者一目了然。结构清晰、主次分明可以使阅读者正确理解报告内容;图文并茂，可以令数据更加生动活泼 ,提供视觉冲击力，有助于阅读者更形象、直观地看清楚问题和结论，从而产生思考。

另外，数据分析报告需要有明确的结论，没有明确结论的分析称不上分析，同时也失去了报告的意义，因为我们最初就是为寻找或者求证一个结论才进行分析的，所以千万不要舍本求末。

最后，好的分析报告一定要有建议或解决方案。作为决策者，需要的不仅仅是找出问题，更重要的是建议或解决方案，以便他们做决策时作参考。所以，数据分析师不仅需要掌握数据分析方法，而且还要了解和熟悉业务，这样才能根据发现的业务问题，提出具有可行性的建议或解决方案。

4. 数据分析行业前景

4.1. 蓬勃发展的趋势

从 20 世纪 90 年代起，欧美国家开始大量培养数据分析师，直到现在，对数据分析师的需求仍然长盛不衰，而且还有扩展之势。

对于**中国数据分析行业前景和特点**，一面网络创始人何明科指出：

一是：市场巨大，许多企业（无论是互联网的新锐还是传统的企业）都在讨论这个，也有实际的需求并愿意为此付钱，但是比较零碎尚不系统化。目前对数据需求最强烈的行业依次是：金融机构（从基金到银行到保险公司到 P2P 公司），以广告投放及电商为代表的互联网企业等；

二是：尚没出现平台级公司的模式（这或许往往是大市场或者大机会出现之前的混沌期）；

三是：企业技术外包的氛围在国内尚没完全形成，对于一些有能力的技术公司，如果数据需求强烈的话，考虑到自身能力的健全以及数据安全性，往往不会外包或者采用外部模块，而倾向于自建这块业务；

四是：未来 BAT 及京东、58 和滴滴打车等企业，凭借其自身产生的海量数据，必然是数据领域的大玩家。但是整个行业很大而且需求旺盛，即使没有留给创业公司出现平台级巨型企业的机会，也将留出各种各样的细分市场机会让大家可以获得自己的领地。

4.2. 数据分析师的职业要求

懂业务：从事数据分析工作的前提就是需要懂业务，即熟悉行业知识、公司业务及流程，最好有自己独特见解，若脱离行业认知和公司业务背景，分析的结果只会是脱了线的风筝，没有太大的实用价值。

从另外一个角度来说，懂业务也是数据敏感的体现。不懂业务的数据分析师，看到的只是一个个数字;懂业务的数据分析师，则看到的不仅仅是数字，他明白数字代表什么意义，知道数字是大了还是小了，心中有数，这才是真正意义的数据敏感性。

懂管理：一方面是搭建数据分析框架的要求，比如数据分析第一步确定分析思路就需要用到营销、管理等理论知识来指导，如果不熟悉管理理论，那你如何指导数据分析框架的搭建，以及开展后续的数据分析呢?

懂管理另一方面的作用是针对数据分析结论提出有指导意义的分析建议，如果没有管理理论的支撑，就难以确保分析建议的有效性。

懂分析：是指掌握数据分析的基本原理与一些有效的数据分析方法，并能灵活运用到实践工作中，以便有效地开展数据分析。

懂工具：是指掌握数据分析相关的常用工具。数据分析工具就是实现数据分析方法理论的工具，面对越来越庞大的数据，依靠计算器进行分析是不现实的，必须利用强大的数据分析工具完成数据分析工作。

同样，应该根据研究的问题选择合适的工具，只要能解决问题的工具就是好工具。

懂设计：是指运用图表有效表达数据分析师的分析观点，使分析结果一目了然。图表的设计是门大学问，如图形的选择、版式的设计、颜色的搭配等，都需要掌握一定的设计原则。

三、 科技发展带来的挑战

在科技的快速发展推动下，在 IT 领域，企业会面临两个方面的问题。

一是如何实现网站的高可用、易伸缩、可扩展、高安全等目标。为了解决这样一系列问题，迫使网站的架构在不断发展。从单一架构迈向高可用架构，这过程中不得不提的就是分布式。

二是用户规模越来越大，由此产生的数据也在以指数倍增长，俗称数据大爆炸。海量数据处理的场景也越来越多。技术上该如何面对？

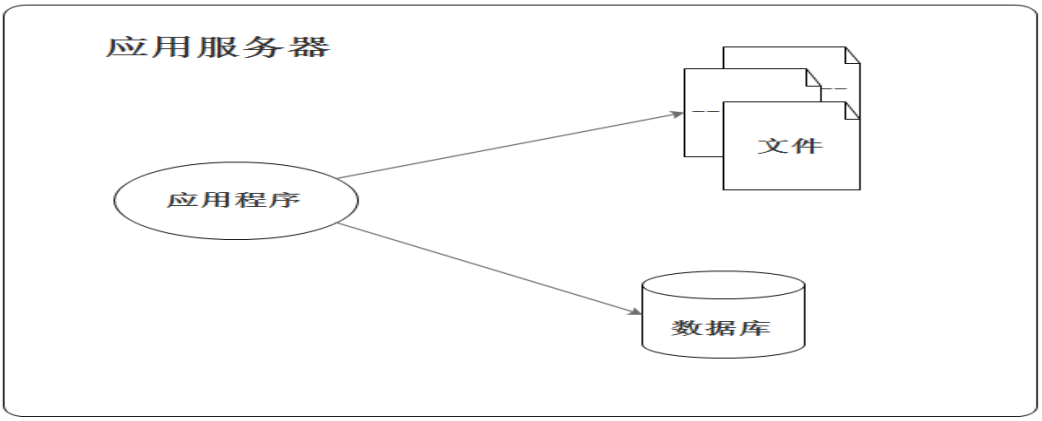
1. 分布式系统

1.1. 概述

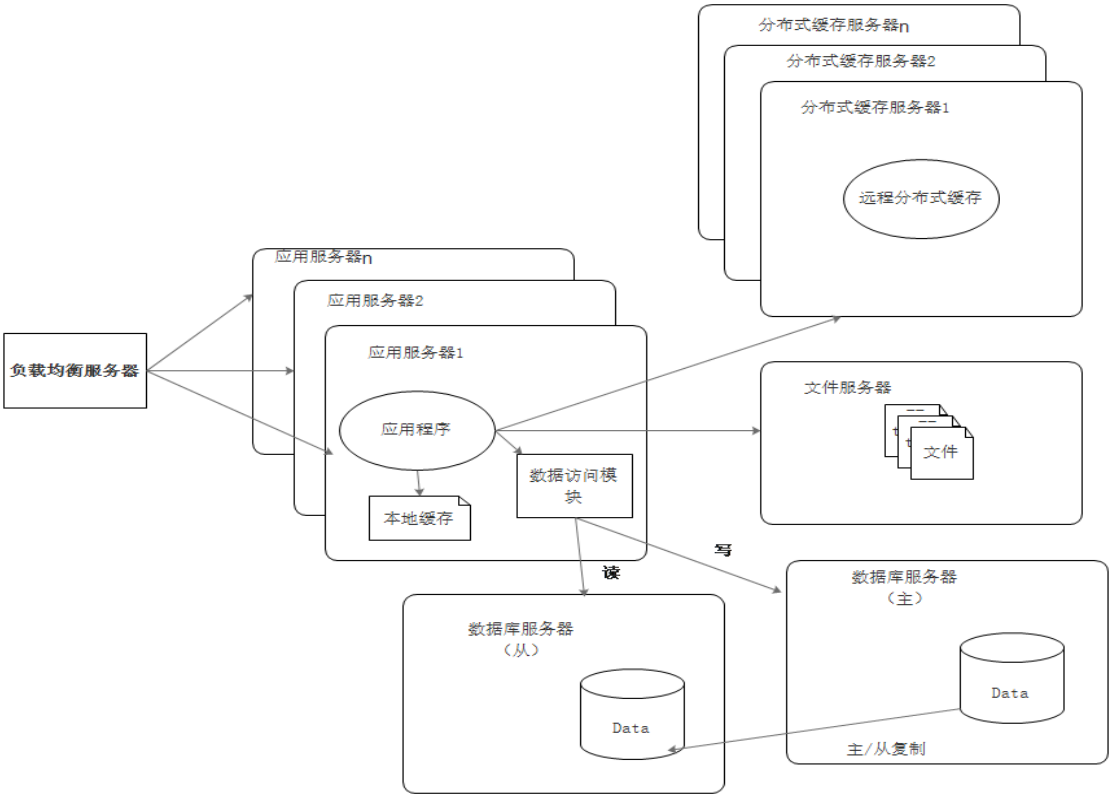
分布式系统是一个硬件或软件组件分布在不同的网络计算机上，彼此之间仅仅通过消息传递进行通信和协调的系统。简单来说就是一群独立计算机集合共同对外提供服务，但是对于系统的用户来说，就像是一台计算机在提供服务一样。

分布式意味着可以采用更多的普通计算机（相对于昂贵的大型机）组成分布式集群对外提供服务。计算机越多，CPU、内存、存储资源等也就越多，能够处理的并发访问量也就越大。

初代的 web 服务网站架构往往比较简单，应用程序、数据库、文件等所有的资源都在一台服务器上。



图：互联网初始阶段的网站架构



图：现在互联网网站常用的架构

从分布式系统的概念中我们知道，各个主机之间通信和协调主要通过网络进行，所以，分布式系统中的计算机在空间上几乎没有任何限制，这些计算机可能被放在不同的机柜上，也可能被部署在不同的机房中，还可能在不同的城市中，对于大型的网站甚至可能分布在不同的国家和地区。

1.2. 特征

分布性：分布式系统中的多台计算机之间在空间位置上可以随意分布，系统中的多台计算机之间没有主、从之分，即没有控制整个系统的主机，也没有受控的从机。

透明性：系统资源被所有计算机共享。每台计算机不仅可以使用本机的资源，还可以使用分布式系统中其他计算机的资源(包括 CPU、文件、打印机等)。

同一性：系统中的若干台计算机可以互相协作来完成一个共同的任务，或者说一个程序可以分布在几台计算机上并行地运行。

通信性：系统中任意两台计算机都可以通过通信来交换信息。

1.3. 常用分布式方案

分布式应用和服务

将应用和服务进行分层和分割，然后将应用和服务模块进行分布式部署。这样做不仅可以提高并发访问能力、减少数据库连接和资源消耗，还能使不同应用复用共同的服务，使业务易于扩展。比如：分布式服务框架 Dubbo。

分布式静态资源

对网站的静态资源如 JS、CSS、图片等资源进行分布式部署可以减轻应用服务器的负载压力，提高访问速度。比如：CDN。

分布式数据和存储

大型网站常常需要处理海量数据，单台计算机往往无法提供足够的内存空间，可以对这些数据进行分布式存储。比如 Apache Hadoop HDFS。

分布式计算

随着计算技术的发展，有些应用需要非常巨大的计算能力才能完成，如果采用集中式计算，需要耗费相当长的时间来完成。分布式计算将该应用分解成许多小的部分，分配给多台计算机进行处理。这样可以节约整体计算时间，大大提高计算效率。比如 Apache Hadoop MapReduce。

1.4. 分布式、集群

分布式（distributed）是指在多台不同的服务器中部署不同的服务模块，通过远程调用协同工作，对外提供服务。

集群（cluster）是指在多台不同的服务器中部署相同应用或服务模块，构成一个集群，通过负载均衡设备对外提供服务。

2. 海量数据处理

公开数据显示，互联网搜索巨头百度 2013 年拥有数据量接近 EB 级别。阿里、腾讯都声明自己存储的数据总量都达到了百 PB 以上。此外，电信、医疗、金融、公共安全、交通、气象等各个方面保存的数据量也都达到数十或者上百 PB 级别。全球数据量以每两年翻倍的速度增长,在 2010 年已经正式进入 ZB 时代，到 2020 年全球数据总量将达到 44ZB。

```
1KB (Kilobyte 千)=1024B,
1MB (Megabyte 兆)=1024KB,
1GB (Gigabyte 吉)=1024MB,
1TB (Trillionbyte 太)=1024GB,
1PB（Petabyte 拍）=1024TB,
1EB（Exabyte 艾）=1024PB,
1ZB (Zettabyte 泽)= 1024 EB,
1YB (Yottabyte 尧)= 1024 ZB,
1BB (Brontobyte 布)= 1024 YB.
```

数据分析的前提是有数据，数据存储的目的是支撑数据分析。究竟怎么去存储庞大的数据量，是开展数据分析的企业在当下面临的一个问题。传统的数据存储模式存储容量是有大小限制或者空间局限限制的，怎么去设计出一个可以支撑大量数据的存储方案是开展数据分析的首要前提。

当解决了海量数据的存储问题，接下来面临的海量数据的计算问题也是比较让人头疼，因为企业不仅追求可以计算，还会追求计算的速度、效率。

以目前互联网行业产生的数据量级别，要处理这些数据，就需要一个更好、更便捷的分析计算方式了。传统的显然力不从心了，而且效率也会非常低下。这正是传统数据分析领域面临的另一个挑战，如何让去分析、计算。

四、 大数据时代

1.1. 概述

最早提出“大数据”时代到来的是全球知名咨询公司麦肯锡，麦肯锡称：“数据，已经渗透到当今每一个行业和业务职能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率增长和消费者盈余浪潮的到来。”

随着互联网快速发展、智能手机以及“可佩带”计算设备的出现，我们的行为、位置，甚至身体生理数据等每一点变化都成为了**可被记录**和分析的数据。这些新技术推动着大数据时代的来临，各行各业每天都在产生数量巨大的数据碎片，数据计量单位已从 Byte、KB、MB、GB、TB 发展到 PB、EB、ZB、YB 甚至 BB 来衡量。

大数据到底是什么，如果简单来理解大数据就是 4V 的特征：

Volume（大量）、Velocity（高速）、Variety（多样）、Value（价值），即数据体量巨大、数据类型繁多、价值密度低、处理速度快。

但是这样理解会显得太浅显，要想更加全面了解大数据概念可以查看附件资料《大数据时代》。

1.2. 大数据分析

当数据分析遇到大数据时代，于是就产生了完美的契合：**大数据分析**。你可以理解大数据分析是指对规模巨大的数据进行分析。大数据被称为当今最有潜质的 IT 词汇，接踵而来的数据挖掘、数据安全、数据分析、数据存储等等围绕大数据的商业价值的利用逐渐成为行业人士争相追捧的利润焦点。随着大数据时代的来临，大数据分析也应运而生。

大数据分析具体含义可以分为以下几个方面：

一是大数据分析可以让人们对数据产生更加优质的诠释，而具有预知意义的分析可以让分析员根据可视化分析和数据分析后的结果做出一些预测性的推断。

二是大数据的分析与存储和数据的管理是一些数据分析层面的最佳实践。通过按部就班的流程和工具对数据进行分析可以保证一个预先定义好的高质量的分析结果。

此外需要注意的是：

传统的数据分析就是在数据中寻找有价值的规律，这和现在的大数据在方向上是一致的。

五、 大数据分析系统

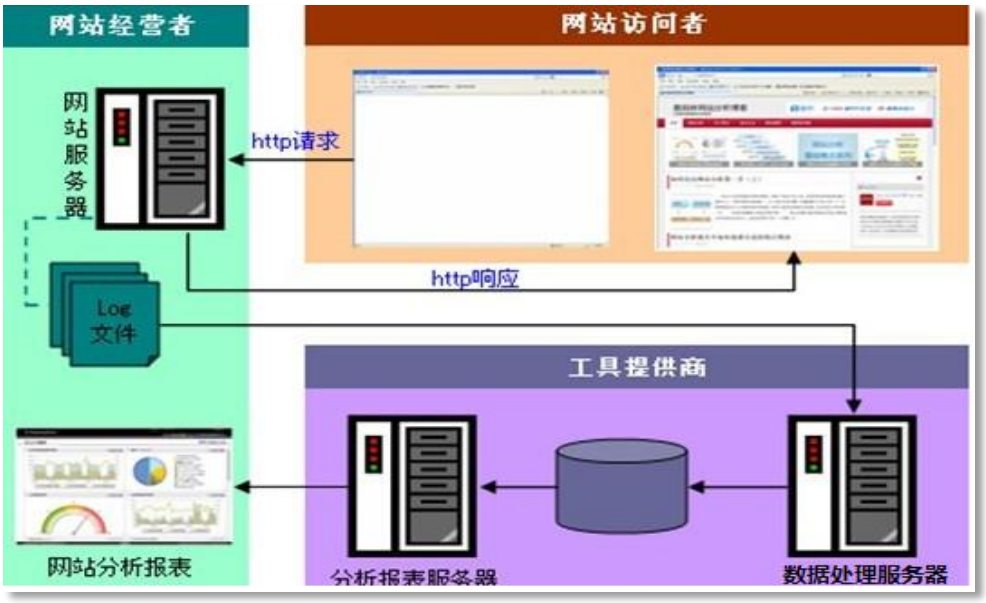
1. 概念、分类

数据分析系统的主要功能是从众多外部系统中，采集相关的业务数据，集中存储到系统的数据库中。系统内部对所有的原始数据通过一系列处理转换之后，存储到数据仓库的基础库中；然后，通过业务需要进行一系列的数据转换到相应的数据集市，供其他上层数据应用组件进行专题分析或者展示。

根据数据的流转流程，一般会有以下几个模块：**数据收集（采集）、数据存储、数据计算、数据分析、数据展示**等等。当然也会有在这基础上进行相应变化的系统模型。

按照数据分析的时效性，我们一般会把大数据分析系统分为实时、离线两种类型。实时数据分析系统在时效上有强烈的保证，数据是实时流动的，相应的一些分析情况也是实时的。而离线数据分析系统更多的是对已有的数据进行分析，时效性上的要求会相对低一点。时效性的标准都是以人可以接受来划分的。

2. 网站流量日志数据分析系统



2.1. 系统的意义

网站流量数据统计分析，可以帮助网站管理员、运营人员、推广人员等实时获取网站流量信息，并从流量来源、网站内容、网站访客特性等多方面提供网站分析的数据依据。从而帮助提高网站流量，提升网站用户体验，让访客更多的沉淀下来变成会员或客户，通过更少的投入获取最大化的收入。

- 技术上

可以合理修改网站结构及适度分配资源，构建后台服务器群组，比如

1、辅助改进网络的拓扑设计，提高性能

2、在有高度相关性的节点之间安排快速有效的访问路径

3、帮助企业更好地设计网站主页和安排网页内容

- 业务上

1、帮助企业改善市场营销决策，如把广告放在适当的 Web 页面上。

2、优化页面及业务流程设计，提高流量转化率。

3、帮助企业更好地根据客户的兴趣来安排内容。

4、帮助企业对客户群进行细分，针对不同客户制定个性化的促销策略等。

终极目标是：

改善网站的运营，获取更高投资回报率（ROI）。也就是赚更多的钱。

2.2. 背景知识—Web 访问日志

访问日志指用户访问网站时的所有访问、浏览、点击行为数据。比如点击了哪一个链接，打开了哪一个页面，采用了哪个搜索项、总体会话时间等。而所有这些信息都可通过网站日志保存下来。通过分析这些数据，可以获知许多对网站运营至关重要的信息。采集的数据越全面，分析就能越精准。

日志的生成渠道分为以下两种：

一是：web 服务器软件（httpd、nginx、tomcat）自带的日志记录功能，如 Nginx 的 access.log 日志；

二是：自定义采集用户行为数据，通过在页面嵌入自定义的 javascript 代码来获取用户的访问行为（比如鼠标悬停的位置，点击的页面组件等），然后通过 ajax 请求到后台记录日志，这种方式所能采集的信息会更加全面。

在实际操作中，有以下几个方面的数据可以自定义的采集：

系统特征：比如所采用的操作系统、浏览器、域名和访问速度等。

访问特征：包括停留时间、点击的 URL、所点击的“页面标签<a>”及标签的属性等。

来源特征：包括来访 URL，来访 IP 等。

产品特征：包括所访问的产品编号、产品类别、产品颜色、产品价格、产品利润、产品数量和特价等级等。

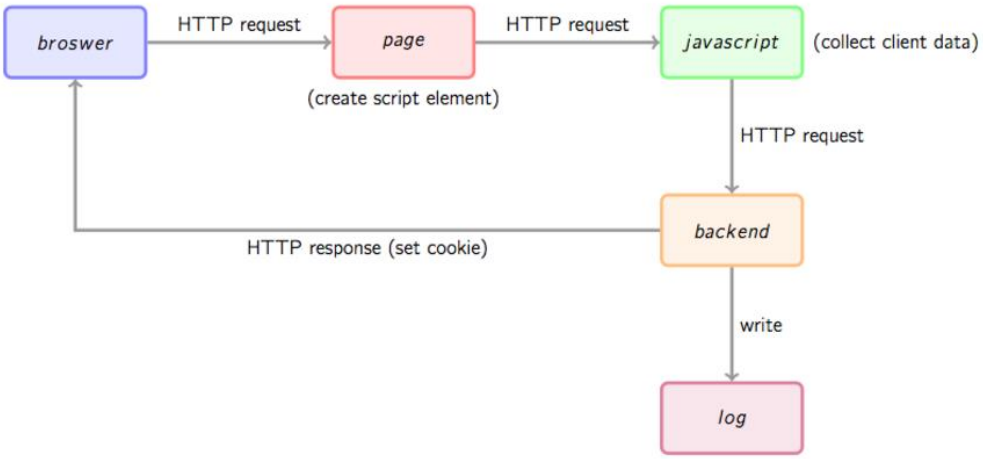
以电商某东为例，其自定义采集的数据日志格式如下：

```
GET /log.gif?t=item.010001&m=UA-J2011-1&pin=-&uid=1679790178&sid=1679790178|12&v=je=1$sc=24-bit$sr=1600x900$ul=zh-cn$cs=GBK$dt=【云南白药套装】云南白药 牙膏 180g×3 （留兰香型）【行情 报价 价格 评测】-京东$hn=item.jd.com$fl=16.0r0$os=win$br=chrome$bv=39.0.2171.95$wb=1437269412$xb=1449548587$yb=1456186252$zb=12$cb=4$usc=direct$ucp=-$umd=none$uct=-$ct=1456186505411$lt=0$tad=-$sku=1326523$cid1=1316$cid2=1384$cid3=1405$brand=20583$pinid=-&ref=&rm=1456186505411 HTTP/1.1
```

六、 网站流量日志数据自定义采集

1. 原理分析

首先，用户的行为会触发浏览器对被统计页面的一个 http 请求，比如打开某网页。当网页被打开，页面中的埋点 javascript 代码会被执行。

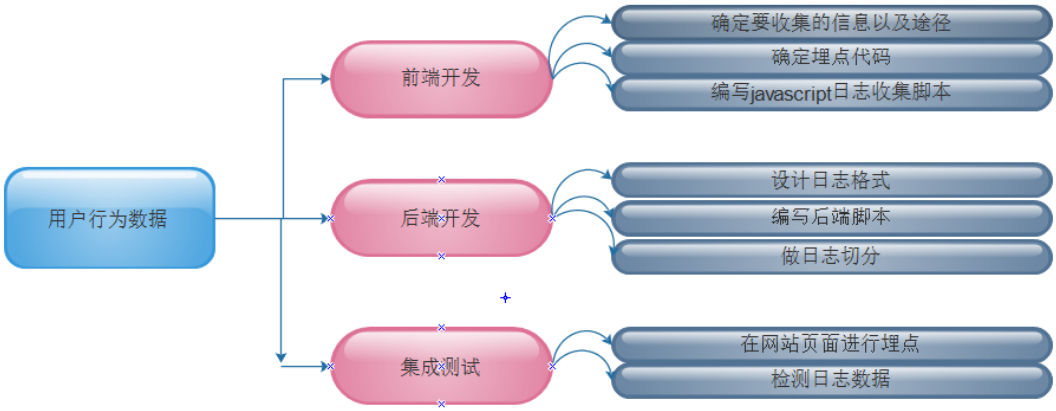


埋点是指：在网页中预先加入小段 javascript 代码，这个代码片段一般会动态创建一个 script 标签，并将 src 属性指向一个单独的 js 文件，此时这个单独的 js 文件（图中绿色节点）会被浏览器请求到并执行，这个 js 往往就是真正的数据收集脚本。

数据收集完成后，js 会请求一个后端的数据收集脚本（图中的 backend），这个脚本一般是一个伪装成图片的动态脚本程序，js 会将收集到的数据通过 http 参数的方式传递给后端脚本，后端脚本解析参数并按固定格式记录到访问日志，同时可能会在 http 响应中给客户端种植一些用于追踪的 cookie。

2. 设计实现

根据原理分析并结合 Google Analytics，想搭建一个自定义日志数据采集系统，要做以下几件事：



2.1. 确定收集信息

名称	途径	备注
访问时间	web server	Nginx $msec
IP	web server	Nginx $remote_addr
域名	javascript	document.domain
URL	javascript	document.URL
页面标题	javascript	document.title
分辨率	javascript	window.screen.height & width
颜色深度	javascript	window.screen.colorDepth
Referrer	javascript	document.referrer
浏览客户端	web server	Nginx $http_user_agent
客户端语言	javascript	navigator.language
访客标识	cookie	Nginx $http_cookie
网站标识	javascript	自定义对象
状态码	web server	Nginx $status
发送内容量	web server	Nginx $body_bytes_sent

2.2. 确定埋点代码

埋点，是网站分析的一种常用的数据采集方法。核心就是在需要进行数据采集的关键点植入统计代码，进行数据的采集。比如以谷歌分析原型来说，需要在页面中插入一段它提供的 javascript 片段，这个片段往往被称为埋点代码。

```
<script type="text/javascript">
  var _maq = _maq || [];
  _maq.push(['_setAccount', 'UA-XXXXX-X']);
  (function() {
    var ma = document.createElement('script'); ma.type =
'text/javascript'; ma.async = true;
    ma.src = ('https:' == document.location.protocol ?
'https://ssl' : 'http://www') + '.google-analytics.com/ma.js';
    var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ma, s);
  })();
</script>
```

其中_maq 是全局数组，用于放置各种配置，其中每一条配置的格式为：

_maq.push(['Action', 'param1', 'param2', ...]);

_maq 的机制不是重点，重点是后面匿名函数的代码，这段代码的主要目的就是引入一个外部的 js 文件（ma.js），方式是通过 document.createElement 方法创建一个 script 并根据协议（http 或 https）将 src 指向对应的 ma.js，最后将这个元素插入页面的 dom 树上。

注意 ma.async = true 的意思是异步调用外部 js 文件，即不阻塞浏览器的解析，待外部 js 下载完成后异步执行。这个属性是 HTML5 新引入的。

扩展知识：js 自调用匿名函数

格式： (function(){})();

第一对括号向脚本返回未命名的函数；后一对空括号立即执行返回的未命名函数，括号内为匿名函数的参数。

自调用匿名函数的好处是，避免重名，自调用匿名函数只会在运行时执行一次，一般用于初始化。

2.3. 前端数据收集脚本

数据收集脚本（ma.js）被请求后会被执行，一般要做如下几件事：

1、通过浏览器内置 javascript 对象收集信息，如页面 title（通过 document.title）、referrer（上一跳 url，通过 document.referrer）、用户显示器分辨率（通过 windows.screen）、cookie 信息（通过 document.cookie）等等一些信息。

2、解析_maq 数组，收集配置信息。这里面可能会包括用户自定义的事件跟踪、业务数据（如电子商务网站的商品编号等）等。

3、将上面两步收集的数据按预定义格式解析并拼接（get 请求参数）。

4、请求一个后端脚本，将信息放在 http request 参数中携带给后端脚本。

这里唯一的问题是步骤 4，javascript 请求后端脚本常用的方法是 ajax，但是 ajax 是不能跨域请求的。一种通用的方法是 js 脚本创建一个 Image 对象，将 Image 对象的 src 属性指向后端脚本并携带参数，此时即实现了跨域请求后端。这也是后端脚本为什么通常伪装成 gif 文件的原因。

示例代码：

```
(function () {
        var params = {};
        //Document 对象数据
        if(document) {
                params.domain = document.domain || '';
                params.url = document.URL || '';
                params.title = document.title || '';
                params.referrer = document.referrer || '';
        }
        //Window 对象数据
        if(window && window.screen) {
                params.sh = window.screen.height || 0;
                params.sw = window.screen.width || 0;
                params.cd = window.screen.colorDepth || 0;
        }
        //navigator 对象数据
        if(navigator) {
                params.lang = navigator.language || '';
        }
        //解析_maq 配置
        if(_maq) {
                for(var i in _maq) {
```

```
                        switch(_maq[i][0]) {
                                case '_setAccount':
                                    params.account = _maq[i][1];
                                    break;
                                default:
                                    break;
                        }
                }
        }
        //拼接参数串
        var args = '';
        for(var i in params) {
                if(args != '') {
                        args += '&';
                }
                args += i + '=' + encodeURIComponent(params[i]);
        }

        //通过 Image 对象请求后端脚本
        var img = new Image(1, 1);
        img.src = 'http://xxx.xxxxx.xxxxx/log.gif?' + args;
})();
```

整个脚本放在匿名函数里，确保不会污染全局环境。其中 log.gif 是后端脚本。

2.4. 后端脚本

log.gif 是后端脚本，是一个伪装成 gif 图片的脚本。后端脚本一般需要完成以下几件事情：

1、解析 http 请求参数得到信息。

2、从 Web 服务器中获取一些客户端无法获取的信息，如访客 ip 等。

3、将信息按格式写入 log。

4、生成一副 1×1 的空 gif 图片作为响应内容并将响应头的 Content-type 设为 image/gif。

5、在响应头中通过 Set-cookie 设置一些需要的 cookie 信息。

之所以要设置 cookie 是因为如果要跟踪唯一访客，通常做法是如果在请求时发现客户端没有指定的跟踪 cookie，则根据规则生成一个全局唯一的 cookie 并种植给用户，否则 Set-cookie 中放置获取到的跟踪 cookie 以保持同一用户 cookie 不变。这种做法虽然不是完美的（例如用户清掉 cookie 或更换浏览器会被认为是

两个用户），但是目前被广泛使用的手段。

我们使用 **nginx 的 access_log 做日志收集**，不过有个问题就是 nginx 配置本身的逻辑表达能力有限，所以选用 OpenResty 做这个事情。

OpenResty 是一个基于 Nginx 扩展出的高性能应用开发平台，内部集成了诸多有用的模块，其中的核心是通过 ngx_lua 模块集成了 Lua，从而在 nginx 配置文件中可以通过 Lua 来表述业务。

Lua 是一种轻量小巧的脚本语言，用标准 C 语言编写并以源代码形式开放，其设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能。

首先，需要在 nginx 的配置文件中定义日志格式：

```
log_format tick
"$msec||$remote_addr||$status||$body_bytes_sent||$u_domain||$u_url||
||$u_title||$u_referrer||$u_sh||$u_sw||$u_cd||$u_lang||$http_user_ag
ent||$u_account";
```

注意这里以 u_开头的是我们待会会自己定义的变量，其它的是 nginx 内置变量。然后是核心的两个 location：

```
location /log.gif {
        #伪装成 gif 文件
        default_type image/gif;
        #本身关闭 access_log，通过 subrequest 记录 log
        access_log off;

        access_by_lua "
                -- 用户跟踪 cookie 名为__utrace
                local uid = ngx.var.cookie___utrace
                if not uid then
                        -- 如果没有则生成一个跟踪 cookie，算法为
md5(时间戳+IP+客户端信息)
                        uid = ngx.md5(ngx.now() ..
ngx.var.remote_addr .. ngx.var.http_user_agent)
                end
                ngx.header['Set-Cookie'] = {'__utrace=' .. uid ..
'; path=/'}
                if ngx.var.arg_domain then
                        -- 通过 subrequest 子请求到/i-log 记录日志，
将参数和用户跟踪 cookie 带过去
                        ngx.location.capture('/i-log?' ..
ngx.var.args .. '&utrace=' .. uid)
```

```
                end
        ";

        #此请求资源本地不缓存
        add_header Expires "Fri, 01 Jan 1980 00:00:00 GMT";
        add_header Pragma "no-cache";
        add_header Cache-Control "no-cache, max-age=0, must-
revalidate";
        #返回一个 1×1 的空 gif 图片
        empty_gif;
}

location /i-log {
        #内部 location，不允许外部直接访问
        internal;

        #设置变量，注意需要 unescape，来自 ngx_set_misc 模块
        set_unescape_uri $u_domain $arg_domain;
        set_unescape_uri $u_url $arg_url;
        set_unescape_uri $u_title $arg_title;
        set_unescape_uri $u_referrer $arg_referrer;
        set_unescape_uri $u_sh $arg_sh;
        set_unescape_uri $u_sw $arg_sw;
        set_unescape_uri $u_cd $arg_cd;
        set_unescape_uri $u_lang $arg_lang;
        set_unescape_uri $u_account $arg_account;
        #打开日志
        log_subrequest on;
        #记录日志到 ma.log 格式为 tick
        access_log /path/to/logs/directory/ma.log tick;

        #输出空字符串
        echo '';
}
```

要完全掌握这段脚本的每一个细节还是比较吃力的，用到了诸多第三方 ngxin 模块（全都包含在 OpenResty 中了），重点都用注释标出来，可以不用完全理解每一行的意义，只要大约知道这个配置完成了我们提到的后端逻辑就可以了。

2.5. 日志格式

日志格式主要考虑日志分隔符，一般会有以下几种选择：

固定数量的字符、制表符分隔符、空格分隔符、其他一个或多个字符、特定的开始和结束文本。

2.6. 日志切分

日志收集系统访问日志时间一长文件变得很大，而且日志放在一个文件不便于管理。通常要按时间段将日志切分，例如每天或每小时切分一个日志。通过 crontab 定时调用一个 shell 脚本实现，如下：

```
_prefix="/path/to/nginx"
time=`date +%Y%m%d%H`
mv ${_prefix}/logs/ma.log ${_prefix}/logs/ma/ma-${time}.log
kill -USR1 `cat ${_prefix}/logs/nginx.pid `
```

这个脚本将 ma.log 移动到指定文件夹并重命名为 ma-{yyyymmddhh}.log，然后向 nginx 发送 USR1 信号令其重新打开日志文件。

USR1 通常被用来告知应用程序重载配置文件，向服务器发送一个 USR1 信号将导致以下步骤的发生：停止接受新的连接，等待当前连接停止，重新载入配置文件，重新打开日志文件，重启服务器，从而实现相对平滑的不关机的更改。

cat ${_prefix}/logs/nginx.pid 取 nginx 的进程号

然后再/etc/crontab 里加入一行：

```
59 * * * * root /path/to/directory/rotatelog.sh
```

在每个小时的 59 分启动这个脚本进行日志轮转操作。

3. 系统环境部署

服务器中安装依赖

```
yum -y install gcc perl pcre-devel openssl openssl-devel
```

上传 LuaJIT-2.0.4.tar.gz 并安装 LuaJIT

```
tar -zxvf LuaJIT-2.0.4.tar.gz -C /usr/local/src/
cd /usr/local/src/LuaJIT-2.0.4/
make && make install PREFIX=/usr/local/luajit
```

设置 LuaJIT 环境变量

```
vi /etc/profile
export LUAJIT_LIB=/usr/local/luajit/lib
export LUAJIT_INC=/usr/local/luajit/include/luajit-2.0
source      /etc/profile
```

创建 modules 文件夹，保存 nginx 依赖的模块

```
mkdir -p /usr/local/nginx/modules
```

上传 nginx 依赖的模块

```
set-misc-nginx-module-0.29.tar.gz
lua-nginx-module-0.10.0.tar.gz
ngx_devel_kit-0.2.19.tar.gz
echo-nginx-module-0.58.tar.gz
```

将依赖的模块直接解压到 modules 目录

```
tar -zxvf lua-nginx-module-0.10.0.tar.gz -C /usr/local/nginx/modules/
tar -zxvf set-misc-nginx-module-0.29.tar.gz -C /usr/local/nginx/modules/
tar -zxvf ngx_devel_kit-0.2.19.tar.gz -C /usr/local/nginx/modules/
tar -zxvf echo-nginx-module-0.58.tar.gz -C /usr/local/nginx/modules/
```

安装 openresty

```
tar -zxvf openresty-1.9.7.3.tar.gz -C /usr/local/src/
cd /usr/local/src/openresty-1.9.7.3/
./configure --prefix=/usr/local/openresty --with-luajit && make && make install
```

安装 nginx

```
tar -zxvf nginx-1.8.1.tar.gz -C /usr/local/src/
```

编译 nginx 并支持其他模块

```
cd /usr/local/src/nginx-1.8.1/
./configure --prefix=/usr/local/nginx \
    --with-ld-opt="-Wl,-rpath,/usr/local/luajit/lib" \
    --add-module=/usr/local/nginx/modules/ngx_devel_kit-0.2.19 \
    --add-module=/usr/local/nginx/modules/lua-nginx-module-0.10.0 \
    --add-module=/usr/local/nginx/modules/set-misc-nginx-module-0.29 \
    --add-module=/usr/local/nginx/modules/echo-nginx-module-0.58
make -j2
make install
```

备注：如果对 linux 相关操作不熟，请严格按照上述步骤搭建环境，切记心细，心细，再心细。

4. 自定义采集数据实现

4.1. 方案一：基本功能实现

a) 创建页面 index.html，添加埋点代码，放入 nginx 默认目录 nginx/html 下。

b) 在默认目录 nginx/html 下添加一个数据采集脚本 ma.js。

c) 修改 nginx 的配置文件，添加自定义相关业务逻辑。

d) 启动 nginx

```
sbin/nginx -c conf/nginx.conf
```

e) 通过游览器访问 nginx

f) 观察自定义日志采集文件是否有对应的内容输出

```
tail  -f  logs/user_defined.log
```

此时还可以观察 nginx 默认的输出日志文件

```
tail  -f  logs/access.log
```

停止 nginx:

```
sbin/nginx  -s stop
```

4.2. 方案二：页面点击事件

详细步骤请参考附件资料。