

Big Data e Machine Learning com Hadoop e Spark



Sobre a Konfidentia

Sobre nós

A Konfidentia IT Solutions é uma empresa com 10 anos de mercado, atuante em serviços de Consultoria e Treinamento em Governança e Gestão de TI, BigData e Machine Learning, DevOps e Desenvolvimento para os seguintes temas:

- E-Commerce
- Inteligência de Negócios com BigData
- Modelos de Aprendizagem de Máquina
- Times de TI de Alta Performance
- Apoio a Pesquisa e Desenvolvimento
- Gestão da Inovação
- Gestão de Continuidade de Negócios / BCP
- Gestão de Riscos
- Gestão de Segurança da Informação
- Auditoria de TI
- Implantação de Escritórios de Projetos e Escritórios de Processos
- Implantação de Modelos de Gestão e Governança



Contato:

Paulo César Rodrigues

<https://www.linkedin.com/in/rodrigpc/>

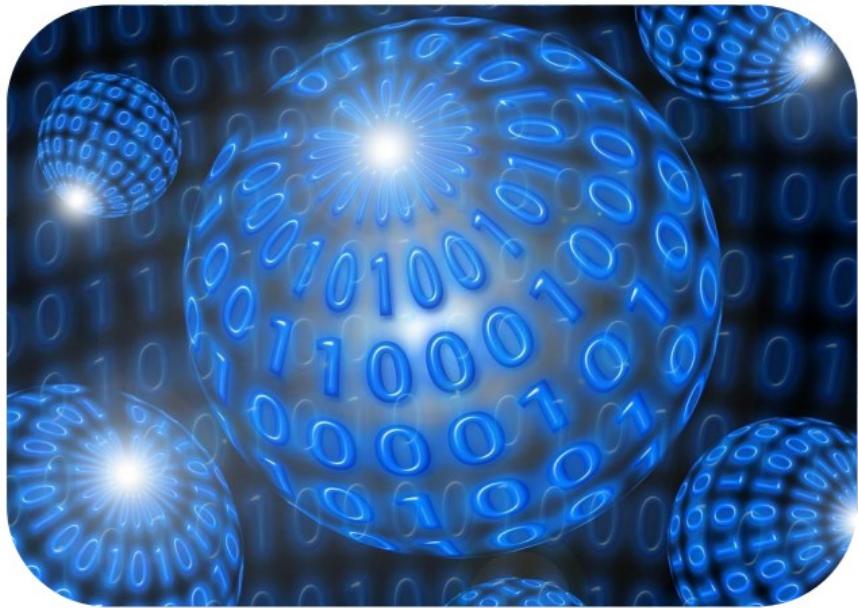
paulo.rodrigues@konfidentia.com

11-3280-6194



Conceituação

BIG DATA é um conjunto de metodologias utilizadas para capturar, armazenar e processar um volume imenso de informações de várias fontes (dados estruturados e não estruturados) com o objetivo de acelerar a tomada de decisão e trazer vantagem competitiva.



Os 7 V's do Big Data

VOLUME

VARIEDADE

VELOCIDADE

VISUALIZAÇÃO

VERACIDADE

VALOR

VULNERABILIDADE



Tipos de Dados – Dados Estruturados



Inicialmente, os modelos eram construídos com base em informações armazenadas em bancos de dados com **dados estruturados**.

Tipos de Dados – Dados Estruturados

	produto character varying(70)	valor numeric	segmento character varying(70)	data date
1	DVD	2.00	Papelaria e informática	2013-09-12
2	HD 500 GB	300.00	Papelaria e informática	2013-10-20
3	Tonner	250.00	Papelaria e informática	2013-11-01
4	Cadeira	50.00	Marcenaria	2013-09-19
5	Mesa	600.00	Marcenaria	2013-10-21
6	Armário	900.00	Marcenaria	2013-11-02
7	Corrimão	400.00	Serralheria	2013-09-12
8	Portão	1500.00	Serralheria	2013-10-22
9	Grade de proteção para janela	800.00	Serralheria	2013-11-03
10	Detergente	5.00	Limpeza e higiêne	2013-09-20
11	Desinfetante	40.00	Limpeza e higiêne	2013-11-23
12	Papel toalha	60.00	Limpeza e higiêne	2013-11-04

Tipos de Dados – Dados Semiestruturados

Um exemplo de arquivo com dados semiestruturados é o arquivo:
XML (eXtensible Markup Language)

```
<estoque>
    <item>
        <nome>Livro</nome>
        <preco>12</preco>
    </item>
    <item>
        <nome>Ventilador</nome>
        <preco>23</preco>
    </item>
    <item>
        <nome>Bolsa</nome>
        <preco>123</preco>
    </item>
</estoque>
```



Tipos de Dados – Dados Semiestruturados

Neste caso, os dados são irregulares com uma estrutura embutida. A estrutura dos dados é heterogênea.

Sua principal característica é a **facilidade de compartilhamento** de informações pela internet.



Tipos de Dados – Dados não estruturados

- Um dado não estruturado é um dado **sem uma estrutura pré-definida.**
 - **Textos** são exemplos de dados não estruturados. Podem ser oriundos de várias fontes como:



Tipos de Dados – Dados não estruturados



Definição – Machine Learning

Machine Learning pode ser traduzido simplesmente como **Aprendizado** (ou Aprendizagem) **de Máquina** (ou Computacional). O termo se refere a um enorme conjunto de técnicas que visam construir sistemas computacionais cujo comportamento seja definido com base em dados existentes.

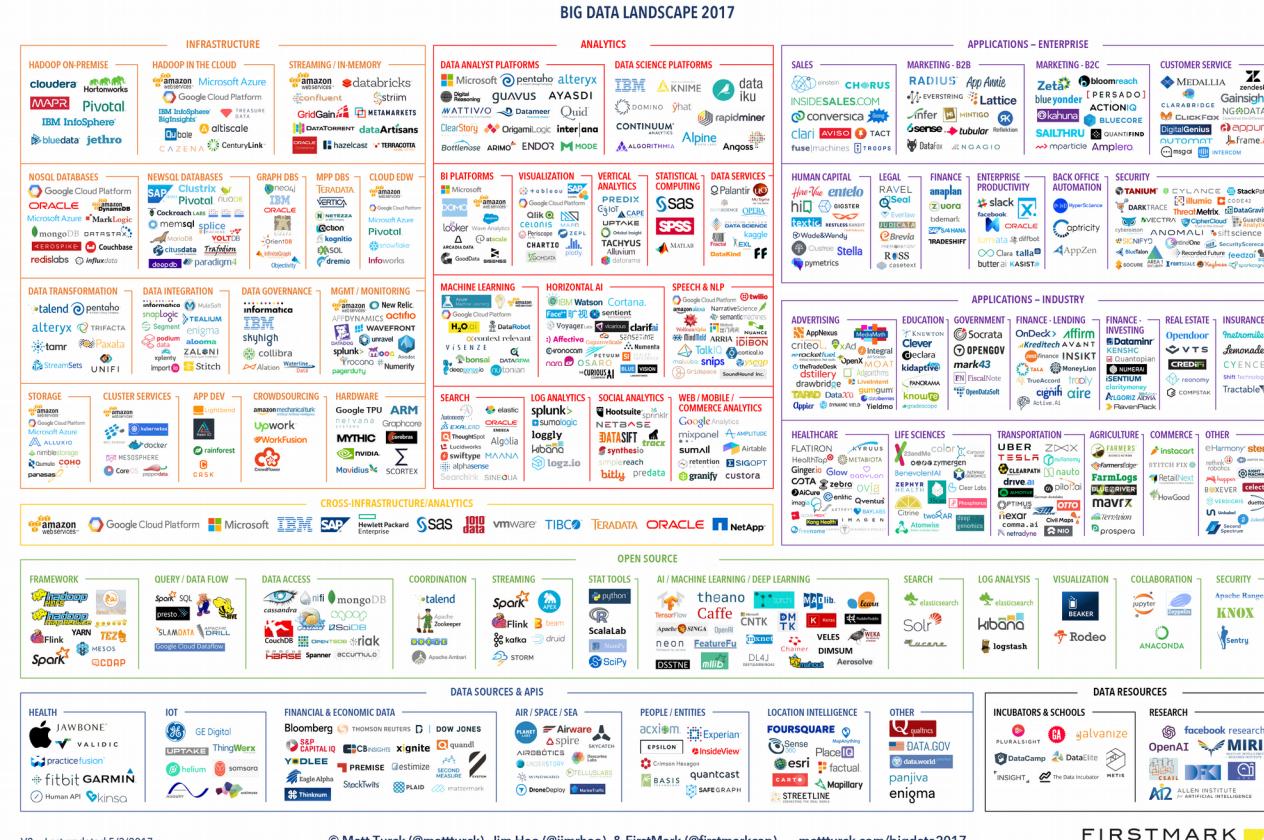
Como o comportamento do sistema não seria diretamente programado, mas sim adaptado de algum "conhecimento" previamente adquirido, essa abordagem teria similaridade com a forma como animais (entre eles, nós humanos) aprendem com a experiência.



Business Cases



Big Data & Machine Learning Landscape



paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop e Spark



O Profissional – Cientista de Dados

Um profissional de Big Data toma decisão com base em todas as informações disponíveis sobre o tema em estudo.

Este profissional considera as informações internas da empresa, as informações disponíveis na internet, nas redes sociais e nos dados gerados por sensores e imagem.



O Profissional – Cientista de Dados



O cientista de dados é o profissional adequado para trabalhar com Big Data.

O cientista de dados é um profissional que precisa capturar informação, armazenar informação e elaborar modelos para a adequada tomada de decisão.

Para realizar estas tarefas o profissional precisa ter conhecimento de programação, banco de dados, segurança de informação, tecnologias de Big Data e modelagem.

O Mercado de Big Data

CARGO (JOB TITLE)	2017		2018		%
APLICAÇÃO & INTEGRAÇÃO DE SISTEMAS (APPLICATION & SYSTEM INTEGRATION) (B)					
Gerente de Sistemas - Systems Manager	R\$ 13.000	- R\$ 25.000	R\$ 15.000	- R\$ 25.000	5,26%
Coordenador de Sistemas - Systems Coordinator	R\$ 9.500	- R\$ 13.000	R\$ 10.000	- R\$ 13.000	2,22%
Analista de sistemas Senior - Senior systems analyst	R\$ 5.200	- R\$ 8.000	R\$ 5.500	- R\$ 8.200	3,79%
Analista de sistemas Pleno - Systems analyst	R\$ 3.300	- R\$ 5.500	R\$ 3.500	- R\$ 5.700	4,55%
Analista de sistemas Junior - Junior systems analyst	R\$ 2.000	- R\$ 3.500	R\$ 2.200	- R\$ 3.500	3,64%
Arquiteto de aplicações - Applications architect	R\$ 8.500	- R\$ 12.000	R\$ 9.000	- R\$ 12.500	4,88%
Analista de Devops - Devops analyst	R\$ 4.000	- R\$ 11.000	R\$ 4.500	- R\$ 11.500	6,67%
Administrador de sistemas - Systems admin	R\$ 4.000	- R\$ 11.000	R\$ 4.500	- R\$ 11.200	4,67%
DBA	R\$ 3.000	- R\$ 9.500	R\$ 3.200	- R\$ 9.700	3,20%
BIG DATA (BIG DATA) (C)					
Especialista de Big Data / Cientista de dados - Big Data / Data Scientist specialist	R\$ 11.500	- R\$ 18.000	R\$ 12.000	- R\$ 22.000	15,25%
Analista de Big Data/ Cientista de dados - Big Data / Data Scientist analyst	R\$ 5.300	- R\$ 10.500	R\$ 5.500	- R\$ 12.500	13,92%
Especialista de BI - Business Intelligence specialist	R\$ 8.000	- R\$ 16.000	R\$ 10.000	- R\$ 16.500	10,42%
Analista de BI - Business Intelligence analyst	R\$ 4.000	- R\$ 10.000	R\$ 4.500	- R\$ 11.000	10,71%

O Mercado de Big Data

LOVE MONDAYS

Salários de Cientista de Dados na Nubank

[Ver todos os salários](#)

Ver salários publicados:

nos últimos 12 meses

nos últimos 24 meses

todos

[ATUALIZAR](#)

Todas as cidades

[FILTRAR](#)

Salário médio bruto

2 Postados

R\$ 24.406/mensal

min.

máx.

n/d

Gráfico de variação salarial



Não existem dados suficientes para exibir o gráfico.

O Mercado de Big Data

LOVE MONDAYS

Salários de Cientista de Dados

Último salário postado - 19/04/2018

Ver salários publicados:

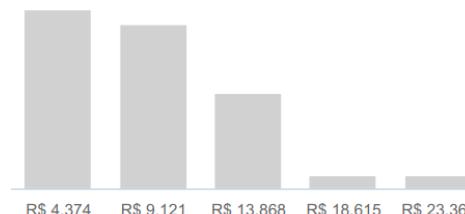
nos últimos 12 meses nos últimos 24 meses todos [ATUALIZAR](#)

Salário médio bruto

33 Postados

R\$ 9.974/mensal

min. R\$ 2.000 máx. R\$ 25.735



Salário	Postos
R\$ 4.374	~33
R\$ 9.121	~30
R\$ 13.868	~15
R\$ 18.615	~5
R\$ 23.362	~5

Gráfico de variação salarial

O Mercado de Big Data



The screenshot shows a web browser window with the URL <https://canaltech.com.br/carreira/cientista-de-dados-e-a-profissao-com-as-melhores-oportunidades-de-carreira-101818/>. The page title is "Cientista de dados é a profissão com as melhores oportunidades de carreira". The text discusses the data scientist profession being listed as one of the most relevant for the world economy by the World Economic Forum until 2020. It quotes Renato Souza from FGV EMAP, mentioning applications in various sectors like finance, education, health, agriculture, geology, and industry. A sidebar on the left has icons for search, video, audio, fix, shopping cart, and more.

Seguro | https://canaltech.com.br/carreira/cientista-de-dados-e-a-profissao-com-as-melhores-oportunidades-de-carreira-101818/

Home > Carreira

Cientista de dados é a profissão com as melhores oportunidades de carreira

Por Redação | 17 de Outubro de 2017 às 13h48

A carreira de cientista de dados foi listada pelo Fórum Econômico Mundial como uma das mais relevantes para o mercado até 2020. Para o professor da Escola de Matemática Aplicada da Fundação Getúlio Vargas (FGV EMAP), Renato Souza, as possibilidades de atuação estão nos mais diversos setores, como finanças, educação, saúde, agricultura, geologia e indústria.

"Qualquer empresa que gere dados pode contratar um profissional para analisá-los e tomar decisões com base em informação, não na intuição. Vamos ver as aplicações disso no dia a dia de governos, sociedade, hospitais e indústrias. O Brasil está entre os grandes produtores e consumidores de informação e, de maneira geral, tem iniciativas nessa área pipocando no mundo todo", afirma Renato Souza.

Conteúdo

CONTEÚDO PROGRAMÁTICO

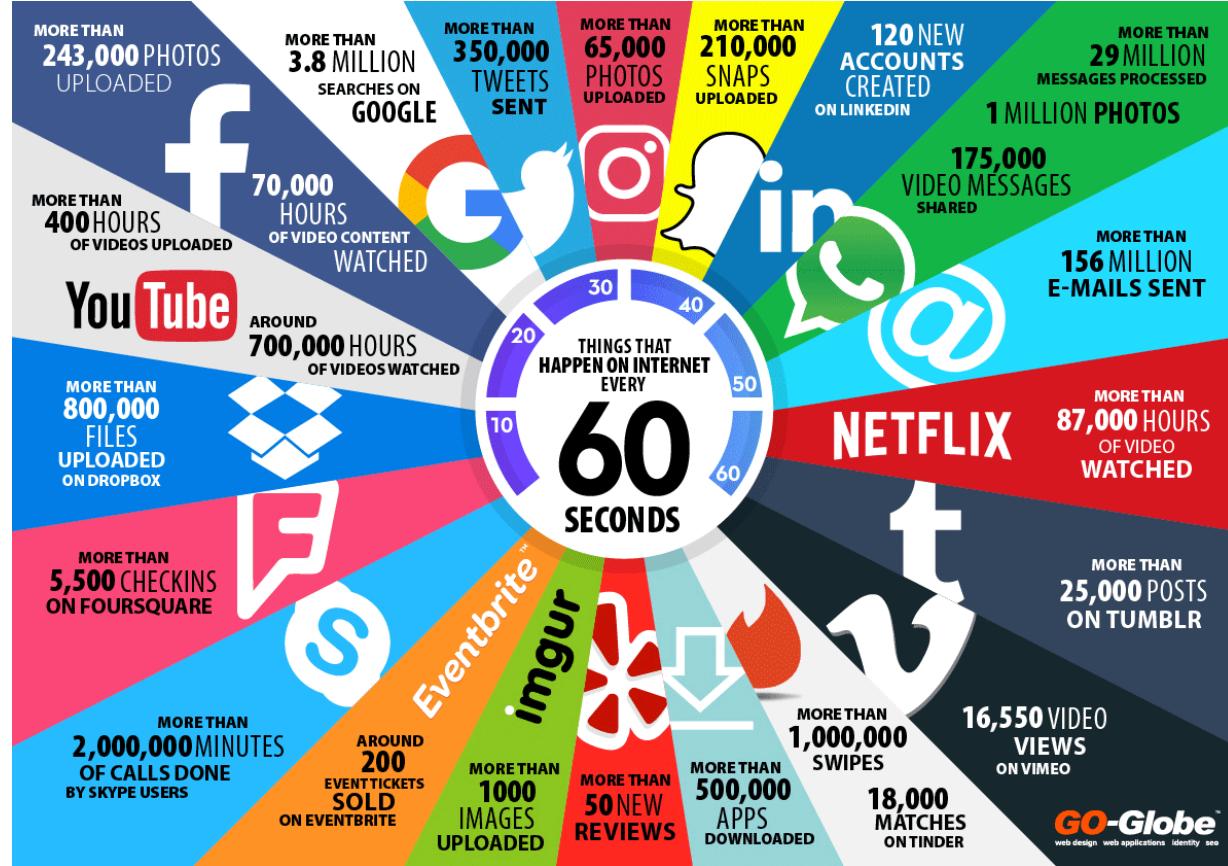
- Visão geral da ciência de dados e aprendizado de máquina em escala
- Visão geral do ecossistema do Hadoop
- Instalação de um Cluster Hadoop
- Trabalhando com dados do HDFS e tabelas do Hive usando o Hue
- Visão geral do Python
- Visão geral do R
- Visão geral do Apache Spark 2
- Leitura e gravação de dados
- Inspeção da qualidade dos dados
- Limpeza e transformação de dados
- Resumindo e agrupando dados
- Combinando, dividindo e remodelando dados
- Explorando dados
- Configuração, monitoramento e solução de problemas de aplicativos Spark
- Visão geral do aprendizado de máquina no Spark MLlib
- Extraíndo, transformando e selecionando recursos
- Construindo e avaliando modelos de regressão
- Construindo e avaliando modelos de classificação
- Construindo e avaliando modelos de cluster
- Validação cruzada de modelos e hiperparâmetros de ajuste
- Construção de pipelines de aprendizado de máquina
- Implantando modelos de aprendizado de máquina

MATERIAL DIDÁTICO

- Slides do treinamento em PDF
- GitHub com exercícios e códigos exemplo
- Máquinas virtuais para exercícios simulados
- Gravação das aulas disponível durante 3 meses



Por que?



Skills

MODERN DATA SCIENTIST

Data Scientist, the sexiest job of the 21th century, requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.



MATH & STATISTICS

- ★ Machine learning
- ★ Statistical modeling
- ★ Experiment design
- ★ Bayesian inference
- ★ Supervised learning: decision trees, random forests, logistic regression
- ★ Unsupervised learning: clustering, dimensionality reduction
- ★ Optimization: gradient descent and variants

PROGRAMMING & DATABASE

- ★ Computer science fundamentals
- ★ Scripting language e.g. Python
- ★ Statistical computing packages, e.g. R
- ★ Databases: SQL and NoSQL
- ★ Relational algebra
- ★ Parallel databases and parallel query processing
- ★ MapReduce concepts
- ★ Hadoop and Hive/Pig
- ★ Custom reducers
- ★ Experience with xaaS like AWS

DOMAIN KNOWLEDGE & SOFT SKILLS

- ★ Passionate about the business
- ★ Curious about data
- ★ Influence without authority
- ★ Hacker mindset
- ★ Problem solver
- ★ Strategic, proactive, creative, innovative and collaborative

COMMUNICATION & VISUALIZATION

- ★ Able to engage with senior management
- ★ Story telling skills
- ★ Translate data-driven insights into decisions and actions
- ★ Visual art design
- ★ R packages like ggplot or lattice
- ★ Knowledge of any of visualization tools e.g. Flare, D3.js, Tableau

MarketingDistillery.com is a group of practitioners in the area of e-commerce marketing. Our fields of expertise include: marketing strategy and optimization, customer tracking and on-site analytics, predictive analytics and econometrics, data warehousing and big data systems: marketing channel insights in Paid Search, SEO, Social, CRM and brand.

Marketing
DISTILLERY
(c) Krzysztof Latawski

paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



Times

Para trabalhar com Big Data é necessário muito mais do que o cientista de dados, as empresas hoje já possuem uma visão evoluída acerca deste tema e vem buscando montar equipes de Big Data.

Estas equipes tendem a trazer mais resultados quando multidisciplinares, normalmente são compostas pelo cientista de dados, um profissional de infraestrutura\DBA, um engenheiro de dados, um gerente de projetos e um especialista funcional.



Times



The Big Data Dream Team

Dr. Raúl Arrabales

Academic / Innovation Director

@MbitSchool



paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



Times

Demystifying the Big Data Superheroes Mindset

Gifted Individuals
Vs.
Talented Teams

How do Big Data Projects Succeed?

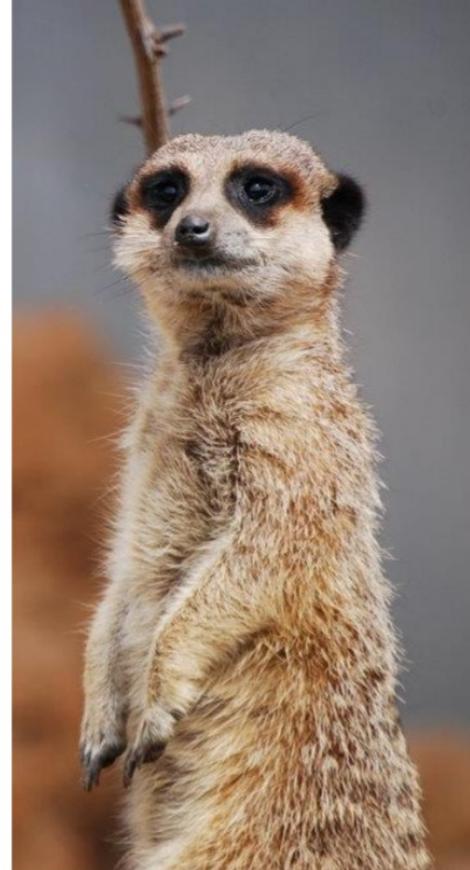
- Team
- Technology
- Project Mngmt
- Business Objective!
- Alignment!!



Business Analyst

- Consultancy background
- Degree in Business
- Strong IT literacy
- Desirably MBA

BA understands data life-cycle and context, identifies opportunities, boosts profit, improves competitiveness, reduces costs.



Times



Project Manager

- PM background
- Degree in IT/Engineering
- Strong IT literacy
- Agile/Lean Methodologies
- Desirably PMP/PRINCE2

Synergy/comm/flow of the team. Facilitates innovation and team talent growth. Manages customer expectations & requirements and ensures quality in solutions delivery.

Big Data Architect

- Storage/DB/Cloud background
- Degree in CS/IT/Engineering
- Datacenter/DWH Architecture
- OS/Network Administration.
- NoSQL, MapReduce, Shell Script

Designs and optimize reliable, robust, fail-tolerant, safe, on demand, elastic computer systems infrastructure that enables big data applications to run and be delivered smoothly.



Times



Big Data Developer

- Software Eng. background
- Degree in Computer Science
- Hardcore coder / Dif. Lang.
- NoSQL, Hadoop API, MR APIs
- Backend developer, scripting
- Real-time / in-memory app.

**Develops and integrates
software applications to run
efficiently and process petabytes
of data across big data clusters.**

Data Scientist

- Data mining background
- Degree in CS/Statistics/Math
- Hardcore algorithm designer/coder
- Analytical problem solver
- Expert in Machine Learning
- Ideally Ph.D. in CS/AI/Analytics

Able to develop/design/apply new efficient algorithms to extract value from big data. Understands the opportunities hidden in massive datasets.



Times



Cybersecurity Engineer

- Information Security Backgr.
- Degree in Computer Science.
- Ideally M.Sc./Ph.D. Security.
- IDS, Datacenter, SysAdm.
- Backend and frontend security.

Able to assure the security of information across all big data platform and infrastructures, detecting and foreseeing cyber attacks end to end.

Times



Folks from other lands

- Members of the board.
- Legal affairs. Digital ecosystem.
- BI/Datawarehouse Mngmt.
- IT infrastructures Mngmt.
- Business Partners.
- Data Owners & Agencies.
- Users & Customers!!

Seamless deployment and integration coordinated with operations, delivery, etc. And external stakeholders.

Times

Teamwork & Specialization



Arquitetura e Componentes

Data Lake é um termo recente, Lago de Dados em tradução livre, trata-se de uma área de armazenamento de dados estruturados ou não.

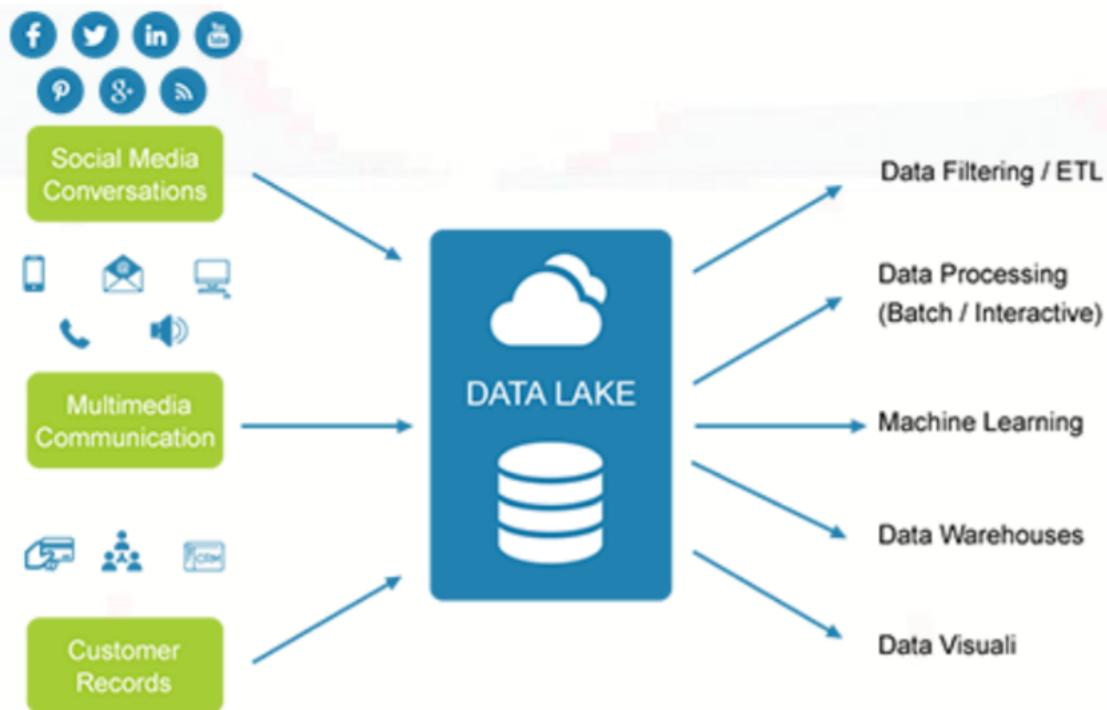
O Data Lake é um ponto essencial para a arquitetura de um ambiente Big Data, ele é o ponto de centralização dos dados da empresa e fonte de dados para os engenheiros e cientistas de dados.



Arquitetura e Componentes

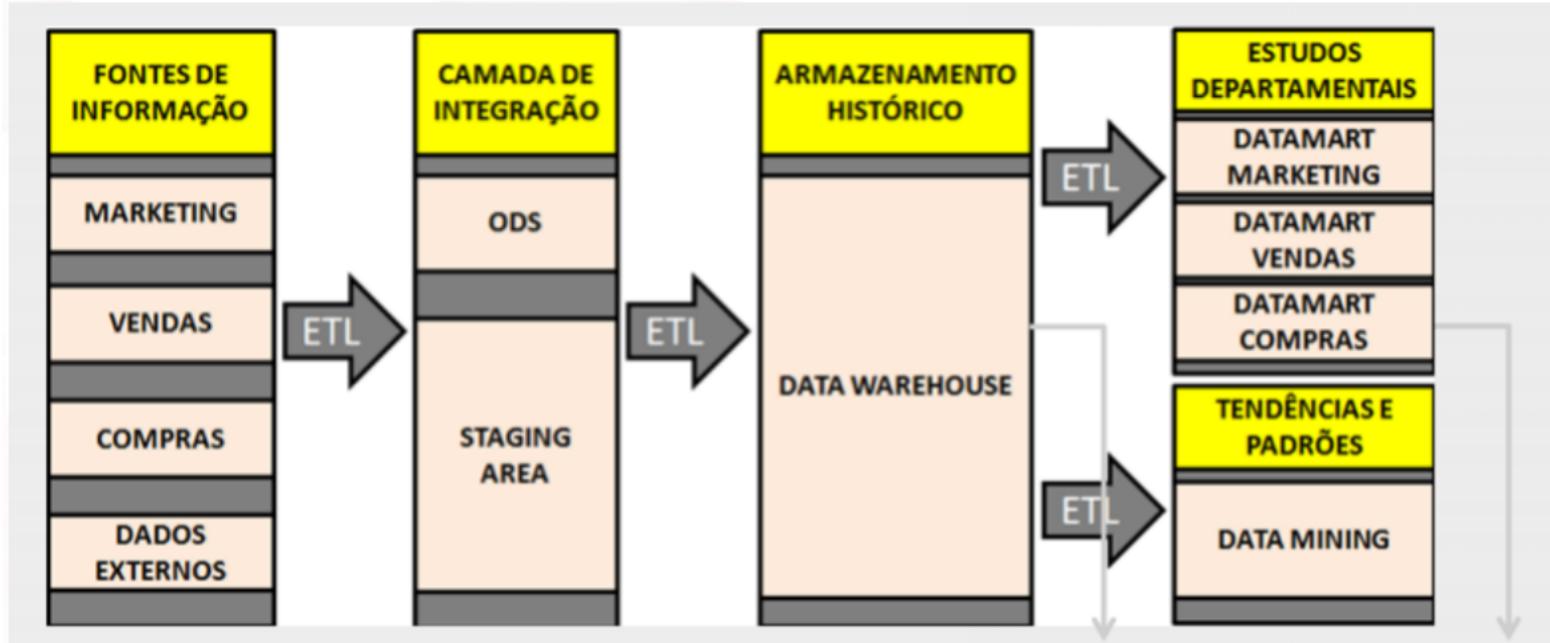
«DataLake tem como objetivo armazenar dados de diversos tipos e origens diferentes para ser utilizado como fonte principal de dados.

Ele também recebe os dados em seu formato original com o objetivo de otimizar o desempenho da ingestão dos dados.

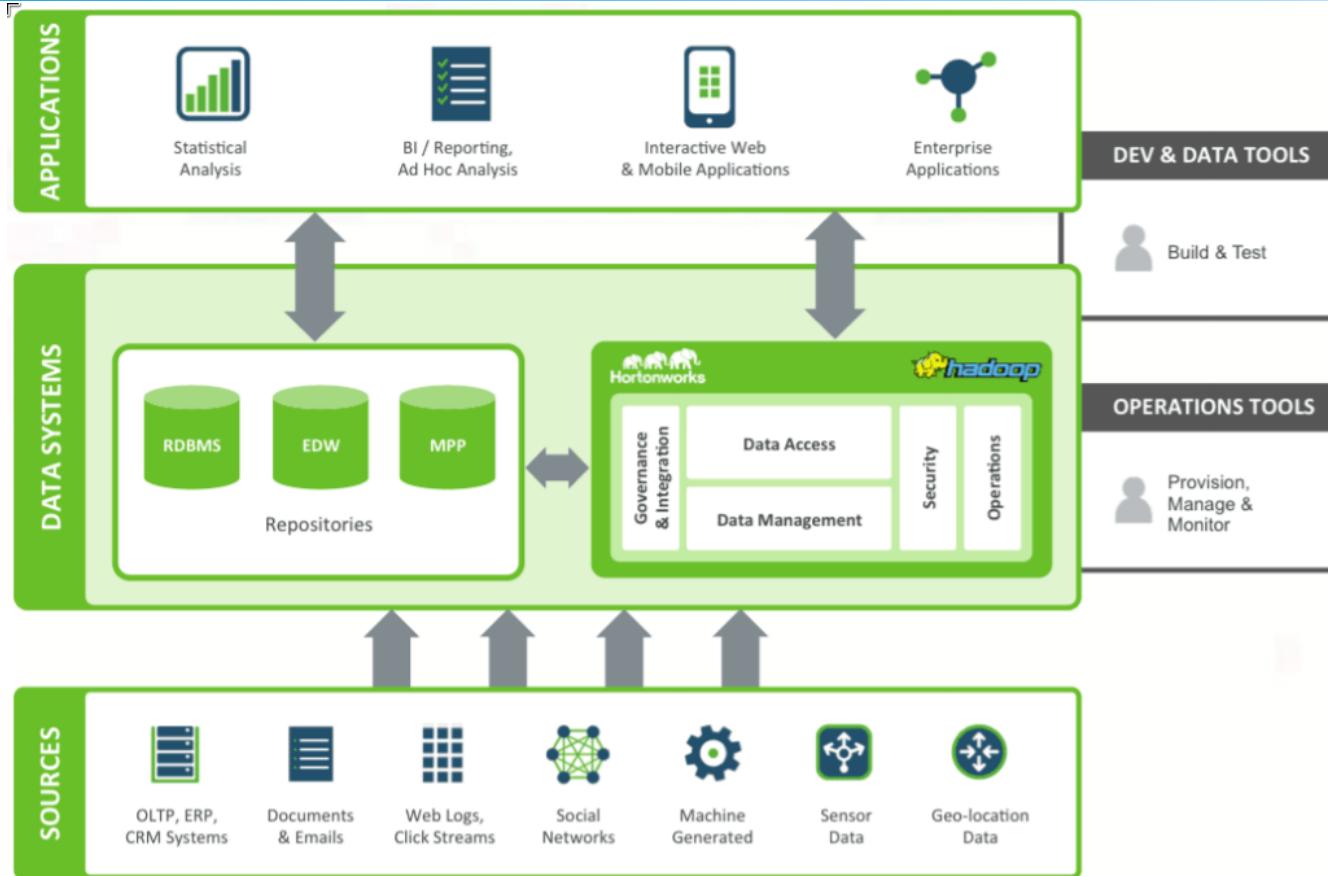


Arquitetura e Componentes

Arquitetura DataWare House tradicional

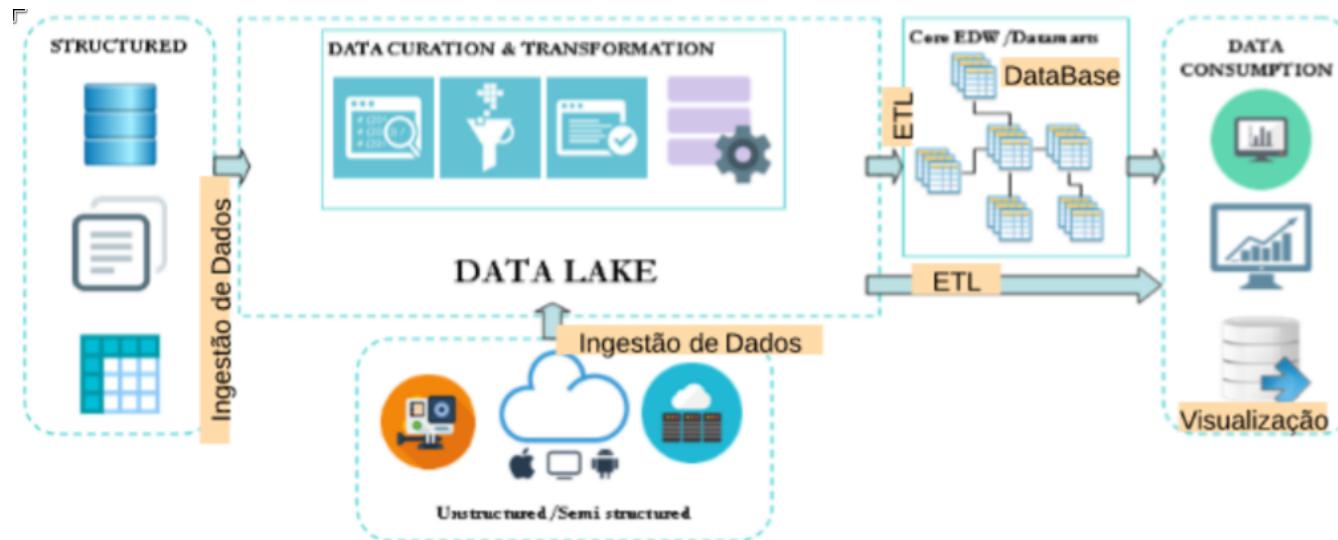


Arquitetura e Componentes



Arquitetura e Componentes

Agora, é hora de entender suas necessidades, objetivos e pesquisar as ferramentas mais adequadas, a seguir temos uma relação com as principais ferramentas de acordo com as etapas da arquitetura básica de Big Data aqui apresentada. Sendo estas Ingestão de Dados, DataLake, ETL, DB's\Datamart's e Visualização.



Ferramentas de Ingestão de Dados



Data Lake



ETL



talend*



alteryx

Database / Datamart



Visualização



Streaming Processing



Apache Flink



SQL Data Flows / Pipelines



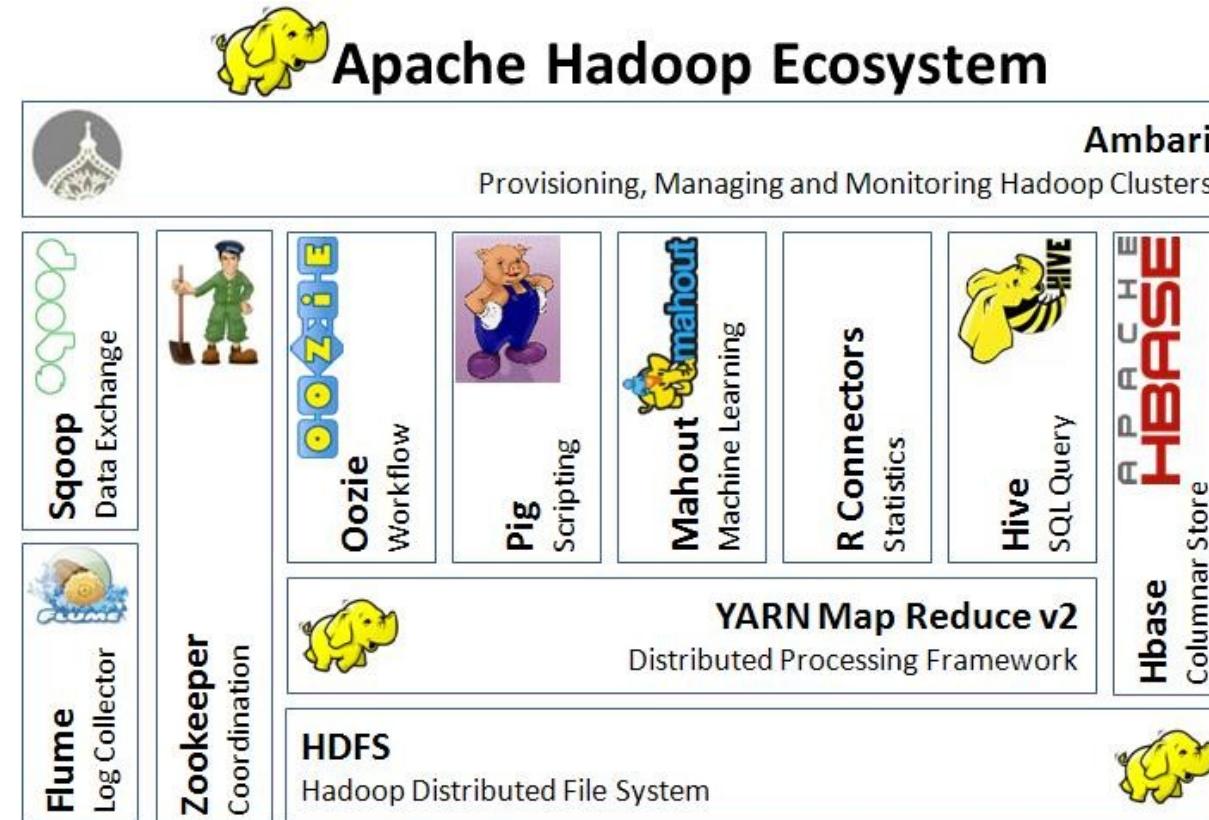
Apache Software Foundation

A Apache Software Foundation (ASF) é uma organização sem fins lucrativos criada para prover suporte a projetos de código aberto, principalmente os Softwares Apache (Apache HTTP Server).

Os softwares criados pela fundação Apache são distribuídos como Software Livre e com a licença Apache.



Apache Software Foundation



Distribuições

cloudera

databricks



DATASTAX

Visão Geral



Apache Hadoop é um FrameWork de Software aberto distribuído através da Apache Software Foundation (ASF), tem como objetivo o armazenamento e processamento de dados de forma distribuída. Ele permite o armazenamento e processamento de grandes volumes de dados utilizando hardware commodity através de seu poder de distribuição.

Dentre os serviços e funcionalidades do Hadoop é possível além de armazenar e processar, realizar gestão, governança, segurança e operações de dados.

O Hadoop foi desenvolvido pela Yahoo e posteriormente entregue para a Apache.

História



- 2003 Google publica artigo do GFS (SOSP'03)
- 2004 Google publica artigo do MapReduce (OSDI'04)
- 2005 Doug Cutting cria uma versão do MapReduce para o projeto Nutch
- 2006 Hadoop se torna um subprojeto do Apache Lucene
- 2007 Yahoo! Inc. se torna o maior contribuidor e utilizador do projeto (aglomerado com mais de 1.000 nós)
- 2008 Hadoop deixa a tutela do projeto Lucene e se transforma em um projeto top-level da Apache
- 2010 Facebook anuncia o maior aglomerado Hadoop do mundo (mais de 2.900 nós e 30 petabytes de dados)
- 2011 Apache disponibiliza a versão 1.0.0

História

O Ecossistema Hadoop é desenvolvido em Java, linguagem orientada a objetos desenvolvida em meados da década de 90, um dos diferenciais dessa linguagem é que ela utiliza uma Maquina Virtual que interpreta seus códigos compilados em bytecode, diferente de outras linguagens que são compiladas diretamente para a linguagem nativa do Sistema Operacional.

A Maquina Virtual que interpreta os códigos Java é conhecida como JVM e é responsável também pela execução do Hadoop.



HDFS – Hadoop Distributed File System

Trata-se do gerenciador de arquivos do Hadoop, é um dos subprojetos dentro do projeto Hadoop e tem por objetivo permitir o armazenamento de grandes quantidades de dados de forma distribuída.

É altamente recomendado para quantidades de armazenamento em proporções de Tera e PetaBytes. O HDFS armazena os dados de maneira contínua e o acesso aos dados é realizado por modo de fluxo, ou seja os dados são resgatados através de comandos MapReduce.



HDFS– Hadoop Distributed File System

O HDFS possui diversas características específicas que o diferenciam de outros sistemas de arquivos distribuídos. A característica que mais chama atenção é o fato dele trabalhar sob o modelo WORM (Write-Once-Ready-Many), o uso deste modelo permite que ele não dependa de forte controle de simultaneidade, simplifica a persistência dos dados e habilita o acesso de alto rendimento.



HDFS– Hadoop Distributed File System

Outra característica positiva do HDFS é que ele orienta o processamento a ser realizado o mais próximo dado possível, ao invés de movimentar o dado para um nó específico e para realizar o processamento, ou seja se você precisa processar o Dado 1 e o Dado 5 e estes estão localizados nos Nós 1 e 5 consecutivamente, não há necessidade de movimentar o Dado 1 e o Dado 5 para o Master realizar o processamento. O próprio nó 1 deve processar o Dado 1 enquanto o Nó 5 irá processar o Dado 5 e paralelamente ambos retornaram o resultado desejado, reduzindo assim o tempo de processamento de acordo com o quanto espalhados estão os dados nos Cluster.



HDFS– Objetivos

- Tolerância a falhas pela detecção de falhas e aplicação de recuperação rápida, automática;
- Acesso a dados por meio do fluxo MapReduce;
- Modelo de simultaneidade simples e robusto;
- Lógica de processamento próxima aos dados, ao invés dos dados estarem próximos à lógica de processamento;
- Portabilidade entre sistemas operacionais e hardware padrão heterogêneos;



HDFS– Objetivos

- Escalabilidade para armazenar e processar de modo confiável grandes quantidades de dados;
- Economia pela distribuição de dados e pelo processamento entre clusters de computadores pessoais padrão;
- Eficiência pela distribuição de dados e pela lógica para processá-los em paralelo nos nós em que os dados estão localizados;
- Confiabilidade pela manutenção automática de várias cópias dos dados e pela reimplementação automática da lógica de processamento no caso de falhas;



Hadoop 2.x vs 3.x

Attributes	Hadoop 2.x	Hadoop 3.x
Handling Fault-tolerance	Through replication	Through erasure coding
Storage	Consumes 200% in HDFS	Consumes just 50%
Scalability	Limited	Improved
File System	DFS, FTP and Amazon S3	All features plus Microsoft Azure Data Lake File System
Manual Intervention	Not needed	Not needed
Scalability	Up to 10,000 nodes in a cluster	Over 10,000 nodes in a cluster
Cluster Resource Management	Handled by YARN	Handled by YARN
Data Balancing	Uses HDFS balancer for this purpose	Uses Intra-data node balancer



Componentes mais usados

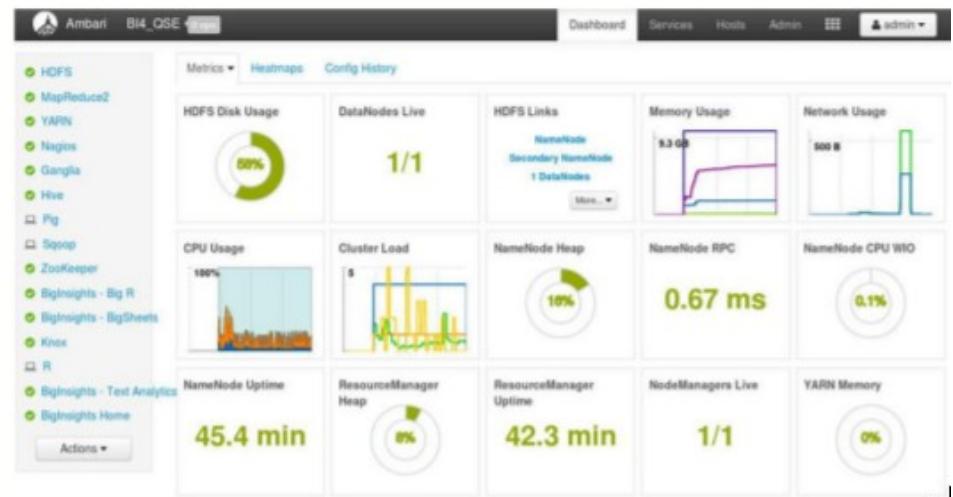
Ambari

Tem o objetivo de simplificar a monitoração e gerenciamento dos Hadoop Clusters.

Por padrão a página inicial do Ambari trás o status operacional de seu cluster, além disso ainda por padrão o Ambari faz exibição das métricas do HDFS, YARN e Hbase. Também é possível incluir ou remover widgets otimizando sua monitoração do ambiente.



Apache
Ambari

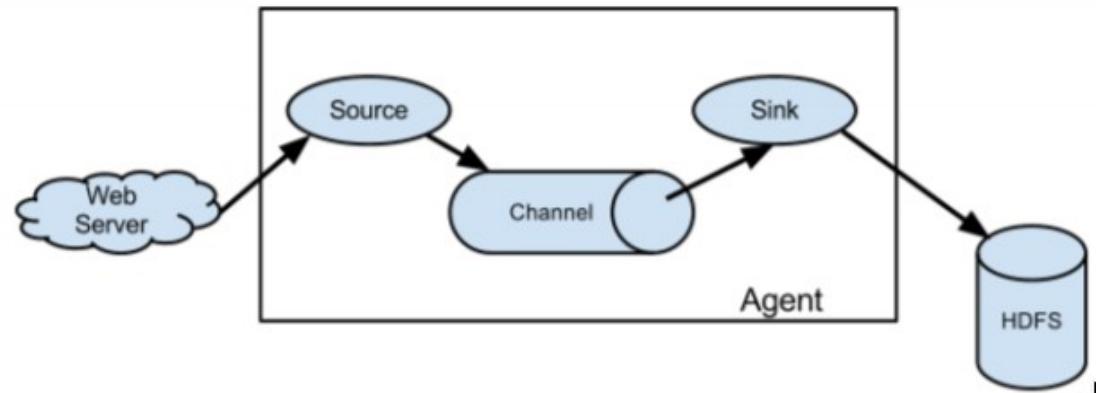


Componentes mais usados

Serviço para coleta, agregação e transporte de grandes quantidades de dados de log para o HDFS. Foi criado especificamente para a ingestão de dados para o HDFS, permite algumas transformações no processo de ingestão como a agregação de dados



O flume é indicado não apenas na ingestão de dados de logs, mas para qualquer fluxo dados gerado em timeseries e grandes quantidades como informações de redes sociais por exemplo.



Componentes mais usados

HBase



Banco de dados NoSQL orientado a colunas, que roda em cima do HDFS, pode ser utilizado como fonte ou destino do para as funções de MapReduce. Permite o particionamento dos dados através do cluster e é indicado para o armazenamento e análise de grandes massas de dados.

Ao trabalhar com o Hbase, deve-se tomar muito cuidado com a modelagem dos dados, das chaves das famílias de colunas e suas partições.

	fname	lname	picture
aputrell@apache.org	"Andrew"	"Putrell"	
jdcryans@apache.org	"Jean-Daniel"	"Cryans"	downfall.jpg
stack@apache.org		"Stack"	dancing_stack.jpg
todd@apache.org	"Todd"	"Lipcon"	turbo.jpg

Row Key identifies a row across Column Families

Column Families store frequently accessed data together. Settings can be customized on a per Column Family basis

Componentes mais usados

Sistema de Data Warehouse para o Hadoop que facilita summarização de dados e queries adhoc. Linguagem similar ao SQL. De forma simplificada o Hive é uma camada de abstração onde é possível criar comandos SQL e ele converte em operações Map Reduce que são executadas nas famílias de colunas do Hbase. Permitindo assim maior facilidade na hora de buscar dados para questões pontuais e isoladas



Componentes mais usados

『Biblioteca escalável de machine learning e mineração de dados

Através d Mahout desenvolvedor e cientistas de dados podem realizar analise avançadas aplicando conceitos de machine learning aproveitando a escalabilidade do framework Hadoop. Está é uma ferramenta especifica para analise avançadas de grandes massas de dados.



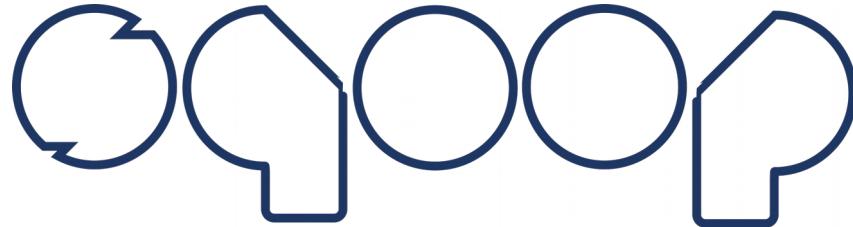
```
frank@frankthetank:~$ mahout
no HADOOP_HOME set, running locally
An example program must be given as the
first argument.
Valid program names are:
arff.vector: : Generate Vectors from an ARFF
               file or directory
canopy: : Canopy clustering
cat: : Print a file or resource as the
               logistic regression models would see
               it
...
...
```

Componentes mais usados

Pig é uma linguagem de alto nível específica para a criação de processos Map Reduce, possui uma sintaxe simples e de fácil aprendizado. Permite não apenas realizar os processos de Map Reduce, mas também alguma transformação nos dados trabalhados. Existem alguns casos em que o Pig é utilizado como ETL para disponibilizar dados do Hbase ou HDFS para DataMarts específicos.



Componentes mais usados



『Ferramenta para importar dados de banco de dados relacionais para o Hadoop e vice-versa.

É utilizada especificamente para a ingestão de dados para o HDFS ou Hbase, porém o sqoop foi criado especificamente para trabalhar com bancos de dados relacionais, ou seja para realizar a ingestão de dados estruturados de bases Oracle, MsSQL, PostGreSQL, entre outro o Sqoop é mais indicado que o Flume.

Sqoop Import Examples

- `Sqoop import --connect jdbc:oracle:thin:@//dbserver:1521/masterdb
--username hr --table emp
--where "start_date > '01-01-2012'"`
- `Sqoop import
jdbc:oracle:thin:@//dbserver:1521/masterdb
--username myuser
--table shops --split-by shop_id
--num-mappers 16`

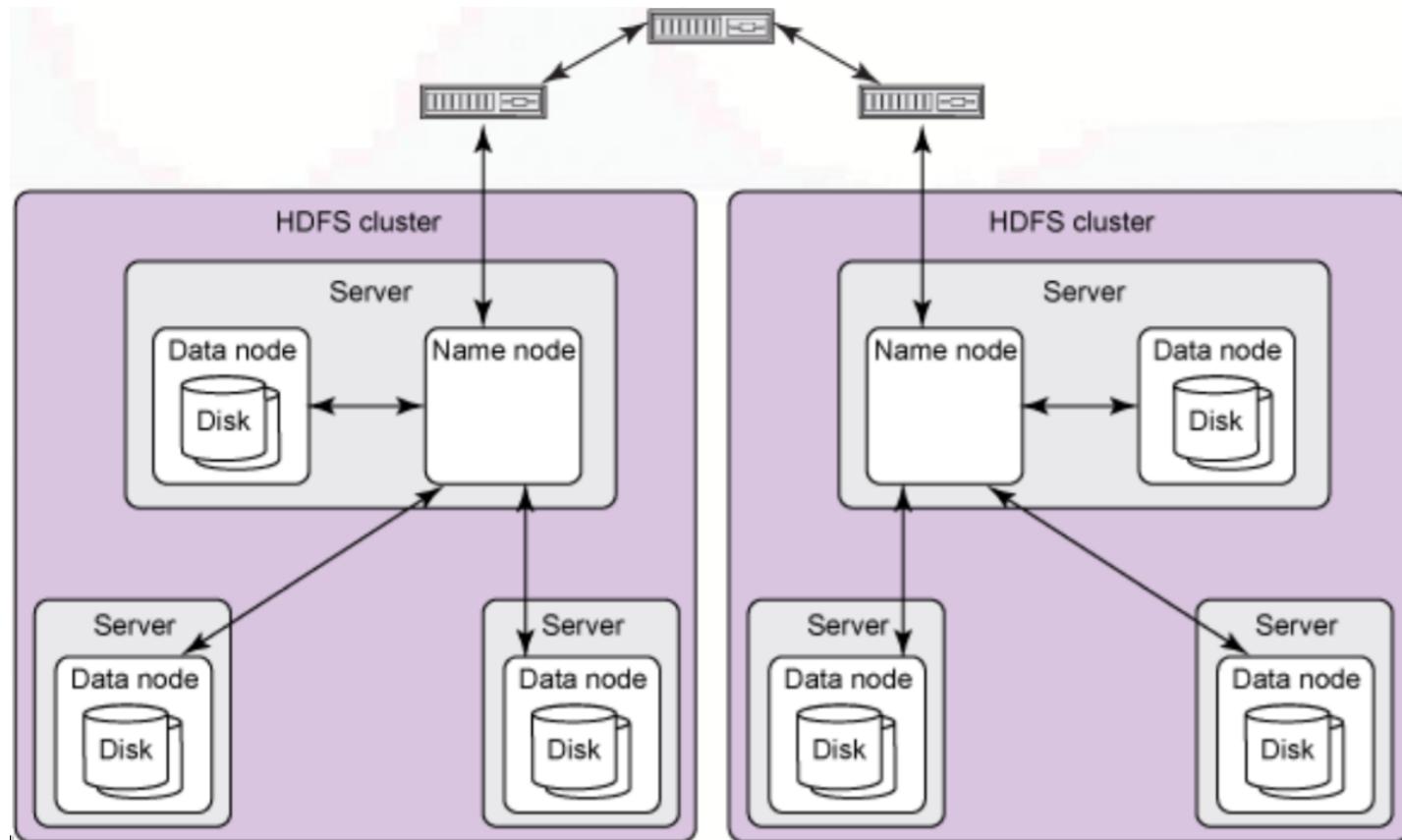
Must be indexed or partitioned to avoid 16 full table scans

Componentes mais usados



- 『 Serviço de coordenação de alto desempenho para sistemas distribuídos.
Basicamente é o responsável por manter todos os serviços do Ecossistema Hadoop em pleno funcionamento.
ZooKeeper resolve este problema com a sua arquitectura simples e API. Permite que os desenvolvedores se concentrem na lógica do aplicativo principal sem se preocupar com a natureza distribuída da aplicação.
 - Confiabilidade - falha de um único ou alguns sistemas não faz todo o sistema falhe.
 - Escalabilidade - O desempenho pode ser aumentado como e quando necessário, adicionando mais máquinas com pequena alteração na configuração do aplicativo sem tempo de inatividade.
 - Transparência - Oculta a complexidade do sistema e mostra-se como uma única entidade / aplicação.

HDFS - Arquitetura



MapReduce - Conceitual

Definição: É um modelo de programação para processamento de dados com chave e valor. Não é uma linguagem ou uma plataforma. Foi inspirado no paper MapReduce do Google - 2004 .

- Consiste basicamente em três etapas:
 - » Map: Extrai algo que você espera de cada registro.
 - » Shuffle and Sort: Embaralha e ordena os registros.
 - » Reduce: Agrega, summariza, filtra ou transforma os dados.
- Objetivo: Facilitar a distribuição das tarefas e execução em paralelo entre os nodes de um cluster Hadoop.



MapReduce - Conceitual

- Motivações:
 - Facilitar o desenvolvimento e execução de aplicações utilizando processamento paralelo.
 - Processar um volume massivo de dados utilizando uma infraestrutura de hardware commodity.



MapReduce - Conceitual

Podemos também definir o MapReduce como um Pseudo Código das funções Map e Reduce com o objetivo de encontrar frequências de palavras ou conjuntos de caracteres.

Map

```
map (String fileName, String document) {  
    List <String> T = tokenize(document)  
    for each token in T {  
        emitIntermediate ( token, 1)  
    }  
}
```

Reduce

```
reduce (String token, List<Integer> values) {  
    Integer sum = 0  
    for each value in values {  
        sum = sum + value  
    }  
    emit ( token, sum)  
}
```

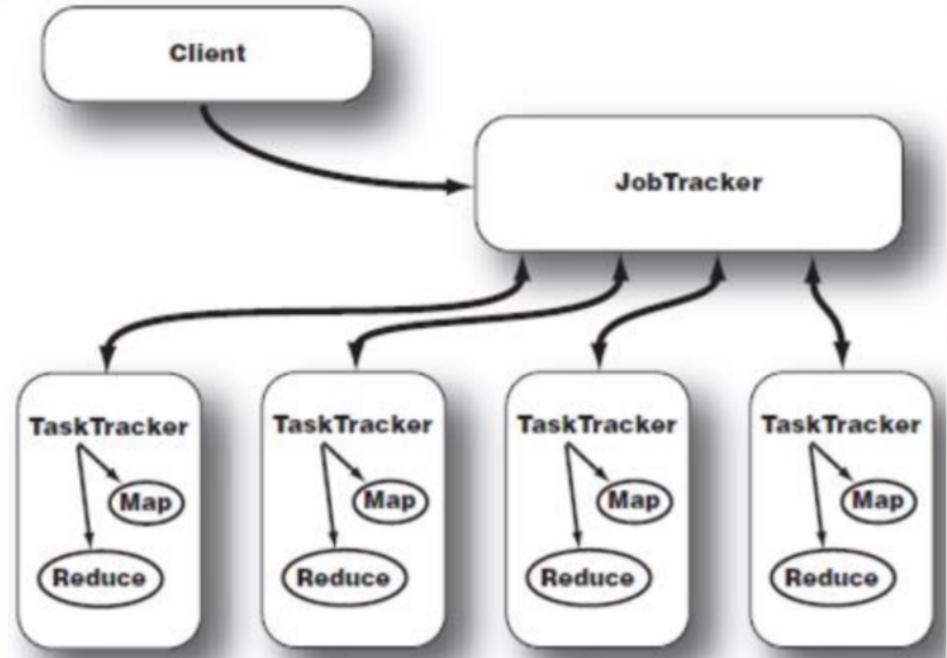
Map em Java

```
import java.io.BufferedReader;  
import java.util.Scanner;  
import java.util.StringTokenizer;  
public class Mapper {  
    private void map () {  
        Scanner sc = new Scanner(new BufferedReader(System.in));  
        String line = "";  
        while (sc.hasNextLine()) {  
            line = sc.nextLine();  
            StringTokenizer tokens = new StringTokenizer(line);  
            while(tokens.hasMoreTokens()){  
                emitIntermediate(tokens.nextToken(), "1");  
            }  
        }  
        sc.close();  
    }  
    private void emitIntermediate (String w, String um) {  
        System.out.println(w + "\t" + um);  
    }  
    public static void main(String[] args) {  
        new Mapper().map();  
    }  
}
```

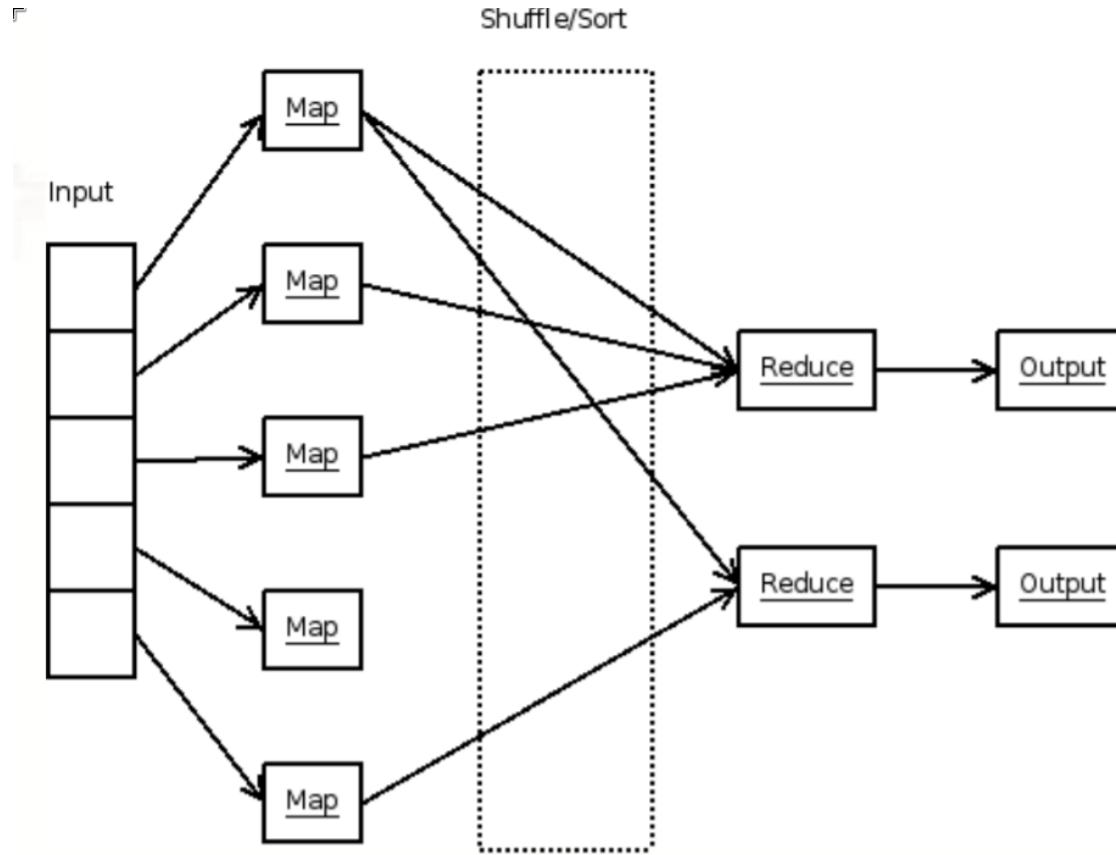
MapReduce - Conceitual

Componentes da arquitetura:

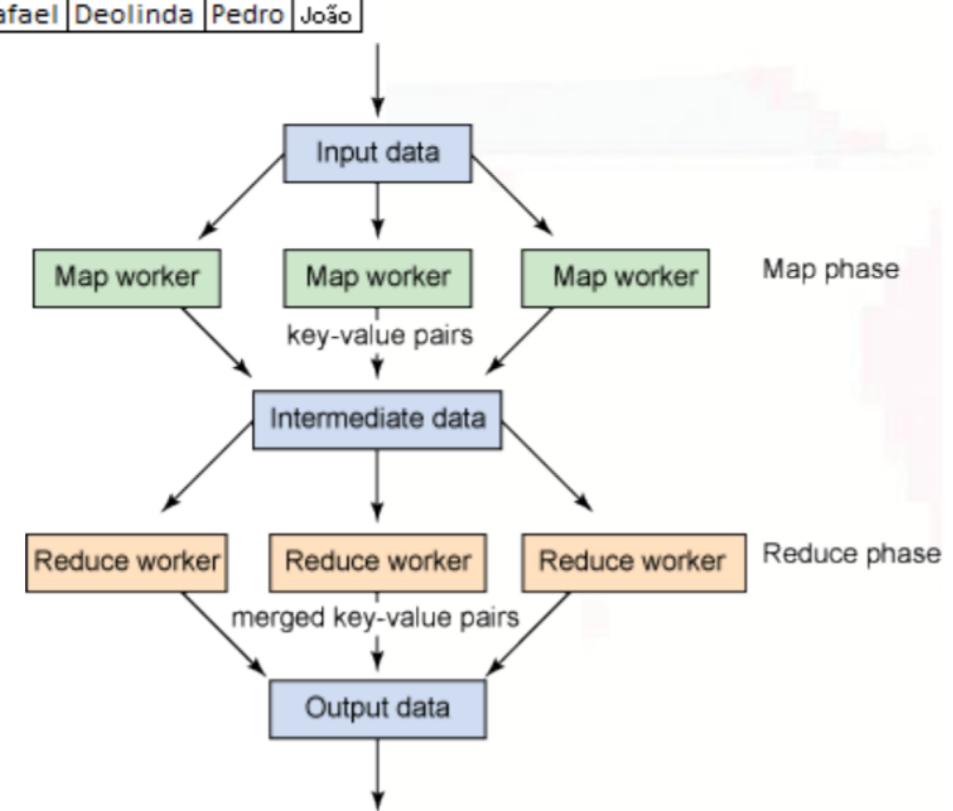
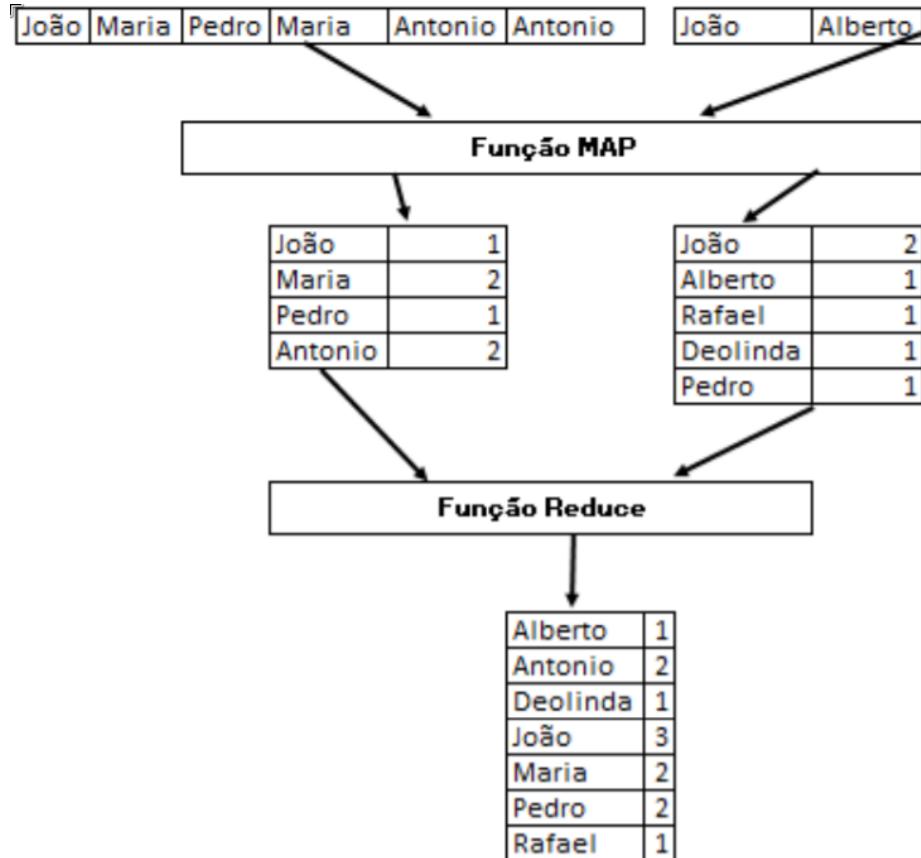
- **JobTracker:** Gerencia os Jobs e distribui as Tasks entre os TaskTrackers.
- **TaskTrackers:** Inicia e monitora a execução individual das Tasks de Map e Reduce.



MapReduce - Conceitual



MapReduce - Conceitual



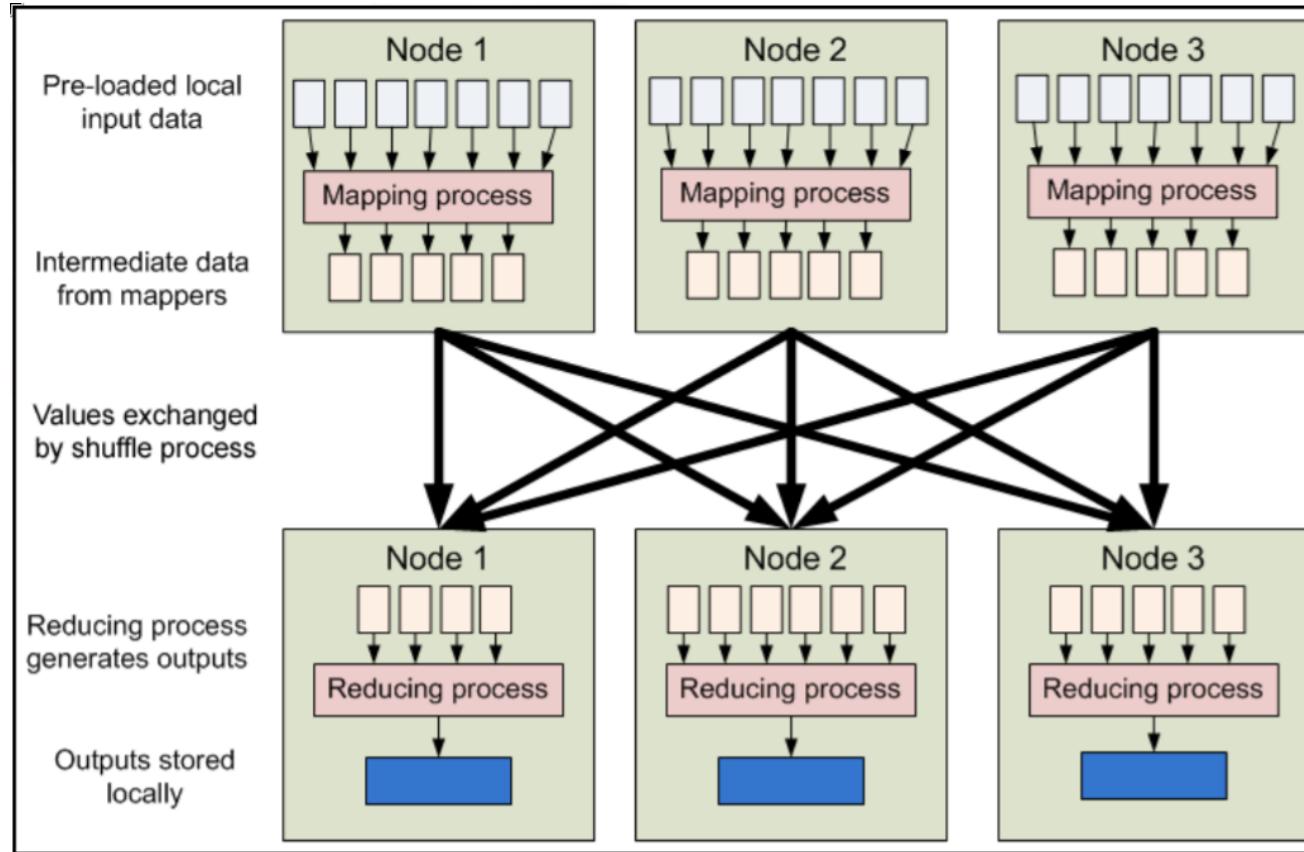
MapReduce no Cluster

O MapReduce por ser um algoritmo orientado ao processamento paralelo e distribuído, possui uma vantagem natural ao ser executado em ambientes Clusterizados. Este é o principal motivo pelo qual o MaReduce é uma das principais funções do Hadoop.

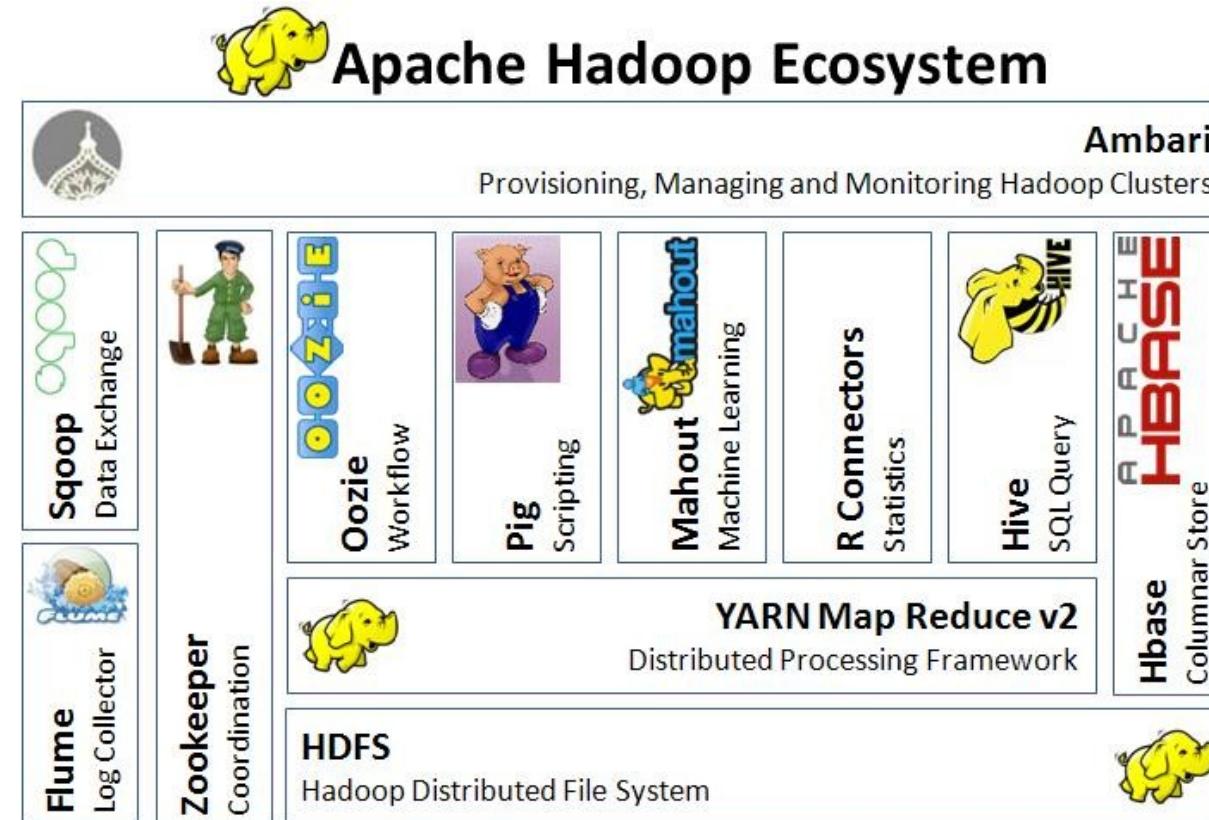
Ele remove pontos de “gargalo” que aplicações de processamento centralizado possuem, como por exemplo o limite de CPU ou de áreas de log. Além disso ele utiliza do Cluster para garantir que você nunca irá perder seu processamento por completo. Por exemplo se um servidor de processamento centralizado perde conexão ou cai por qualquer motivo, você irá perder todos os dados já processados e terá de reiniciar seu processo, enquanto utilizando o MapReduce em ambientes clusterizados caso um node perca conexão ou fique offline, você poderá perder apenas uma pequena fração de seu processamento, porém normalmente não há perda de dados ou processamento, pois os dados estão replicados e nesta situação o MapReduce automaticamente irá encontrar outro Node que tenha aquele dado e retomar o processamento.



MapReduce no Cluster



Apache Software Foundation



Ambari

The screenshot shows the Ambari Metrics dashboard. On the left, a sidebar lists services: HDFS, YARN, MapReduce2, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Falcon, Storm, Flume, Ambari Infra, Atlas, Kafka, Knox, Ranger, and Spark. The main area displays metrics in a grid:

Category	Metric	Value
HDFS	HDFS Disk Usage	47%
	DataNodes Live	1/1
YARN	HDFS Links	NameNode Secondary NameNode 1 DataNodes
	Memory Usage	No Data Available
MapReduce2	Network Usage	No Data Available
	CPU Usage	No Data Available
Tez	Cluster Load	No Data Available
	NameNode Heap	10%
Hive	NameNode RPC	1.68 ms
	NameNode CPU WIO	n/a
HBase	NameNode Uptime	1.8 hr
	HBase Master Heap	8%
Pig	HBase Links	HBase Master 1 RegionServers Master Web UI
	HBase Ave Load	7
Sqoop	HBase Master Uptime	30.6 min
	More...	
Oozie	More...	
ZooKeeper	More...	
Falcon	More...	
Storm	More...	
Flume	More...	
Ambari Infra	More...	
Atlas	More...	
Kafka	More...	
Knox	More...	
Ranger	More...	
Spark	More...	



Yet Another Resource Negotiator

- Introduzido no Hadoop 2
- Isola o problema de gerenciar recursos no cluster de MapReduce
- Habilita alternativas MapReduce (Spark, Tez) construídas no topo de YARN



TEZ

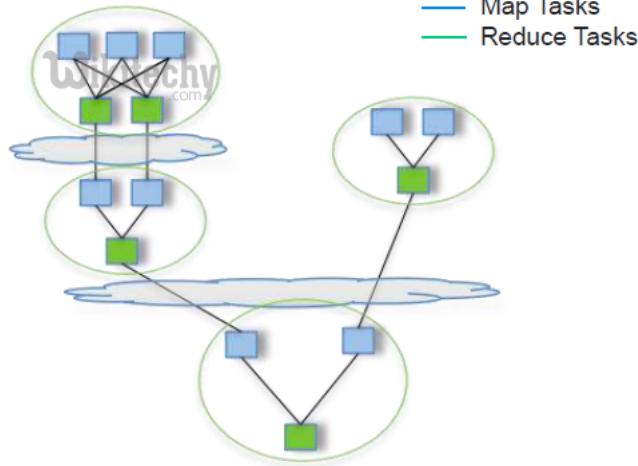
TEZ

- Directed Acyclic Graph Framework
- Isola o problema de gerenciar recursos no cluster de MapReduce
- Habilita alternativas MapReduce (Spark, Tez) construídas no topo do YARN
- Torna seus trabalhos Hive, Pig ou MapReduce mais rápidos!
- É um framework de aplicativo para a qual os clientes podem codificar como substitutos para MapReduce
- Constrói gráficos acíclicos direcionados (DAGs) para um processamento mais eficiente de trabalhos distribuídos
- Depende de uma visão mais holística do seu trabalho; elimina etapas desnecessárias e dependências.
- Otimiza o fluxo de dados físicos e o uso de recursos



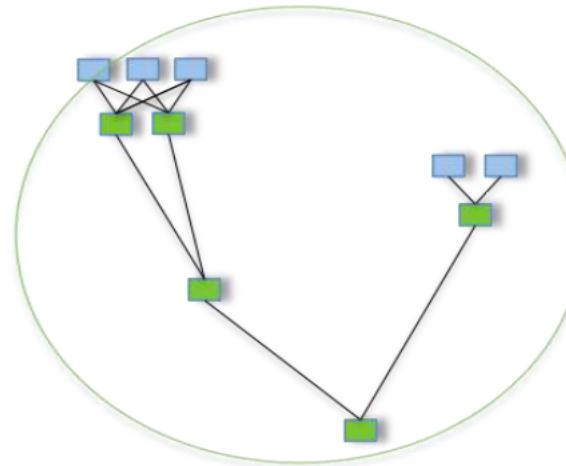
TEZ

MAPREDUCE



- Mapper and Reducer phases
- Shuffle between mapper and reducer tasks
- JobControl to run group of jobs with dependencies

TEZ



- Directed Acyclic Graph (DAG) with vertices
- Shuffle/One-One/Broadcast between vertex tasks
- Whole plan runs in a single DAG

HIVE

HIVE

- Traduz consultas SQL para trabalhos MapReduce ou Tez no seu cluster



HIVE

Por que usar?



- Usa sintaxe familiar de SQL (HiveQL)
- Interativo
- Escalável - funciona com “big data” em um grupo
 - Realmente mais apropriado para aplicações de armazém de dados
- consultas fáceis OLAP - maneira mais fácil do que escrevendo MapReduce em Java
- Altamente otimizado
- Altamente extensível
 - Funções definidas pelo usuário
 - Servidor Thrift
 - Driver JDBC / ODBC



Por que NÃO usar?

- Alta latência – não apropriada para OLTP
- Armazena dados desnormalizados
- SQL é limitada no que pode fazer – PIG, Spark permitem coisas mais complexas
- Nenhuma transação
- Sem atualizações, inserções, exclusões em nível de registro

HBASE

**Banco de dados escalável e não relacional
construído em HDFS**

- CRUD
- Não tem linguagem de query, só uma API de CRUD



Hbase Data Model

- Acesso rápido a qualquer ROW
- Uma linha é referenciada por uma única chave
- Cada ROW tem um pequeno número de COLUMN FAMILIES
- UMA FAMÍLIA DE COLUNA pode conter COLUNAS arbitrárias
- Você pode ter um número muito grande de COLUMNS em uma COLUMN FAMILY
- Cada CELL pode ter muitas VERSÕES com determinados timestamps
- Dados esparsos são A-OK - as colunas ausentes em uma linha não consomem armazenamento.



Algumas maneiras de acessar o HBase



- HBase shell
- Java API
- – Wrappers para Python, Scala, etc.
- Spark, Hive, Pig
- REST service
- Thrift service
- Avro service

Por que o Pig?

- Escrever mappes e reducers à mão leva muito tempo.
- Pig apresenta Pig Latin, um script linguagem que permite usar sintaxe SQL-like para definir seu mapa e reduzir passos.
- Altamente extensível com user-defined funções (UDFs)



Pig Exemplo

```
ratings = LOAD '/user/maria_dev/data/u.data' AS  
(userID:int, movieID:int, rating:int, ratingTime:int);
```

- Isso cria uma ***relation*** chamada "ratings" com um determinado esquema.

```
(660,229,2,891406212)  
(421,498,4,892241344)  
(495,1091,4,888637503)  
(806,421,4,882388897)  
(676,538,4,892685437)  
(721,262,3,877137285)
```



Pig Exemplo

Use PigStorage se você precisar de um delimitador diferente.

```
metadata = LOAD '/user/maria_dev/data/u.item' USING  
    PigStorage('|')AS (movieID:int, movieTitle:chararray,  
    releaseDate:chararray, videoRelease:chararray,  
    imdbLink:chararray);  
DUMP metadata;
```

```
(1,Toy Story (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)  
(2,GoldenEye (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?GoldenEye%20(1995)  
(3,Four Rooms (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Four%20Rooms  
%20(1995)  
(4,Get Shorty (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)  
(5,Copycat (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Copycat%20(1995))
```



Pig Exemplo

Criando uma *relation* de outra *relation*; FOREACH / GENERATE

```
metadata = LOAD '/user/maria_dev/ml-100k/u.item' USING PigStorage('|')
AS (movieID:int, movieTitle:chararray, releaseDate:chararray,
videoRelease:chararray, imdbLink:chararray);
```

```
nameLookup = FOREACH metadata GENERATE movieID, movieTitle,
ToUnixTime(ToDate(releaseDate, 'dd-MMM-yyyy')) AS releaseTime;
```

```
(1,Toy Story (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Toy%20Story%20(1995))
```



```
(1,Toy Story (1995),788918400)
```



Pig Exemplo

Group By

```
ratingsByMovie = GROUP ratings BY movieID;
```

```
DUMP ratingsByMovie;
```

```
(1,{(807,1,4,892528231),(554,1,3,876231938),(49,1,2,888068651), ... }  
(2,{(429,2,3,882387599),(551,2,2,892784780),(774,2,1,888557383), ... })
```



Pig Exemplo

```
ratingsByMovie = GROUP ratings BY movieID;
```

```
avgRatings = FOREACH ratingsByMovie GENERATE group AS movieID,  
AVG(ratings.rating) AS avgRating;
```

```
DUMP avgRatings;
```

```
(1,3.8783185840707963)  
(2,3.2061068702290076)  
(3,3.033333333333333)  
(4,3.550239234449761)  
(5,3.302325581395349)
```



Pig Exemplo

```
DESCRIBE ratings;  
DESCRIBE ratingsByMovie;  
DESCRIBE avgRatings;
```

```
ratings: {userID: int, movieID: int, rating: int, ratingTime: int}  
ratingsByMovie: {group: int, ratings: {(userID: int, movieID: int, rating: int, ratingTime: int)}}  
avgRatings: {movieID: int, avgRating: double}
```



Pig Exemplo

```
DESCRIBE ratings;  
DESCRIBE ratingsByMovie;  
DESCRIBE avgRatings;
```

```
ratings: {userID: int, movieID: int, rating: int, ratingTime: int}  
ratingsByMovie: {group: int, ratings: {(userID: int, movieID: int, rating: int, ratingTime: int)}}  
avgRatings: {movieID: int, avgRating: double}
```



Pig Exemplo

FILTER

```
fiveStarMovies = FILTER avgRatings BY avgRating > 4.0;
```

```
(12,4.385767790262173)  
(22,4.151515151515151)  
(23,4.1208791208791204)  
(45,4.05)
```



Pig Exemplo

JOIN

```
DESCRIBE fiveStarMovies;  
DESCRIBE nameLookup;  
fiveStarsWithData = JOIN fiveStarMovies BY movieID, nameLookup BY movieID;  
DESCRIBE fiveStarsWithData;  
DUMP fiveStarsWithData;
```

```
fiveStarMovies: {movieID: int, avgRating: double}  
nameLookup: {movieID: int, movieTitle: chararray, releaseTime: long}  
fiveStarsWithData: {fiveStarMovies::movieID: int, fiveStarMovies::avgRating: dou  
nameLookup::movieID: int, nameLookup::movieTitle:  
chararray, nameLookup::releaseTime: long}
```

```
(12, 4.385767790262173, 12, Usual Suspects, The (1995), 808358400)  
(22, 4.151515151515151, 22, Braveheart (1995), 824428800)  
(23, 4.1208791208791204, 23, Taxi Driver (1976), 824428800)
```



Pig Exemplo

ORDER BY

```
oldestFiveStarMovies = ORDER fiveStarsWithData BY  
nameLookup::releaseTime;
```

```
DUMP oldestFiveStarMovies;
```

```
(493,4.15,493,Thin Man, The (1934),-1136073600)  
(604,4.012345679012346,604,It Happened One Night (1934),-1136073600  
(615,4.0508474576271185,615,39 Steps, The (1935),-1104537600)  
(1203,4.0476190476190474,1203,Top Hat (1935),-1104537600)
```



Pig



Hey, ho, let's go!



Pig Exemplo

ORDER BY

```
oldestFiveStarMovies = ORDER fiveStarsWithData BY  
nameLookup::releaseTime;
```

```
DUMP oldestFiveStarMovies;
```

```
(493,4.15,493,Thin Man, The (1934),-1136073600)  
(604,4.012345679012346,604,It Happened One Night (1934),-1136073600  
(615,4.0508474576271185,615,39 Steps, The (1935),-1104537600)  
(1203,4.0476190476190474,1203,Top Hat (1935),-1104537600)
```





INTEGRAÇÃO MYSQL & HADOOP

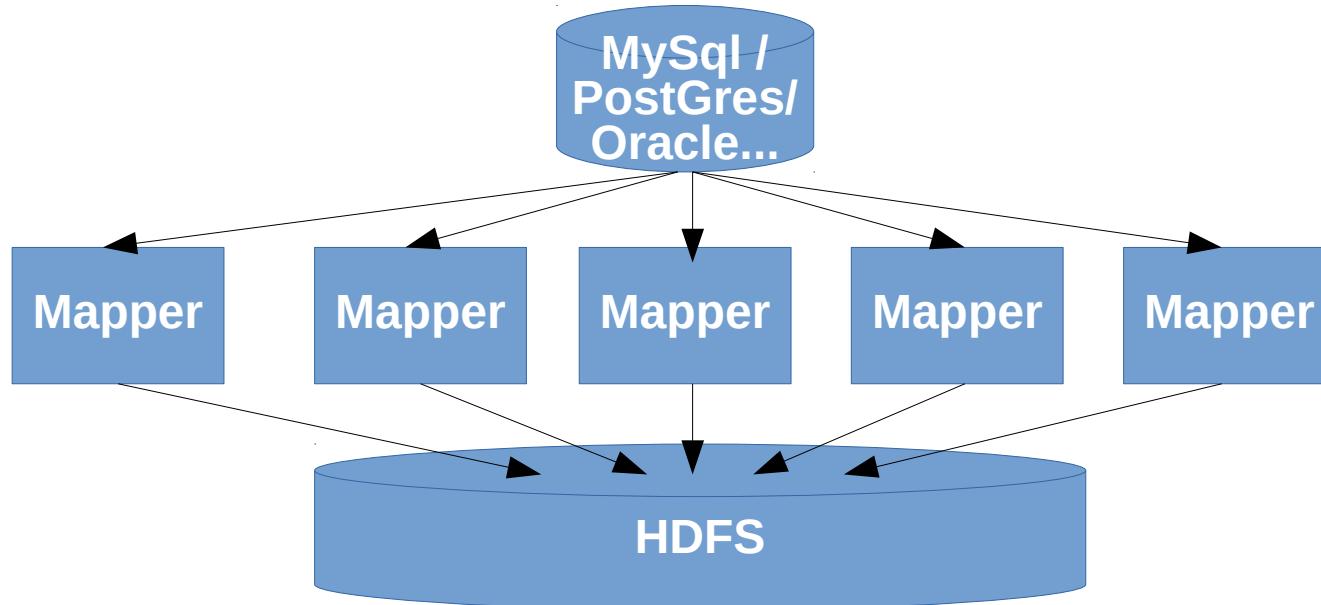
O que é o MySQL?

- Banco de dados relacional popular e gratuito
- Geralmente monolítico por natureza
- Mas, pode ser usado para OLTP - então exportar dados para o MySQL pode ser útil
- Dados existentes podem existir no MySQL que você deseja importar para o Hadoop

Sqoop

Sqoop pode manipular BIGDATA

- Na verdade, inicia trabalhos do MapReduce para manipular a importação ou exportação de seus dados!



Sqoop

Importar dados do MySQL para HDFS

```
sqoop import --connect jdbc:mysql://localhost/movielens --driver com.mysql.jdbc.Driver --table movies -m 1
```

Importar dados do MySQL diretamente no Hive

```
sqoop import --connect jdbc:mysql://localhost/movielens --driver com.mysql.jdbc.Driver --table movies --hive-import -m 1
```



Sqoop

Importações incrementais

- Você pode manter seu banco de dados relacional e o Hadoop em sincronia
- --check-column e -last-value



Sqoop

Sqoop: Exportar dados do Hive para MySQL

- `sqoop export --connect jdbc:mysql://localhost/movielens -m 1 --driver com.mysql.jdbc.Driver --table exported_movies --export-dir /apps/hive/warehouse/movies - --input-fields-terminated-by '\0001'`
- A tabela de destino já deve existir no MySQL, com colunas na ordem esperada



Sqoop

Vamos praticar com o MySQL e o Sqoop

- Importe dados do MovieLens para um banco de dados MySQL
- Importe os filmes para o HDFS
- Importe os filmes para o Hive
- Exportar os filmes de volta para o MySQL



Zookeeper

O que é o ZooKeeper?

- Basicamente, ele controla as informações que devem ser sincronizadas em todo seu cluster
 - Qual nó é o mestre?
 - Quais tarefas são atribuídas a quais workers?
 - Quais workers estão atualmente disponíveis?
- É uma ferramenta que os aplicativos podem usar para recuperar falhas parciais no cluster.
- Uma parte integrante do HBase, High-Availability (HA) MapReduce, Drill, Storm, Solr, e muito mais



Zookeeper

Modos de falha

- Master falha, precisa fazer failover para um backup
- Worker falha - seu trabalho precisa ser redistribuído
- Problemas de rede - parte do seu cluster não consegue ver o restante



Zookeeper

Operações “primitivas” em um sistema distribuído

- Eleição mestre

- Um nó se registra como um mestre e mantém um "bloqueio" nesses dados
- Outros nós não podem se tornar mestre até que esse bloqueio seja liberado
- Apenas um nó permitido manter o bloqueio de cada vez

- Detecção de falhas

- Os dados "efêmeros" sobre a disponibilidade de um nó desaparecem automaticamente se o nó desconecta ou falha ao atualizar após um período de tempo limite.

- Gerenciamento de grupo

- Metadados

- Lista de tarefas pendentes, atribuições de tarefas



Zookeeper

API do ZooKeeper:

```
- Criar, excluir, existir, setData, getData, getChildren  
/  
|  
|-----/master "Master1.foobar.com:2223"  
|  
|-----/workers  
|  
|----- worker-1 "worker-2.foobar.com:2225"  
|----- worker-2 "worker-5.foobar.com:2225"
```



Zookeeper

Notificações

- Um cliente pode se registrar para notificações em um znode
 - Evita a pesquisa contínua
 - Exemplo: registrar para notificação no /master - se ele parar, tente assumir como o novo mestre.



Zookeeper

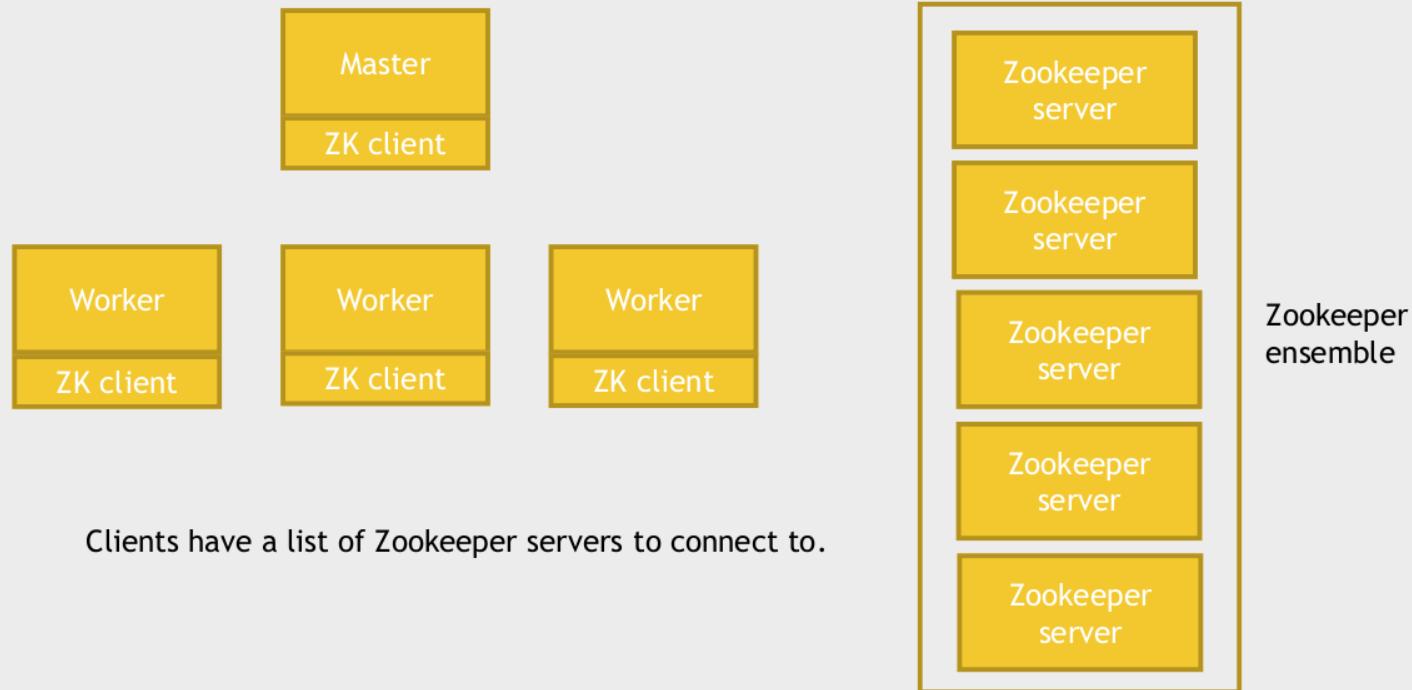
Znodes persistentes e efêmeros

- Os znodes persistentes permanecem armazenados até serem excluídos explicitamente
 - isto é, a atribuição de tarefas aos trabalhadores deve persistir mesmo se o mestre travar
- Os znodes efêmeros desaparecem se o cliente que o criou falhar ou perder conexão ao ZooKeeper
 - ou seja, se o mestre travar, ele deve liberar seu bloqueio no znode que indica qual nó é o mestre!



Zookeeper

ZooKeeper Architecture



Storm

Processamento de fluxo em tempo real

O que é o Apache Storm?

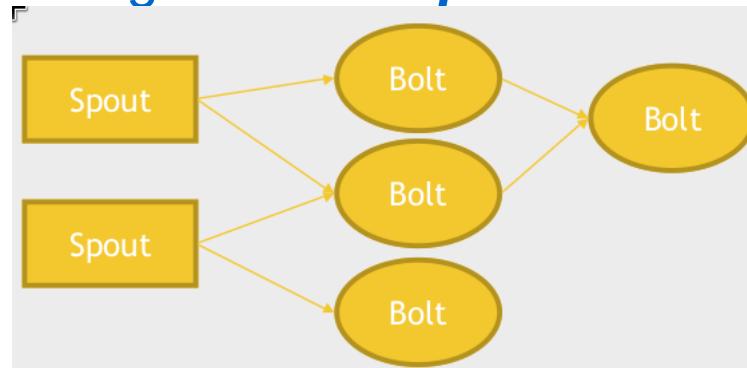
- Framework para processar fluxos contínuos de dados em um cluster
 - Pode rodar em cima do YARN (como Spark)
- Funciona em eventos individuais, não em micro lotes (como o Spark Streaming)
 - Se você precisa de latência de sub-segundo, Storm é a opção



Storm

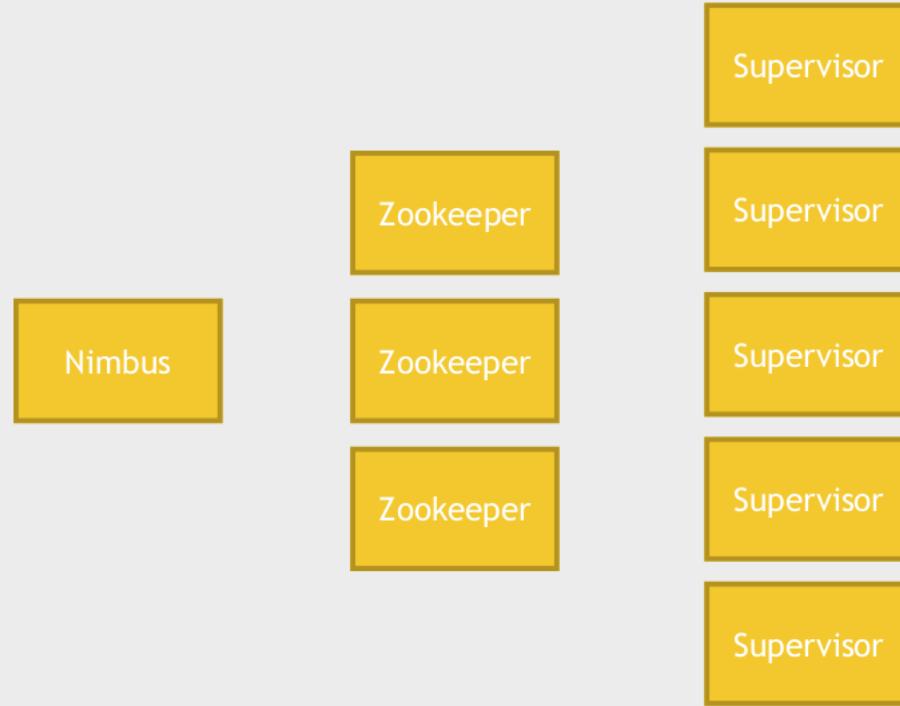
Terminologia do Storm

- Um ***stream*** consiste em tuplas que fluem através de...
- ***Spouts*** que são fontes de dados de fluxo (Kafka, Twitter, etc.)
- ***Bolts*** que processam dados de fluxo conforme são recebidos
 - Transformar, agregar, gravar em bancos de dados / HDFS
- Uma ***topology*** é um gráfico de ***Spouts*** e ***Bolts*** que processam seu fluxo



Storm

Storm architecture



Storm

Desenvolvendo aplicativos Storm

- Geralmente feito com Java

- Embora os bolts possam ser direcionados através de scripts em outros idiomas

Núcleo do Storm

- A API de nível inferior para o Storm

- Semântica "pelo menos uma vez"

- Trident

- API de nível superior para tempestade

- semântica “Exactly once”

- O Storm executa seus aplicativos “para sempre” depois de enviado - até você explicitamente pare



Storm

Storm vs. Spark Streaming

- Apesar do Spark ter mais ferramentas... se você precisar de processamento em tempo real (sub-segundo) de eventos, em, Storm é sua escolha
- Kafka + Storm é uma combinação bastante popular



Flume

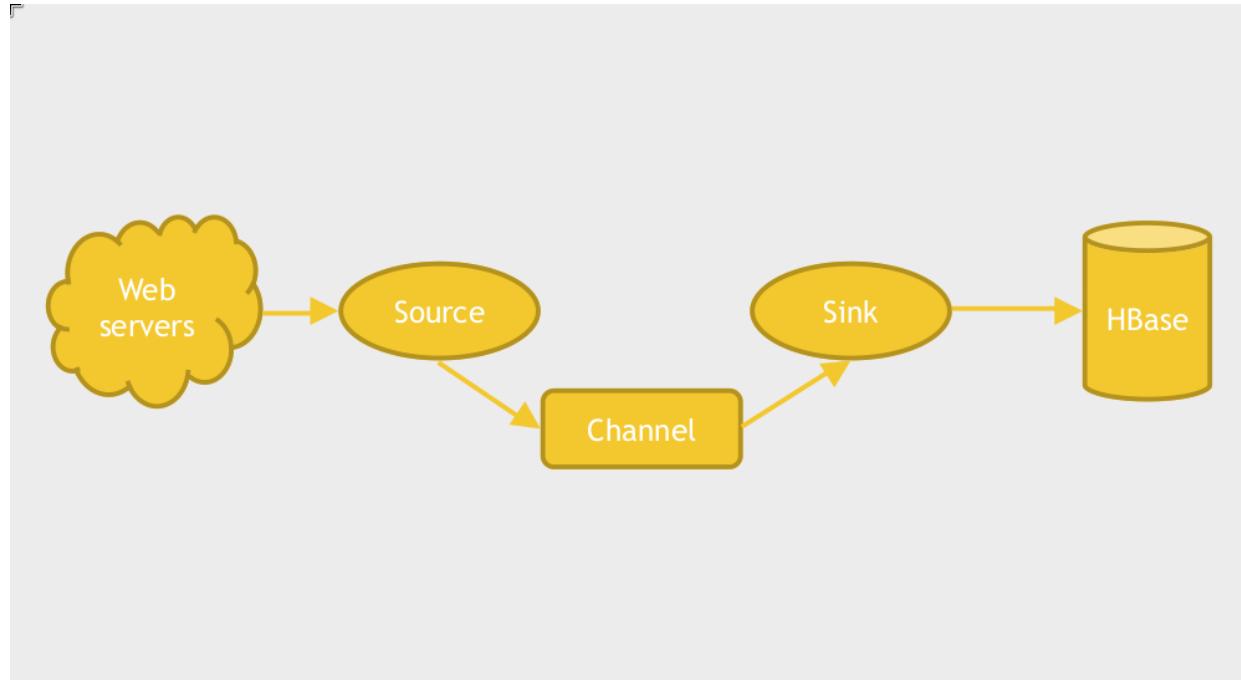
O que é o Flume?

- Outra maneira de transmitir dados em seu cluster
- Feito desde o início com o Hadoop em mente
 - Dissipadores embutidos para HDFS e Hbase
- Originalmente feito para manipular a agregação de logs



Flume

Anatomia de um Flume Agent and Flow



Flume

Componentes de um agente

- Fonte

- De onde vêm os dados
- Opcionalmente, pode ter Seletores de Canal e Interceptores

- Canal

- Como os dados são transferidos (via memória ou arquivos)

- Pia (Sink)

- Onde os dados estão indo
- Pode ser organizado em grupos de pia
- Uma pia pode se conectar a apenas um canal



Flume

Componentes de um agente

- Fonte

- De onde vêm os dados
- Opcionalmente, pode ter Seletores de Canal e Interceptores

- Canal

- Como os dados são transferidos (via memória ou arquivos)

- Pia (Sink)

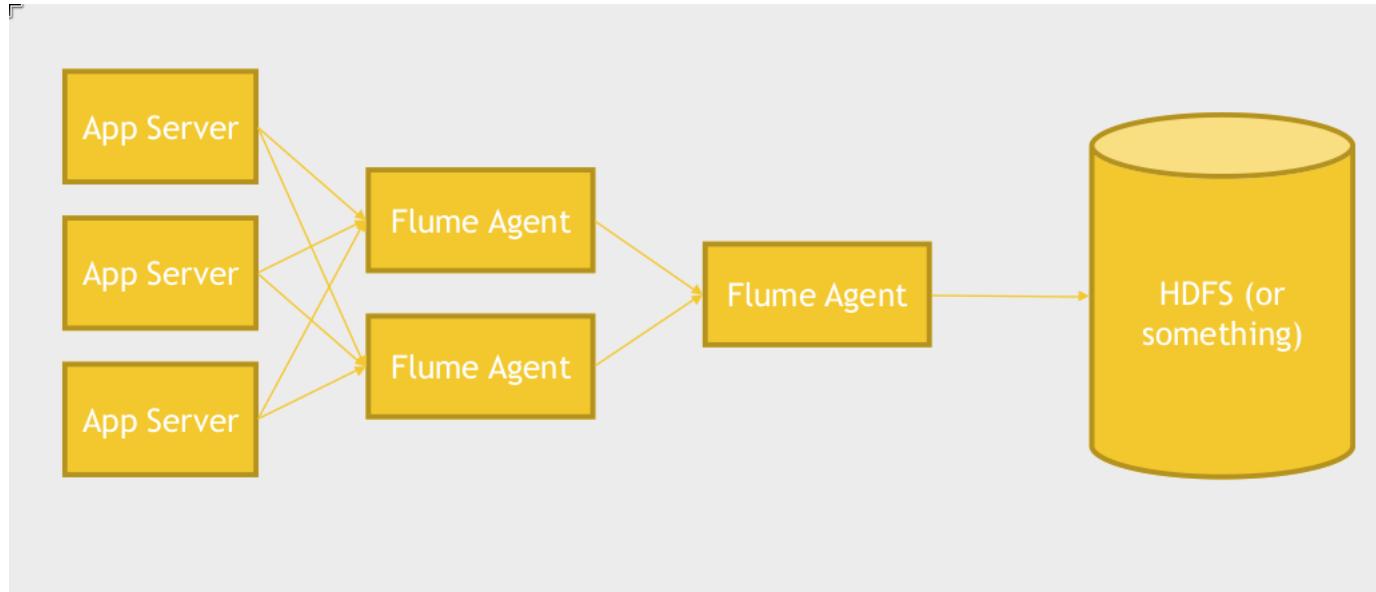
- Onde os dados estão indo
- Pode ser organizado em grupos de pia
- Uma pia pode se conectar a apenas um canal

O canal é notificado para excluir uma mensagem quando o coletor a processa.



Flume

Usando o Avro, os agentes podem se conectar outros agentes também



Flume

Tipos de Sink Built-in

- HDFS
- Hive
- HBase
- Avro
- Thrift
- Elasticsearch
- Kafka
- Personalizado
- E mais!



Kafka

STREAMING COM KAFKA

Publicar / Assinar Mensagens com Kafka

O que é streaming?

- Conversamos sobre o processamento de big data histórico e existente
 - residente no HDFS
 - residente em um banco de dados
- Mas como os novos dados chegam ao seu cluster? Especialmente se for "Big Data"?
 - Novas entradas de log dos seus servidores da web
 - Novos dados do sensor do seu sistema IoT
 - Novas negociações de ações
- Streaming permite publicar esses dados, em tempo real, em seu cluster.
 - E você pode até processá-lo em tempo real quando ele chegar!



Dois problemas

- Como obter dados de várias fontes diferentes fluindo para o cluster
- Processaestes dados quando chegarem
- Primeiro, vamos nos concentrar no primeiro problema



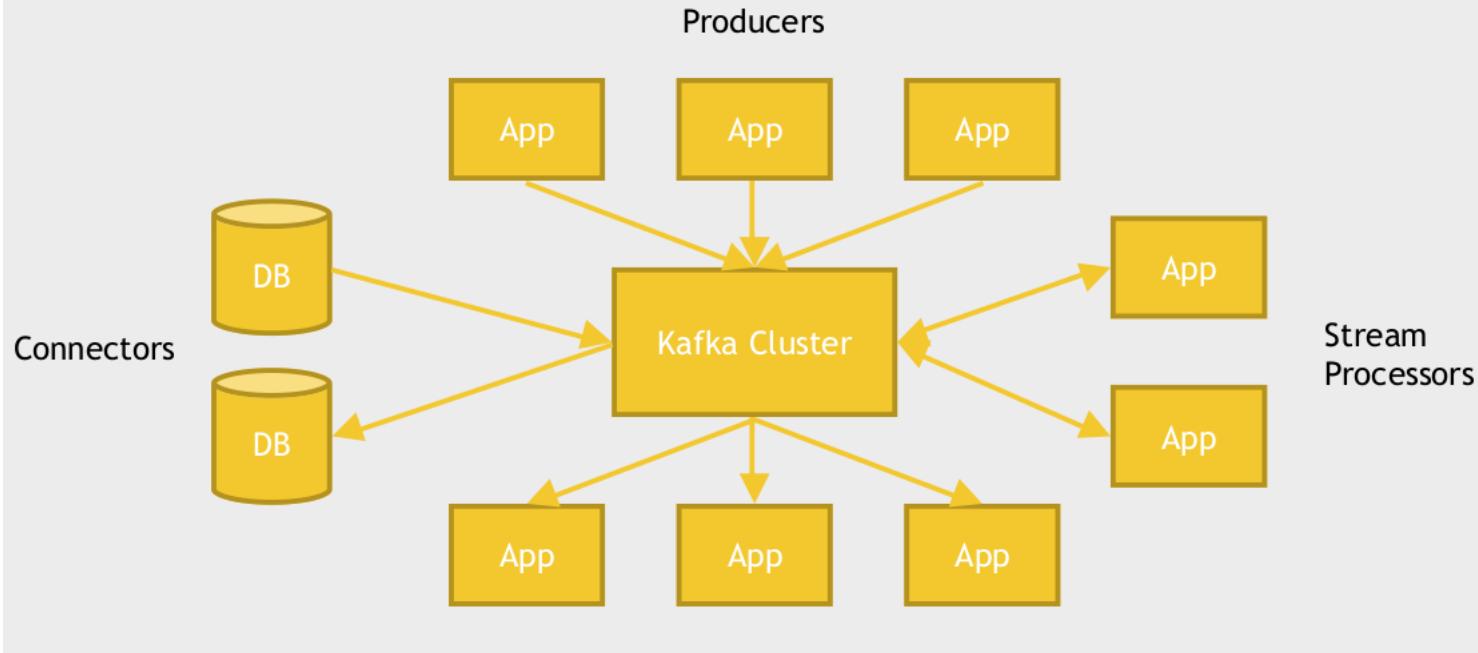
Kafka

- Kafka é um sistema de publicação / assinatura de mensagens de propósito geral
- Os servidores Kafka armazenam todas as mensagens recebidas dos ***publishers*** por um período de tempo, e publica-os em um fluxo de dados (***stream***) chamado ***topic***.
- Os ***consumers*** se inscrevem em um ou mais ***topics*** e recebem dados a medida que são publicados
- Um stream / topic pode ter muitos ***consumers*** diferentes, todos com suas próprios
posição no fluxo mantido
- Não é apenas para o Hadoop



Kafka

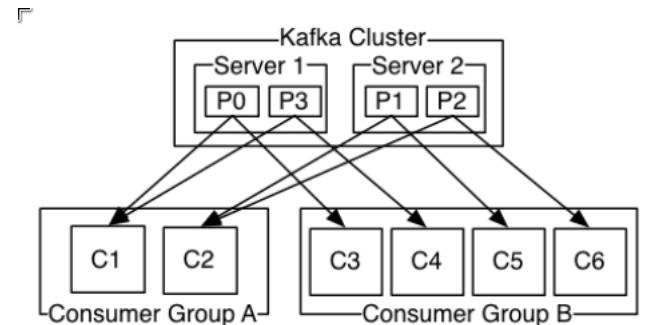
Kafka architecture



Kafka

Como o Kafka escala:

- O próprio Kafka pode ser distribuído entre muitos processos em muitos servidores
 - Distribuirá o armazenamento do fluxo dados bem
- Os consumidores também podem ser distribuídos
 - Consumidores do mesmo grupo tem mensagens distribuídas entre eles
 - Consumidores de diferentes grupos receberão sua própria cópia de cada mensagem

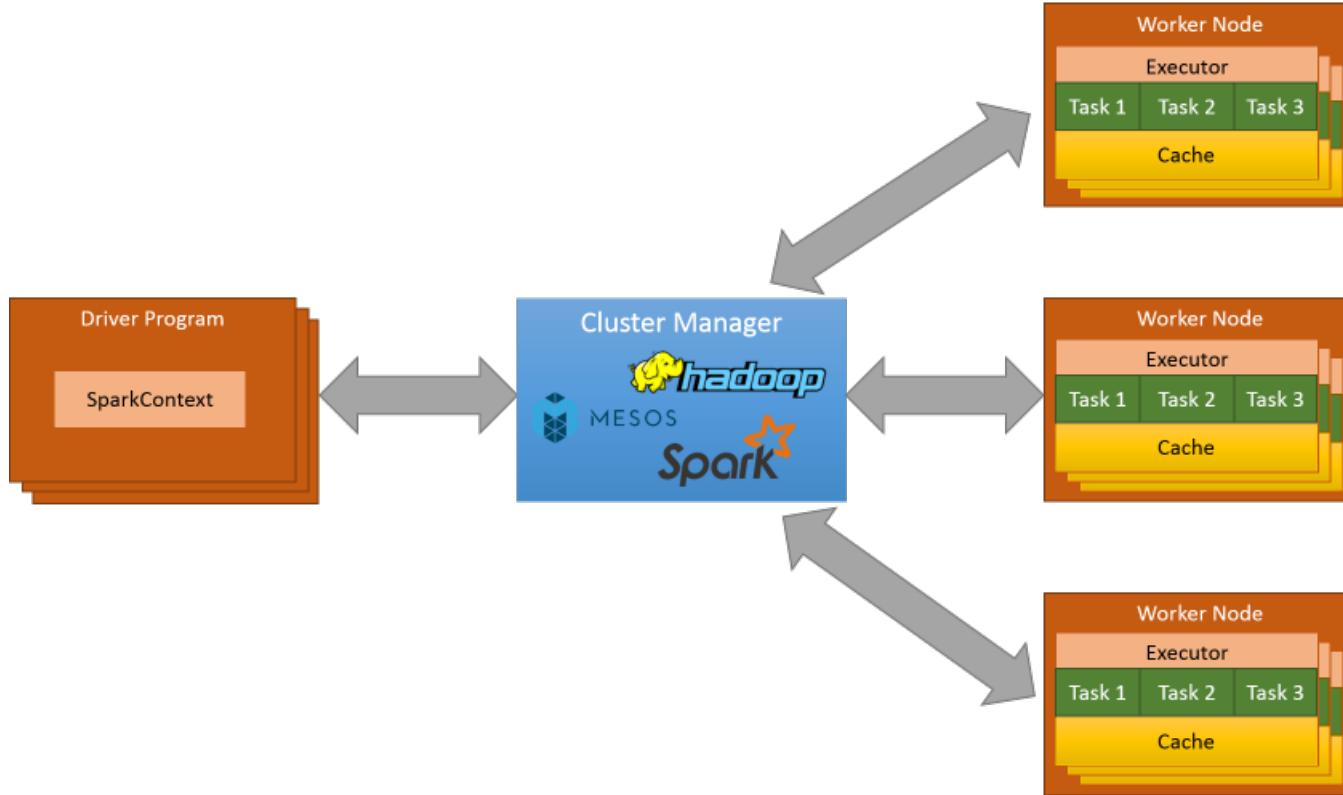


O que é o Spark?



- "Um mecanismo rápido e geral para processamento de dados em larga escala"

Escalável



Rápido

- "Executa programas até 100x mais rápido que o Hadoop MapReduce na memória ou 10x mais rápido no disco. "
- O mecanismo DAG (gráfico acíclico direcionado) otimiza os fluxos de trabalho



Preferido por muitos

- Amazon
- Ebay: log analysis and aggregation
- NASA JPL: Deep Space Network
- Groupon
- TripAdvisor
- Yahoo
- Outros:

[https://cwiki.apache.org/confluence/display/SPARK/
Powered+By+Spark](https://cwiki.apache.org/confluence/display/SPARK/Powered+By+Spark)



Não é tão complicado...

- Código em Python, Java ou Scala
- Construído em torno de um conceito principal: o Conjunto de dados distribuído resiliente (RDD)



Componentes

Spark
SQL

Spark
Streaming

MLlib
(machine
learning)

GraphX
(graph)

Apache Spark



Vamos usar Python

■ Por que Python?

- É muito mais simples e isso é apenas uma visão geral.
- Não precisa compilar nada, lidar com JARs, dependências, etc.

■ Mas ...

- O próprio Spark está escrito em Scala
- O modelo de programação funcional do Scala é adequado para distribuir em processamento
- Oferece desempenho rápido (**Scala compila para bytecode Java**)



Sem medo...

- O código Scala no Spark parece muito com o código Python.
 - Código Python para números quadrados em um conjunto de dados:

```
nums = sc.parallelize ([1, 2, 3, 4])
quadrado = nums.map (lambda x: x *
x) .collect ()
```

- Código Scala para números quadrados em um conjunto de dados:

```
val nums = sc.parallelize (List (1, 2, 3, 4))
val squared = nums.map (x => x * x) .collect ()
```



O que é o RDD?

- Resilient
- Distributed
- Dataset



O Contexto Spark (SparkContext)

- Criado pelo seu programa de driver
- É responsável por tornar resiliente e distribuído o RDD!
- Cria RDD's
- O shell Spark cria um objeto "sc" para você



Criando RDD's

- `nums = parallelize ([1, 2, 3, 4])`
- `sc.textFile ("file: //c: /users/frank/gobs-o-text.txt")`
 - ou `s3n: //`, `hdfs: //`
- `hiveCtx = HiveContext (sc)` `rows = hiveCtx.sql ("SELECT nome, idade FROM usuários")`
- também pode criar a partir de:
 - JDBC
 - Cassandra
 - HBase
 - Elastisearch
 - JSON, CSV, arquivos de sequência, arquivos de objetos, vários formatos compactados



Transformando RDD's

- map
- flatmap
- filter
- distinct
- sample
- union, intersection, subtract, cartesian



Exemplo map

- `rdd = sc.parallelize([1, 2, 3, 4])`
- `squaredRDD = rdd.map(lambda x: x*x)`
- Resultando em 1, 4, 9, 16



O que é essa tal de lambda???

Muitos métodos RDD aceitam uma função como um parâmetro

```
rdd.map (lambda x: x * x)
```

É a mesma coisa que

```
def squareIt (x):
```

```
    return x * x
```

```
rdd.map (squareIt)
```

Assim, você agora entende de programação funcional!!!!



Ações RDD

- collect
- count
- countByValue
- take
- top
- reduce
- ... e outras ...



Lazy evaluation

- Nada realmente acontece no seu programa de driver até que uma ação seja chamada!



Configuração



Checklist

Instalar o Enthought Canopy (versão $\geq 1.6.2!$)

Abra uma janela de edição e vá para o prompt de comando interativo:

```
!pip install pydot2
```

Abra o package manager, e instale:
scikit_learn, numpy, pandas, stastmodels,
xlrd, pydotplus



Python Basics

paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



Vamos ver algum código.



Tipos de Dados



Muitos sabores de dados



paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



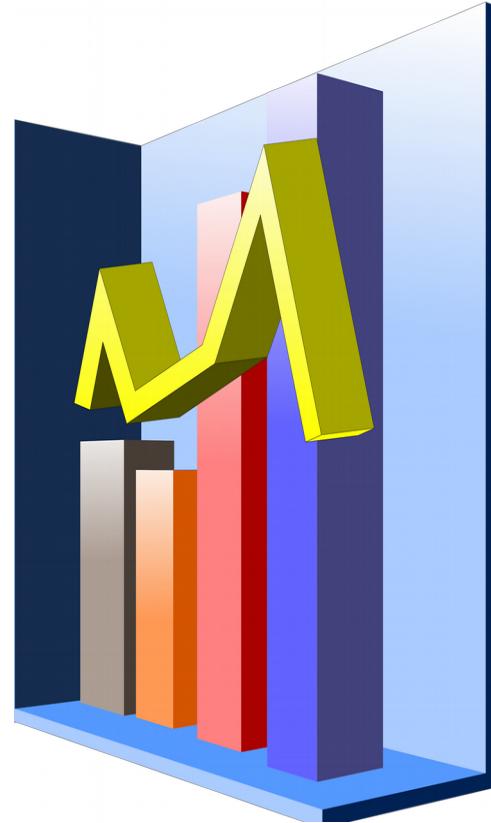
Principais Tipos de Dados

- Numéricos
- Nominais (ou Categóricos)
- Ordinais



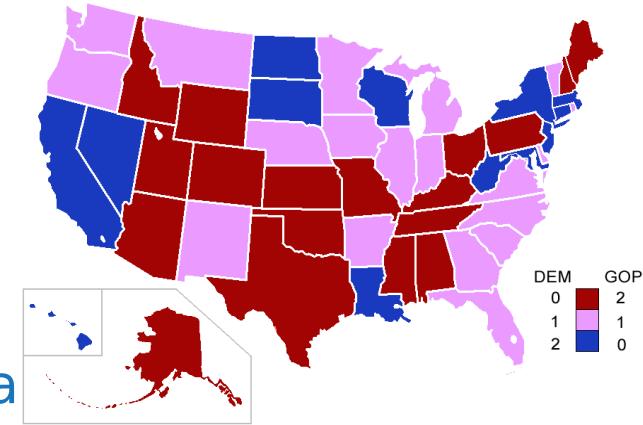
Numéricos

- Representam algum tipo de medida quantitativa
 - Altura da população, tempo de carga de páginas
Preço de ações, etc...
- Dados Discretos
 - Baseado em inteiros; usualmente contam algo.
 - Quantas compras um cliente faz ao ano?
 - Quantas vezes eu virei a cabeça?
- Dados Contínuos
 - Tem um número infinito de valores possíveis
 - Quanto tempo um usuário gasta no check out?
 - Quanta chuva cai em um determinado dia?



Nominais

- Dados qualitativos que não têm significado matemático inerente
 - Sexo, Sim/Não (dados binários), Raça, Estado De Residência, Categoria de Produto, Partido, etc.
- Você pode assinalar números para as categorias. Para representá-las mais compactadamente, mas os números não tem significado matemático.



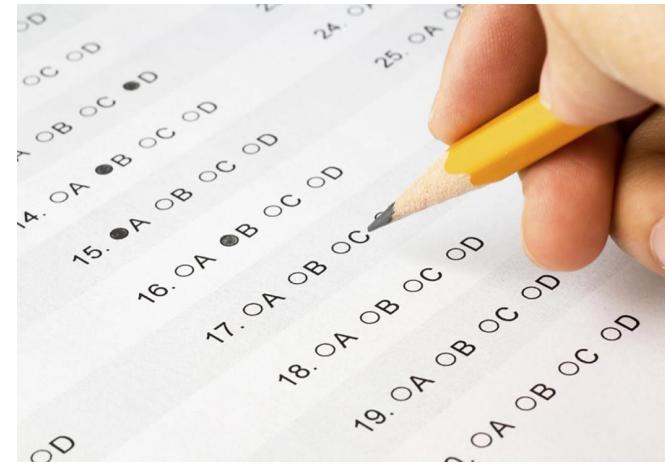
Ordinais

- Mistura de numéricos e nominais
- Dados Nominais não tem significado matemático
- Exemplo: escala de 1-5 para ratings.
 - Ratings devem ser 1, 2, 3, 4, ou 5
 - Mas este valores tem significado matemático; 1 significa um filme pior do que um 2.



Quiz time!

- Estes tipos de dados são numéricos, ordinais ou nominais?
 - Quanta gasolina tem no tanque do seu carro?
 - Um rating de sua saúde geral, onde as opções são 1, 2, 3, ou 4, correspondendo a “ruim”, “moderada”, “boa”, e “excelente”
 - As raças de seus colegas de classe
 - Idade em anos
 - Dinheiro gasto em uma loja



Média, mediana e moda



Média

- Soma / número de amostras
- Exemplo:
 - Número de crianças em cada casa da minha rua:

0, 2, 3, 2, 1, 0, 0, 2, 0

A **MÉDIA** é $(0+2+3+2+1+0+0+2+0) / 9 = 1.11$



Mediana

- Ordene os valores e pegue o do meio da lista.
- Exemplo:

0, 2, 3, 2, 1, 0, 0, 2, 0

Ordene:

0, 0, 0, 0, 1, 2, 2, 2, 3



Mediana

- Se você tiver um número par de amostras, Tire a média dos dois do meio.
- Mediana é menos suscetível outliers do que the mean
 - Exemplo: renda familiar média nos EUA é \$72,641, mas a mediana é apenas \$51,939 - Porque a média é distorcida por um punhado De bilionários.
 - Mediana representa melhor o Americano “típico” Nesse exemplo.



Moda

- O valor mais comum em um dataset
 - Não relevante para dados numéricos contínuos
- De volta ao exemplo do número de crianças:

0, 2, 3, 2, 1, 0, 0, 2, 0

Quantos de cada valor temos?

0: 4, 1: 1, 2: 3, 3: 1

A MODA é 0

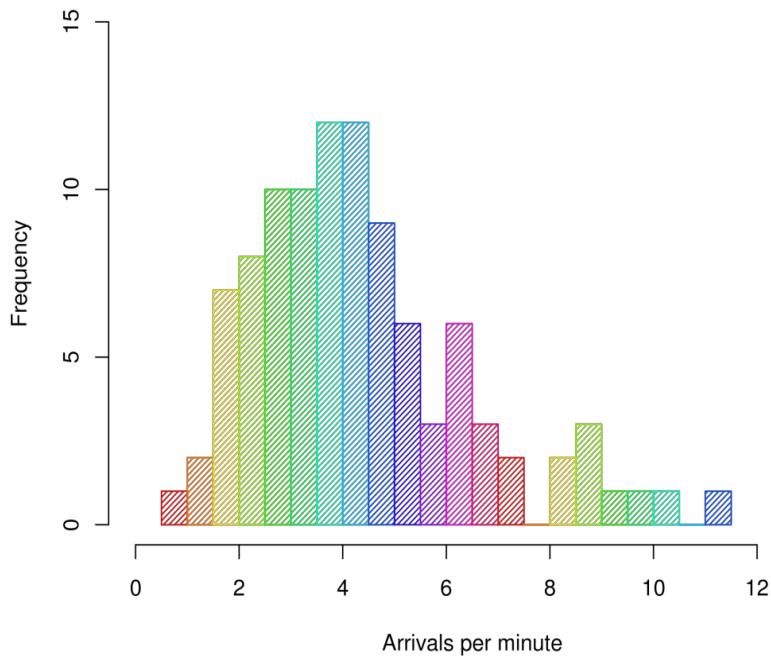


Desvio Padrão e Variância



Exemplo de um Histograma

Histogram of arrivals



Variância mede quanto “espalhados” os dados são.

- Variância (σ^2) é simplesmente a **média das diferenças quadradas da média**
- Exemplo: Qual a variância deste dataset (1, 4, 5, 4, 8)?
 - Calcule a Média: $(1+4+5+4+8)/5 = 4.4$
 - Agora encontre as diferenças da média: (-3.4, -0.4, 0.6, -0.4, 3.6)
 - Encontre o quadrado das diferenças: (11.56, 0.16, 0.36, 0.16, 12.96)
 - Calcule a média do quadrado das diferenças:

$$\sigma^2 = (11.56 + 0.16 + 0.36 + 0.16 + 12.96) / 5 = 5.04$$

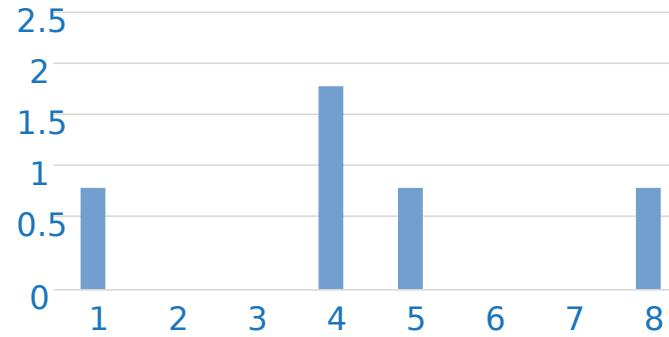


Desvio Padrão é a raiz quadrada da Variância

$$\sigma^2 = 5.04$$

$$\sigma = \sqrt{5.04} = 2.24$$

Então o Desvio Padrão de (1, 4, 5, 4, 8) é 2.24.



Isso é normalmente usado para identificar outliers. Pontos que ficam a mais de um Desvio Padrão da Média podem ser considerados não usuais.

Você pode se referir a quanto extremo é um ponto de dados dizendo “quantos sigmas” longe da média ele está.

População vs. Amostra

- Se você está trabalhando com uma Amostra dos dados ao invés de Um dataset completo de dados (a *População* inteira)...
 - Então você vai querer usar a “variância da amostra” ao invés da “variância da população”
 - Para N amostras, você divide a variância quadrada por N-1 ao invés de N.
 - Então, no nosso exemplo, calculamos a variância da população assim:
 - $\sigma^2 = (11.56 + 0.16 + 0.36 + 0.16 + 12.96) / 5 = 5.04$
 - But the sample variance would be:
 - $S^2 = (11.56 + 0.16 + 0.36 + 0.16 + 12.96) / 4 = 6.3$



Fórmulas

- Variância da População:

$$\sigma^2 = \frac{\sum(X-\mu)^2}{N}$$

- Variância da Amostra:

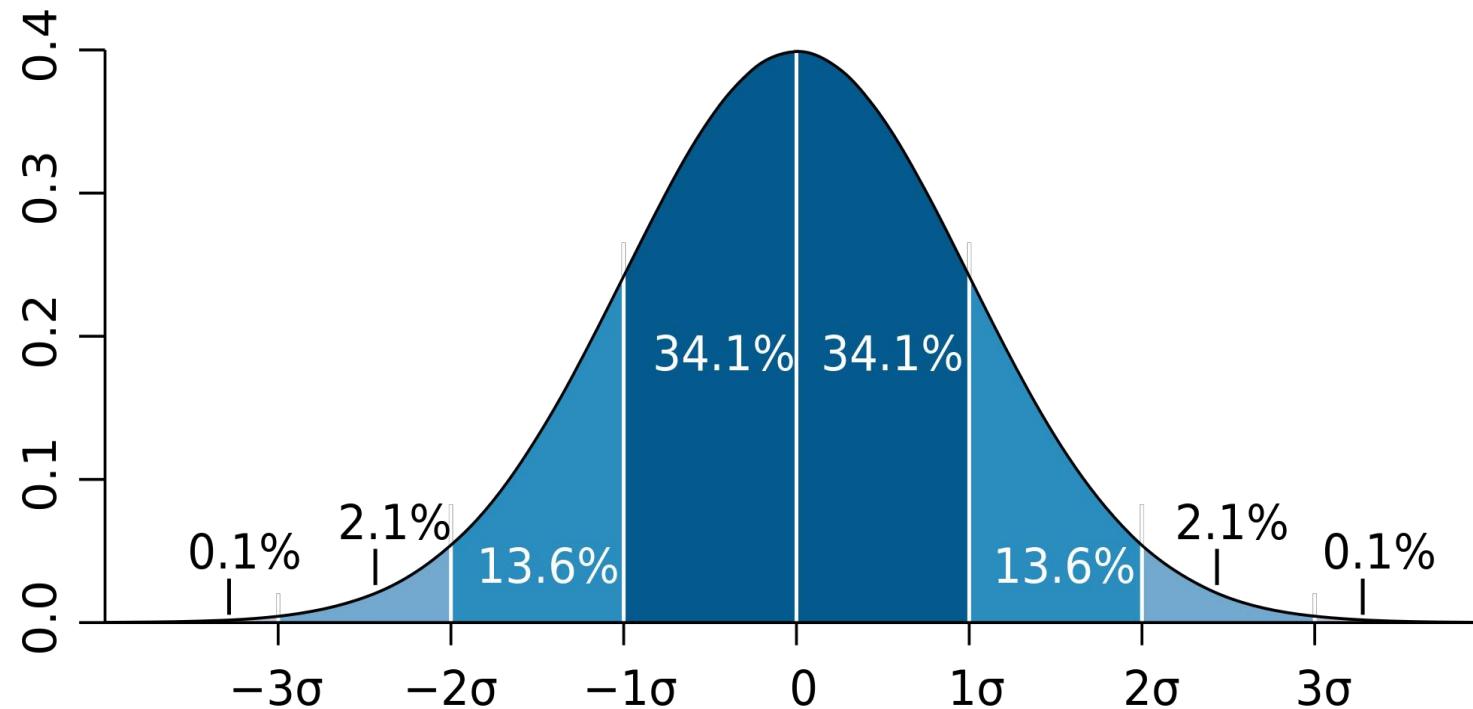
$$s^2 = \frac{\sum(X-M)^2}{N-1}$$



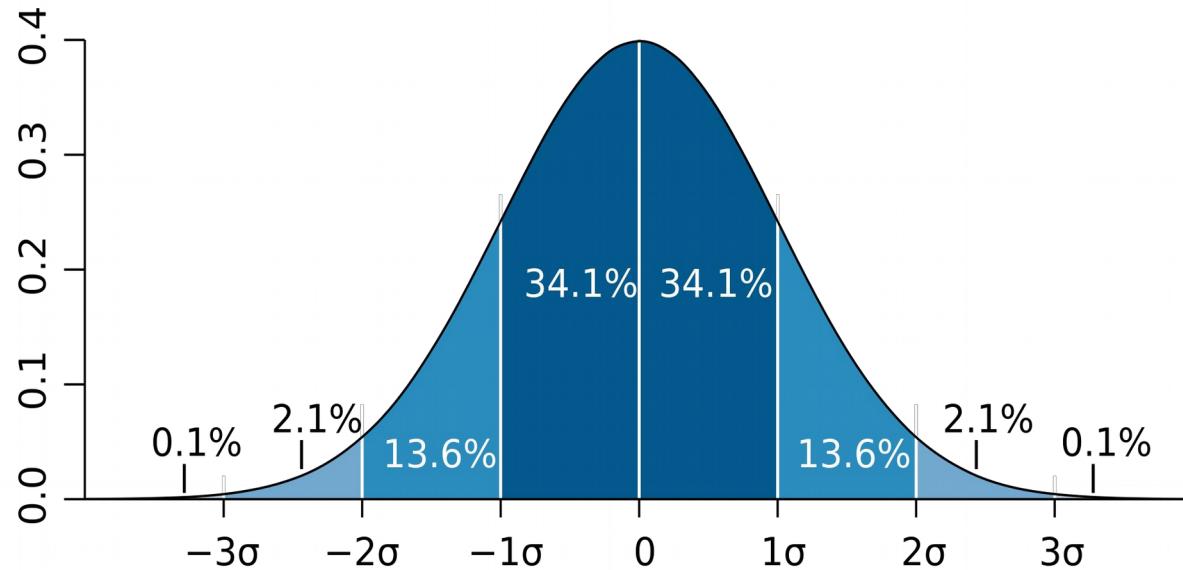
Funções de densidade de probabilidade



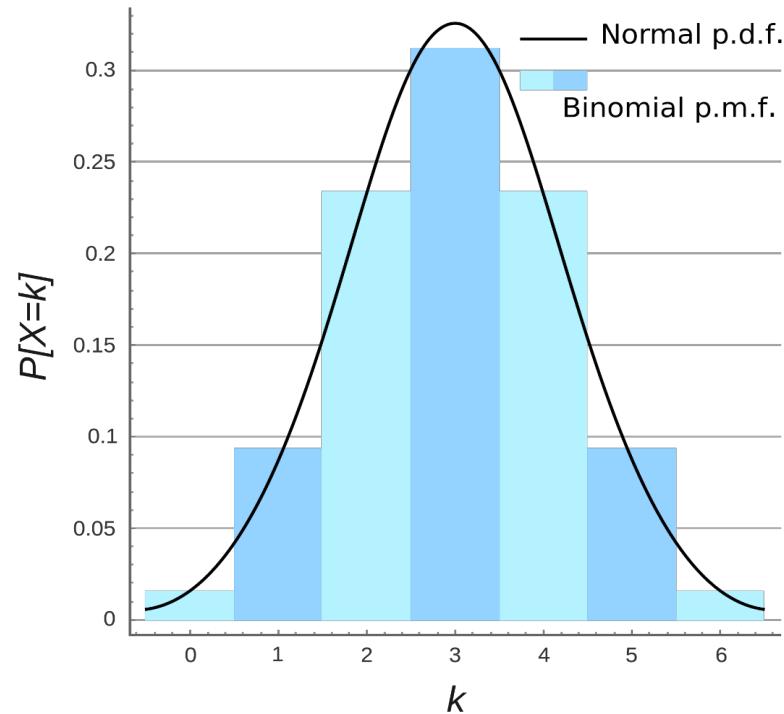
Exemplo: uma “distribuição normal”



Dá a probabilidade de um ponto de dados cair dentro de um dado intervalo de um dado valor.



Função de massa de probabilidade



Vamos ver alguns exemplos



Percentis e Momentos

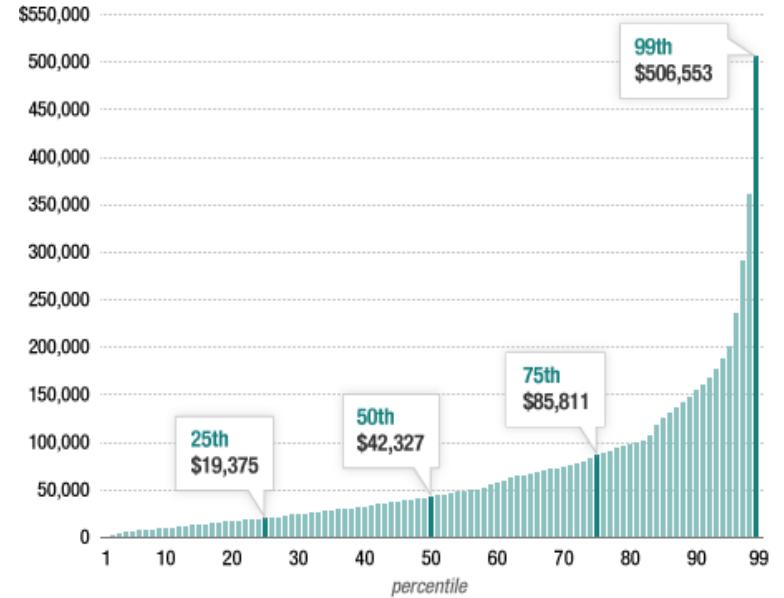
paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark

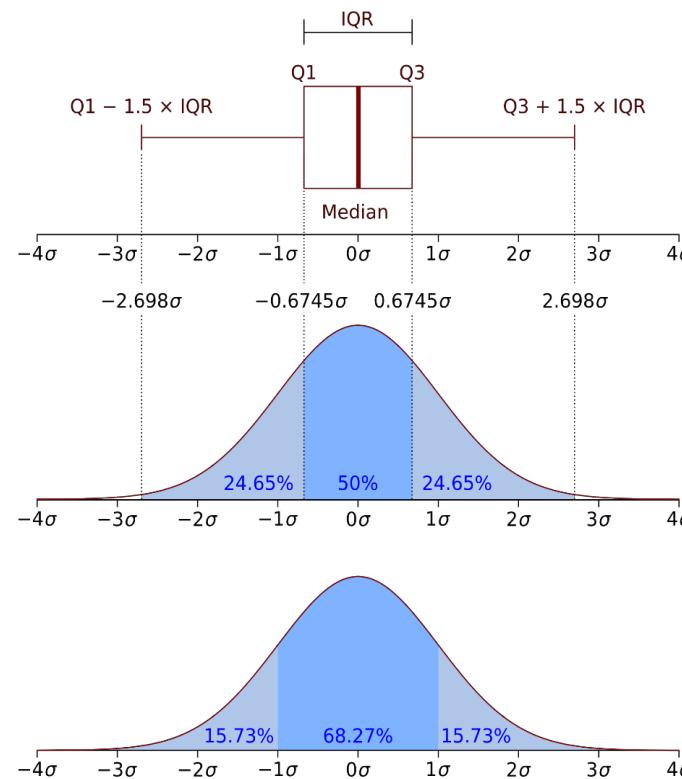


Percentis

- Em um conjunto de dados, qual é o ponto em que X% dos valores são menores que esse valor?
- Exemplo: distribuição de renda



Percentis em uma distribuição normal



Vamos ver alguns exemplos



Momentos

Medidas quantitativas da forma de uma função de densidade de probabilidade Matematicamente elas são um pouco difíceis de entender:

$$\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx \quad (\text{para um momento } n \text{ em torno do valor } c).$$

Mas intuitivamente, é muito mais simples em estatística.



O primeiro momento é a média



O segundo momento é a variância

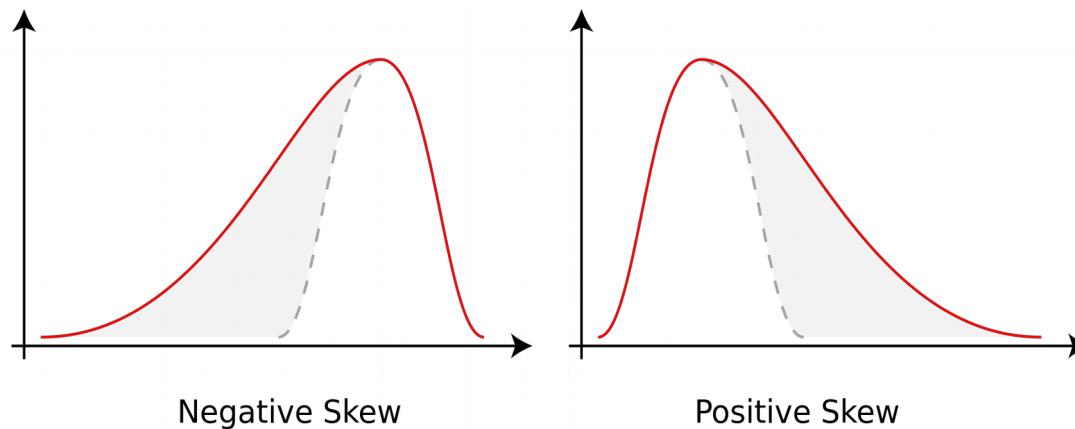


Simples assim...



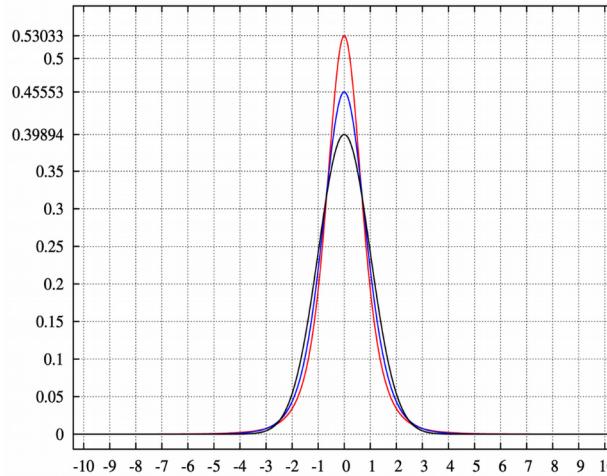
O terceiro momento “inclinação”

Quão “desequilibrada” é a distribuição? Uma distribuição com uma cauda mais longa à esquerda ficará inclinada para a esquerda e terá uma inclinação negativa.



O quarto momento é "curtose"

Quão espessa é a cauda e quão nítido é o pico, comparado a uma distribuição normal? Exemplo: picos mais altos têm maior curtose



Vamos computar os 4 momentos com Python

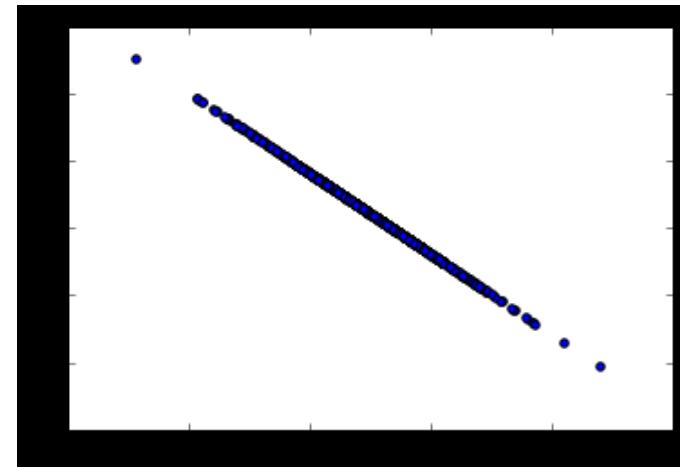
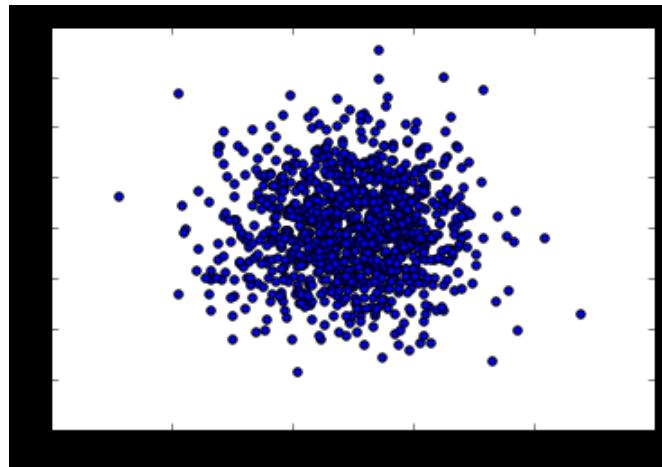


Covariância e Correlação



Covariância

Mede como duas variáveis variam em conjunto a partir de suas médias.



Medindo a Covariância

- Pense nos conjuntos de dados para as duas variáveis como vetores de alta dimensionalidade
- Converta-os em vetores de variações a partir da média
- Pegue o produto escalar (cosseno do ângulo entre eles) dos dois vetores
- Divida pelo tamanho da amostra



Interpretar covariância é difícil

- Sabemos que uma pequena covariância, próxima de 0, significa que não há muito correlação entre as duas variáveis.
- E grandes covariâncias - ou seja, longe de 0 (pode ser negativo para inverso relacionamentos) significa que há uma correlação
- Mas quão grande é “grande”?



É aí que entra a correlação!

- Apenas divide a covariância pelos desvios padrão de ambas as variáveis, e isso normaliza as coisas.
- Portanto, uma correlação de -1 significa uma correlação inversa perfeita
- Correlação de 0: sem correlação
- Correlação 1: correlação perfeita



Lembre-se: a correlação não implica causalidade!

- Somente um experimento controlado e randomizado pode fornecer informações sobre causalidade.
- Use a correlação para decidir quais experimentos realizar!



Vamos ver alguns exemplos



Probabilidade Condicional



Probabilidade Condicional

- Se eu tiver dois eventos que dependem um do outro, qual é a probabilidade que ambos irão ocorrer?
- Notação: $P(A, B)$ é a probabilidade de A e B ocorrerem ambos
- $P(B | A)$: Probabilidade de B, dado que A ocorreu
- Nós sabemos:

$$P(B | A) = \frac{P(A, B)}{P(A)}$$



Por exemplo

- Eu passo aos meus alunos dois testes. 60% dos meus alunos passaram nos dois testes, mas o primeiro teste foi mais fácil - 80% foi aprovado. Qual porcentagem de os alunos que passaram no primeiro teste também passaram o segundo?
- A = passando no primeiro teste, B = passando no segundo teste
- Então, estamos pedindo $P(B | A)$ - a probabilidade de B dado A

$$P(B|A) = \frac{P(A,B)}{P(A)} = \frac{0.6}{0.8} = 0.75$$

- 75% dos alunos que passaram no primeiro teste passaram no segundo.



Vamos ver um exemplo



Teorema de Bayes



Teorema de Bayes

- Agora que você entende a probabilidade condicional, você pode entender o Teorema de Bayes:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Descrição - a probabilidade de A dado B, é a probabilidade de A vezes o probabilidade de B dado A sobre a probabilidade de B.

O principal insight é que a probabilidade de algo que depende de B depende muito sobre a probabilidade básica de B e A. As pessoas ignoram isso o tempo todo.



Caso do Teorema de Bayes

- O teste de drogas é um exemplo comum. Mesmo um “altamente preciso” teste de drogas pode produzir mais falso-positivos do que verdadeiro-positivos.
- Digamos que tenhamos um teste de drogas que possa determinar com precisão identificar usuários de uma droga 99% do tempo, e com precisão tem um resultado negativo para 99% de não usuários. Mas apenas 0,3% da população total realmente usa essa droga.



Teorema de Bayes

- Evento A = É um usuário do medicamento, Evento B = testado positivamente para o medicamento.
- Podemos calcular a partir dessa informação que P (B) é de 1,3% ($0,99 * 0,003 + 0,01 * 0,997$ - a probabilidade de teste positivo se você usar, mais o probabilidade de teste positivo se você não fizer isso).

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} = \frac{0.003 * 0.99}{0.013} = 22.8\%$$

- Então, as chances de alguém ser um usuário real da droga, dado que eles testado positivo é de apenas 22,8%!
- Embora P (B | A) seja alto (99%), não significa que P (A | B) esteja alto.

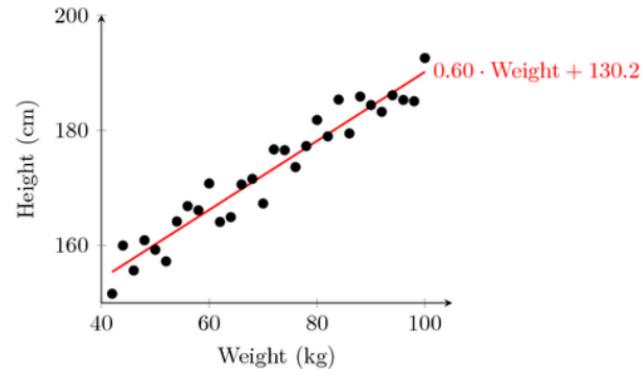


Régressão Linear



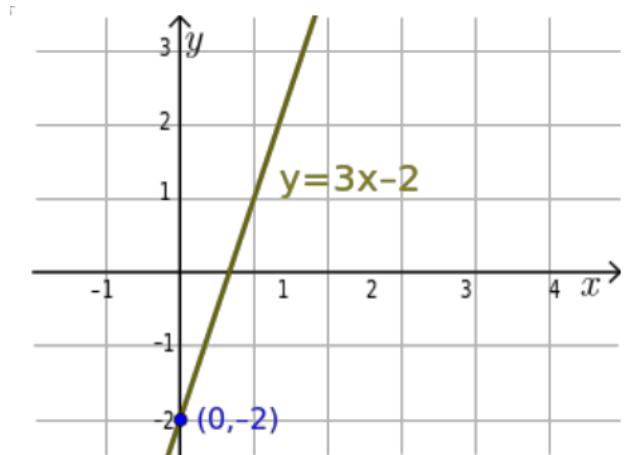
Regressão Linear

- Ajustar uma linha a um conjunto de dados de observações
- Use esta linha para prever valores não observados
- Eu não sei por que eles chamam de "regressão". É realmente enganador. Você pode usá-lo para prever pontos no futuro, o passado, tanto faz. Na verdade, o tempo geralmente não tem nada a ver com isso.



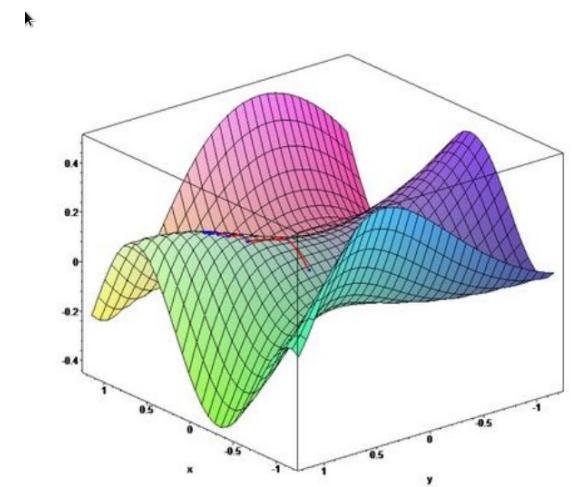
Régressão Linear: como funciona?

- “Mínimos quadrados” minimiza a soma dos erros quadrados.
- Isto é o mesmo que maximizar a probabilidade dos dados observados se você começar a pensar no problema em termos de probabilidades e probabilidades funções de distribuição
- Isso às vezes é chamado de “estimativa de máxima verossimilhança”



Mais de uma maneira de fazer isso

- Gradiente descendente é um método alternativo aos mínimos quadrados.
- Basicamente itera para encontrar a linha que melhor segue os contornos definidos pelos dados.
- Pode fazer sentido quando se lida com dados 3D
- Fácil de experimentar em Python e apenas comparar resultados para mínimos quadrados
 - Mas geralmente os mínimos quadrados são perfeitamente boas escolha.



Medição de Erro com R-Quadrado

- Como medimos quão bem nossa linha se ajusta aos nossos dados?
- Medidas de R-Quadrado (coeficiente de determinação):

A fração da variação total em Y que é capturado pelo modelo



Computação R-Quadrado

1,0 -

$$\frac{\text{soma de erros quadrados}}{\text{soma da variação quadrática da média}}$$



Interpretando o R-Quadrado

- Varia de 0 a 1
- 0 é ruim (nenhuma das variações é capturada), 1 é bom (todas as variações são capturadas).



Vamos ver um exemplo



Métodos Bayesianos



Lembre-se do teorema de Bayes?

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

- Vamos usá-lo para aprendizagem de máquina! Eu quero um classificador de spam.
- Exemplo: como podemos expressar a probabilidade de um email ser spam se contém a palavra "free"?

$$P(\text{Spam} | \text{Free}) = \frac{P(\text{Spam})P(\text{Free} | \text{Spam})}{P(\text{Free})}$$

- O numerador é a probabilidade de uma mensagem ser spam e conter a palavra "free" (isso é sutilmente diferente do que procuramos)
- O denominador é a probabilidade geral de um email contendo a palavra "free". (Equivalente a $P(\text{Free} | \text{Spam})P(\text{Spam}) + P(\text{Free} | \text{Not Spam})P(\text{Not Spam})$)
- Então, juntos - essa proporção é a % de emails com a palavra "free" que são spam.

E todas as outras palavras?

- Podemos construir $P(\text{Spam} | \text{Word})$ para cada palavra (significativa) que encontramos durante o treinamento
- Depois, multiplique-os juntos ao analisar um novo e-mail para obter a probabilidade de ser spam.
- Pressupõe a presença de palavras diferentes independentes uns dos outros - uma razão pela qual isso é chamado "Naïve Bayes".



Soa como um monte de trabalho.

- Scikit-learn auxilia nesse trabalho
- O CountVectorizer nos permite operar em muitas palavras de uma só vez, e MultinomialNB faz todo o trabalho pesado em Naïve Bayes.
- Vamos treiná-lo em conjuntos conhecidos de spam e e-mails "ham" (sem spam)
 - Então, isso é aprendizado supervisionado!
- Vamos fazer isso



Exemplo

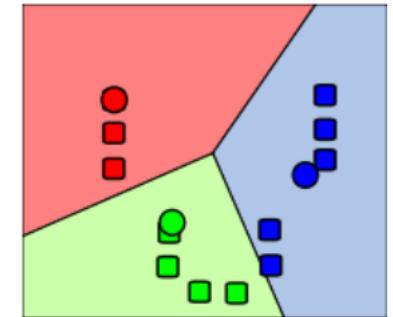


K-Means Clustering



K-Means clusters

- Tenta dividir os dados em grupos K que são mais próximos de K centroids
- Aprendizado não supervisionado - usa somente posições de cada ponto de dados
- Pode descobrir grupos interessantes de pessoas / coisas / comportamento
 - Exemplo: onde vivem milionários?
 - Que gêneros de música / filmes / etc. naturalmente vem dos dados?
 - Crie seus próprios estereótipos de dados demográficos



K-Means Clusters

- Funcionamento simples.
 - Escolher aleatoriamente K centróides (k-means)
 - Atribuir cada ponto de dados ao centróide mais próximo
 - Recompute os centróides com base na posição média de cada ponto centróide
 - Iterar até que os pontos parem de mudar a designação para centróides
- Se você quiser prever o cluster para novos pontos, basta encontrar o centróide que eles estão mais próximos.



Exemplo gráfico



Desafios do K-Means Clustering

- Escolhendo K
 - Tente aumentar os valores de K até que você pare de obter grandes reduções no erro quadrado (distâncias de cada ponto aos seus centróides)
- Evitar mínimos locais
 - A escolha aleatória dos centróides iniciais pode produzir resultados diferentes
 - Execute algumas vezes apenas para garantir que seus resultados iniciais não sejam malucos
- Rotulando os clusters
 - O K-Means não tenta atribuir nenhum significado aos clusters encontrados
 - Cabe a você pesquisar os dados e tentar determinar



Exemplo

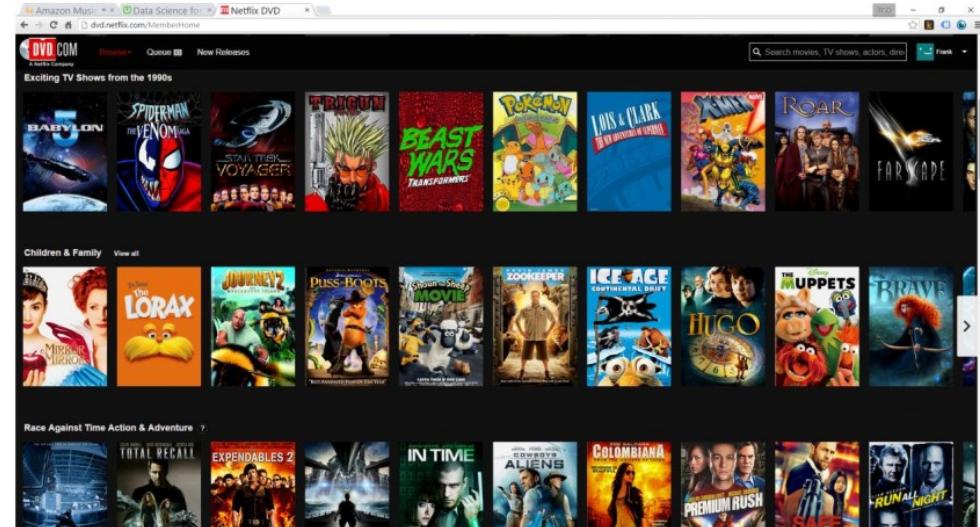
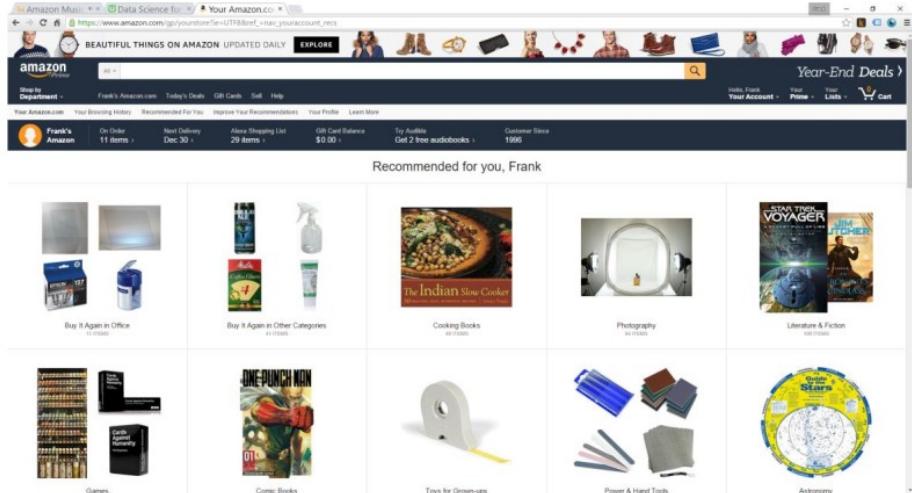


Sistemas de Recomendação

paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark





paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



Filtragem Colaborativa Baseada no Usuário

- Constrói uma matriz de coisas que cada usuário comprou / visualizou / avaliou
- Computa pontuações de similaridade entre usuários
- Encontra usuários semelhantes a você
- Recomenda coisas que outros compraram / visualizaram / classificaram e que você ainda não.



Problemas com Filtragem Colaborativa Baseada no Usuário

- As pessoas são inconstantes; gostos mudam
- Geralmente há muito mais pessoas do que coisas
- As pessoas fazem coisas ruins



E se nós basearmos recomendações em relacionamentos entre as coisas em vez de pessoas?

- Um filme será sempre o mesmo filme - não muda
- Geralmente há menos coisas que pessoas (menos computação para fazer)
- Difícil de enganar o sistema

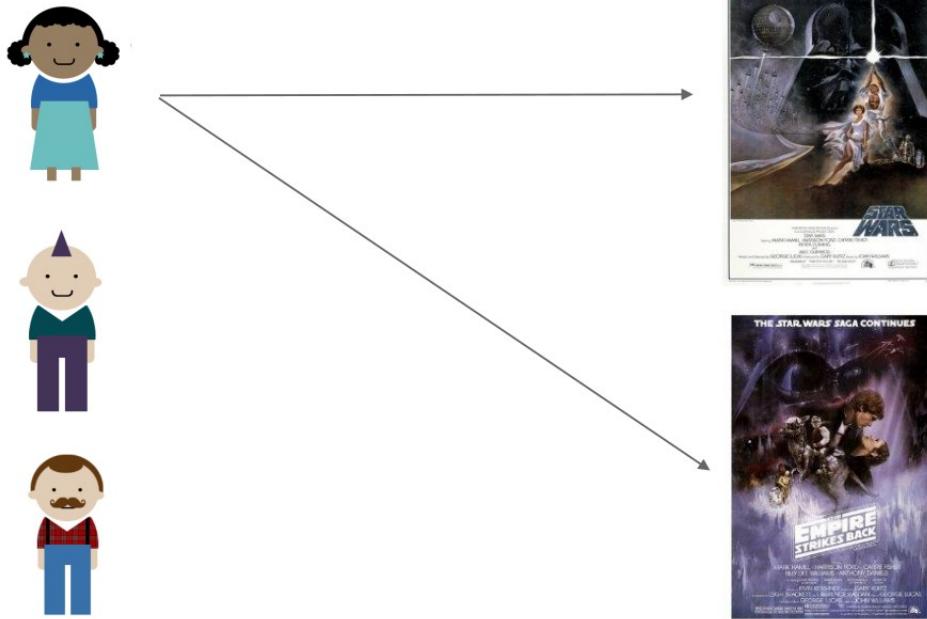


Filtragem Colaborativa Baseada em Itens

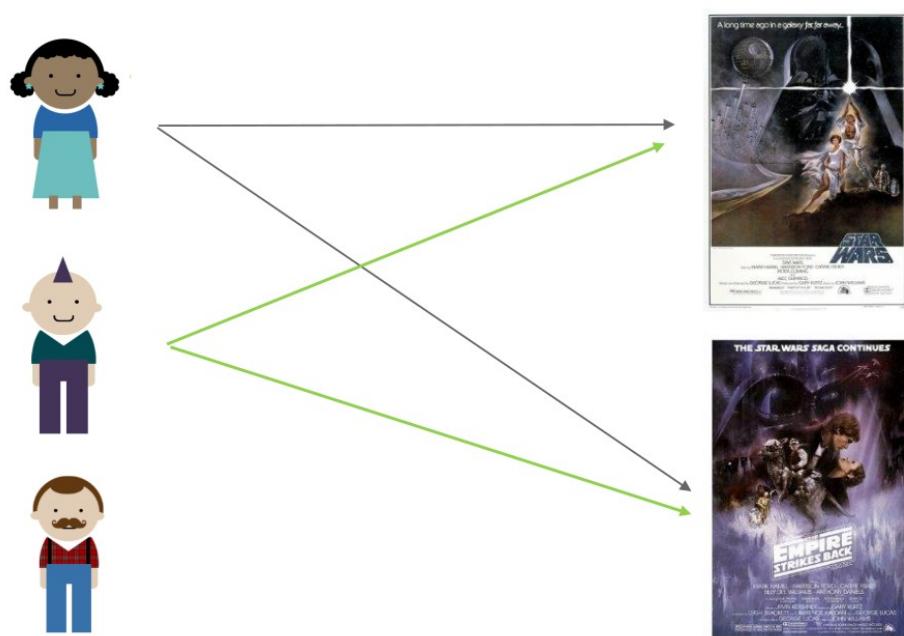
- Encontre cada par de filmes que foram assistidos pela mesma pessoa
- Avalie a similaridade de suas classificações em todos os usuários que assistiram ambos
- Classificar por filme e depois por força de similaridade
- (Esta é apenas uma maneira de fazer isso!)



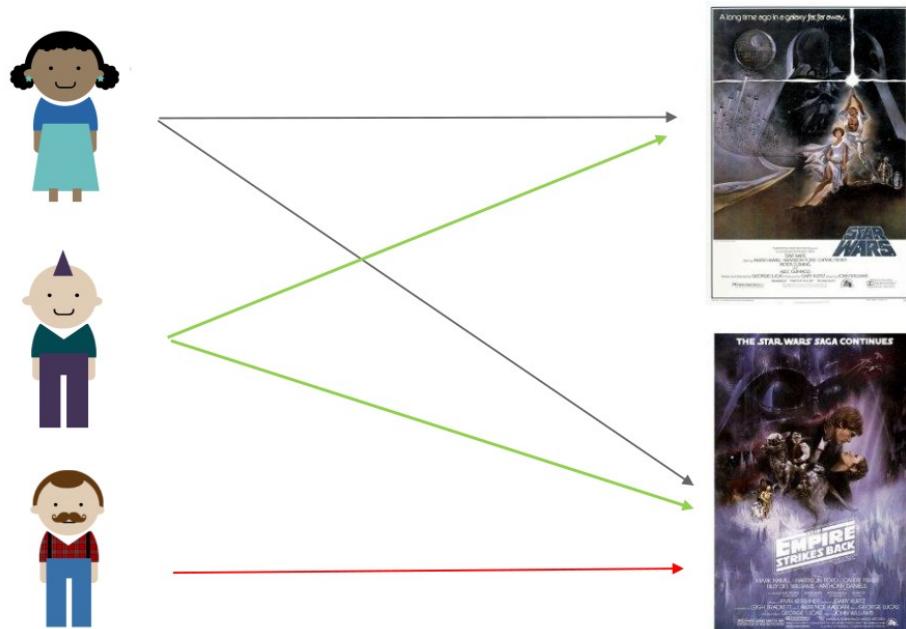
Filtragem Colaborativa Baseada em Itens



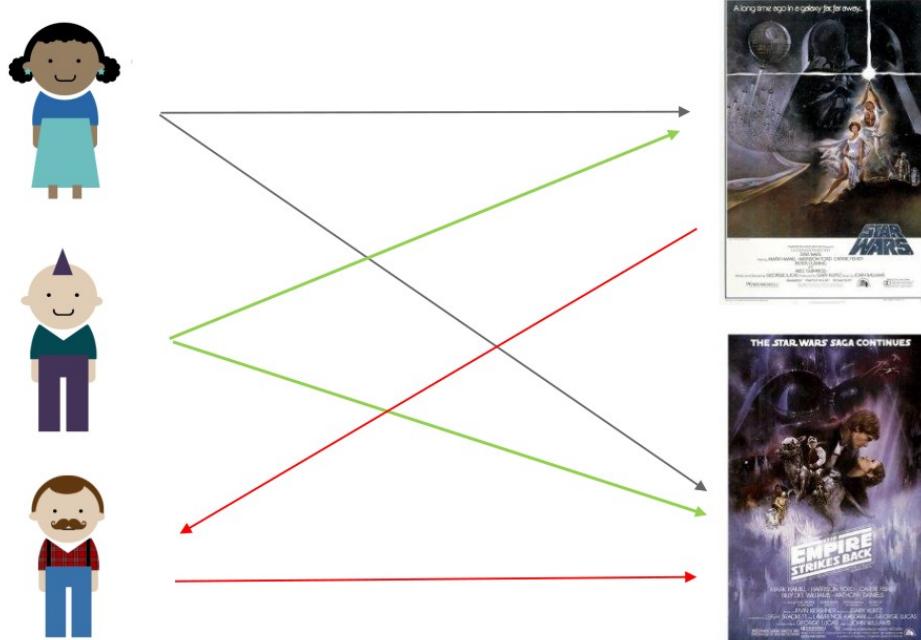
Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Vamos fazer isso

- Em seguida, usaremos o Python para criar "semelhanças de filme" reais usando o Conjunto de dados MovieLens.
 - Além de ser importante para a filtragem colaborativa baseada em itens, esses resultados são valiosos em si mesmos - pense que "as pessoas que gostaram do X também gostaram de Y"
- São dados do mundo real e encontraremos problemas do mundo real
- Então, usaremos esses resultados para criar recomendações de filmes para indivíduos



Exemplo



K-Nearest Neighbor

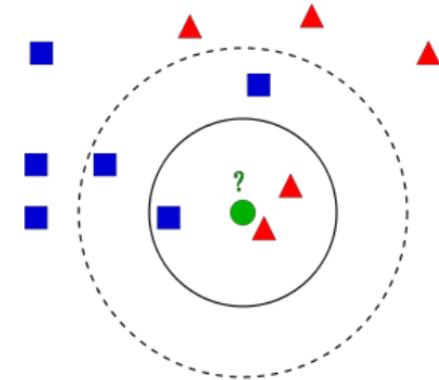


K-Nearest Neighbor

Usado para classificar novos pontos de dados com base em "distância" para dados conhecidos

Encontre os K vizinhos mais próximos, com base na sua métrica de distância

Deixe que todos votem na classificação



É realmente simples

- Embora seja um dos modelos de aprendizado de máquina mais simples que existe, ainda se qualifica como “aprendizado supervisionado”.
- Mas vamos fazer algo mais complexo com isso
- Semelhanças cinematográficas baseadas apenas em metadados!

Customers Who Watched This Item Also Watched



Exemplo



Modelos de Escolha Discreta



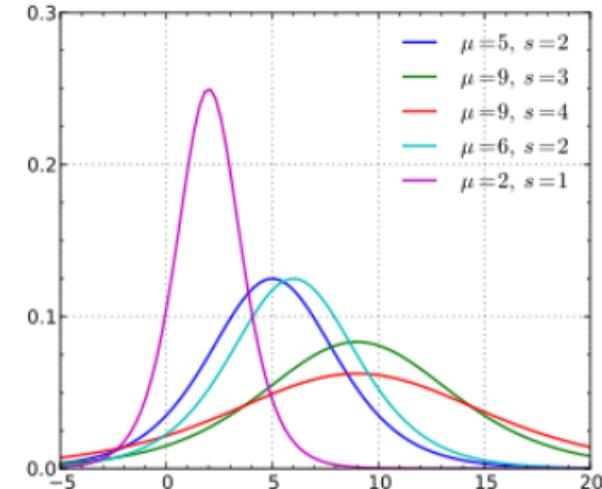
Modelos de Escolha Discreta

- Prevêm algumas escolhas entre alternativas discretas
 - Eu pego o trem, ônibus ou carro para trabalhar hoje? (Escolha Multinomial)
 - Para qual faculdade eu vou? (Multinomial)
 - Vou trair meu cônjuge? (Binário)
- As alternativas devem ser finitas, exaustivas e mutuamente exclusivas



Modelos de Escolha Discreta

- Usa algum tipo de regressão nos atributos relevantes
 - Atributos das pessoas
 - Variáveis das alternativas
- Geralmente usa modelos Logit ou Probit
 - Regressão Logística, Modelo Probit
 - Baseado em alguma função de utilidade você define
 - Similar - um usa a distribuição logística, o Probit usa distribuição normal. Logística parece muito com normal, mas com caudas mais gordas (maior curtose)



Limpando seus dados



Limpando seus dados

- Muito do seu tempo como um cientista de dados será gasto preparando e “limpando” seus dados
- Outliers
- Dados ausentes
- Dados maliciosos
- Dados errados
- Dados irrelevantes
- Dados inconsistentes
- Formatação



Garbage In, Garbage Out

- Olhe seus dados! Examine!
- Questione seus resultados!
 - Sempre faça isso - não apenas quando você não tenha um resultado que você goste!



Vamos analisar alguns dados de log da web.

- O que eu quero: as páginas mais populares no meu site de notícias sem fins lucrativos.
- Quão difícil isso pode ser?



Exemplo



Apache Spark

Introdução à MLLib



Alguns recursos do MLLib

- Extração de características
 - Freqüência a termo / Freqüência de documento inversa útil para pesquisa
- Estatísticas básicas
 - Qui-quadrado, correlação de Pearson ou Spearman, min, max, média, variância
- Regressão Linear, Regressão Logística
- Máquinas de vetores de suporte
- Classificador Naïve Bayes
- Árvores de decisão
- K-significa clusters
- Análise de componentes principais, decomposição de valores singulares
- Recomendações usando Mínimos Quadrados Alternados



Exemplo

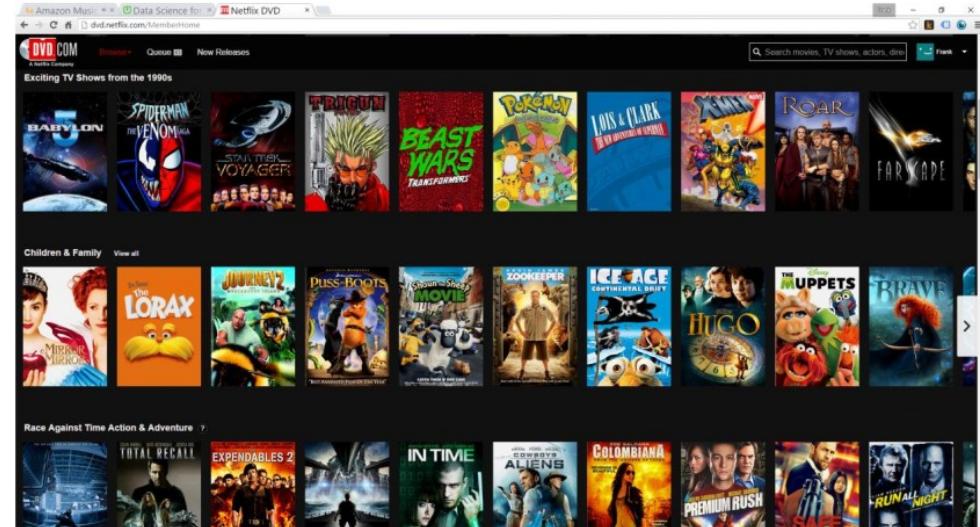
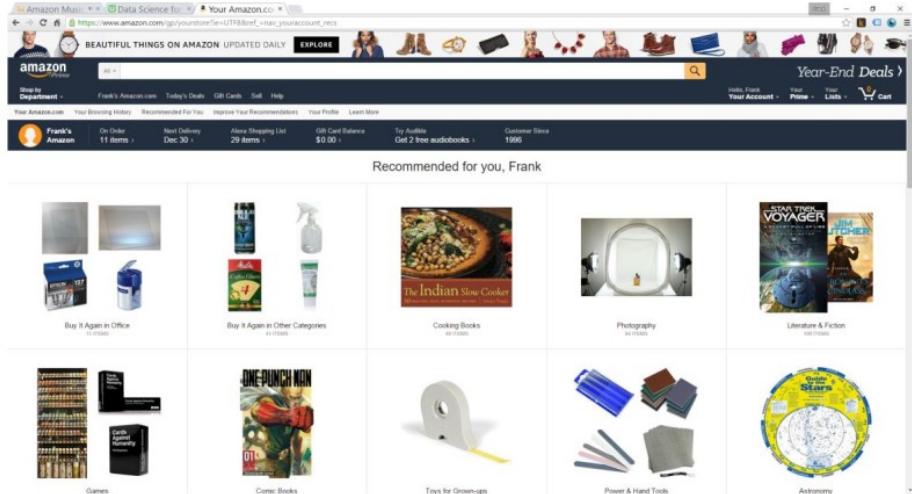


Sistemas de Recomendação

paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark





paulo.rodrigues@konfidentia.com

Big Data e Machine Learning com Hadoop
e Spark



Filtragem Colaborativa Baseada no Usuário

- Constrói uma matriz de coisas que cada usuário comprou / visualizou / avaliou
- Computa pontuações de similaridade entre usuários
- Encontra usuários semelhantes a você
- Recomenda coisas que outros compraram / visualizaram / classificaram e que você ainda não.



Problemas com Filtragem Colaborativa Baseada no Usuário

- As pessoas são inconstantes; gostos mudam
- Geralmente há muito mais pessoas do que coisas
- As pessoas fazem coisas ruins



E se nós basearmos recomendações em relacionamentos entre as coisas em vez de pessoas?

- Um filme será sempre o mesmo filme - não muda
- Geralmente há menos coisas que pessoas (menos computação para fazer)
- Difícil de enganar o sistema

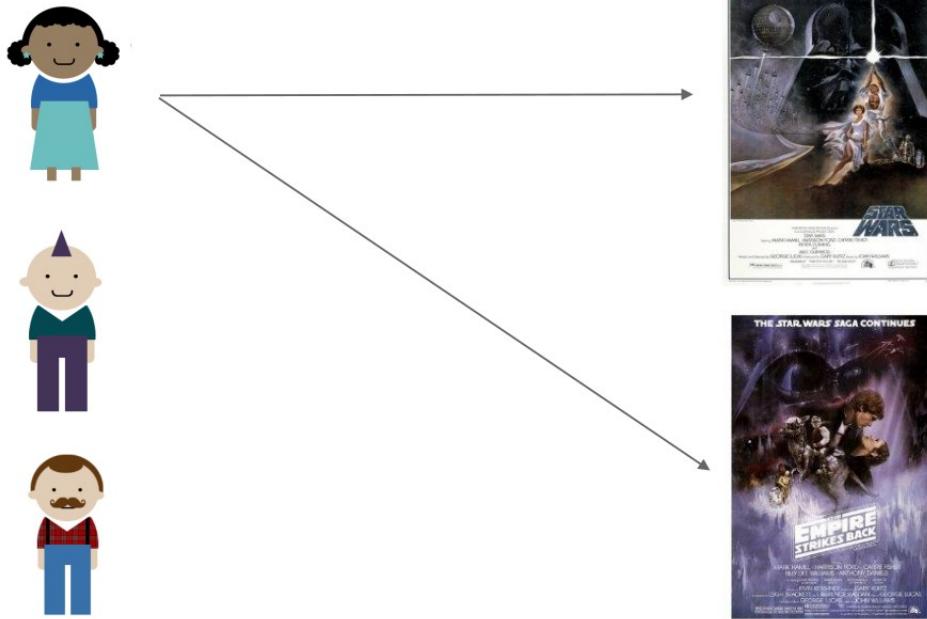


Filtragem Colaborativa Baseada em Itens

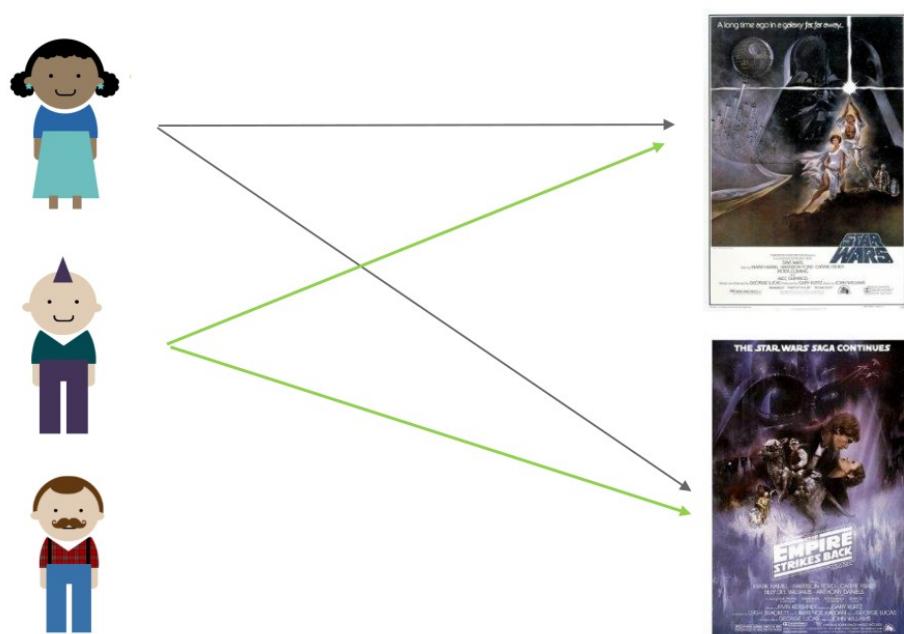
- Encontre cada par de filmes que foram assistidos pela mesma pessoa
- Avalie a similaridade de suas classificações em todos os usuários que assistiram ambos
- Classificar por filme e depois por força de similaridade
- (Esta é apenas uma maneira de fazer isso!)



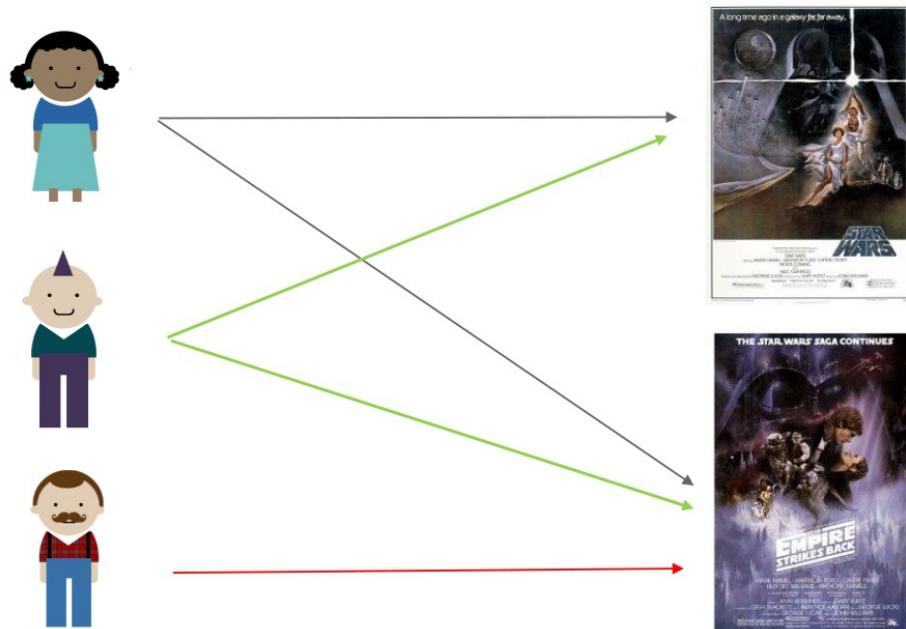
Filtragem Colaborativa Baseada em Itens



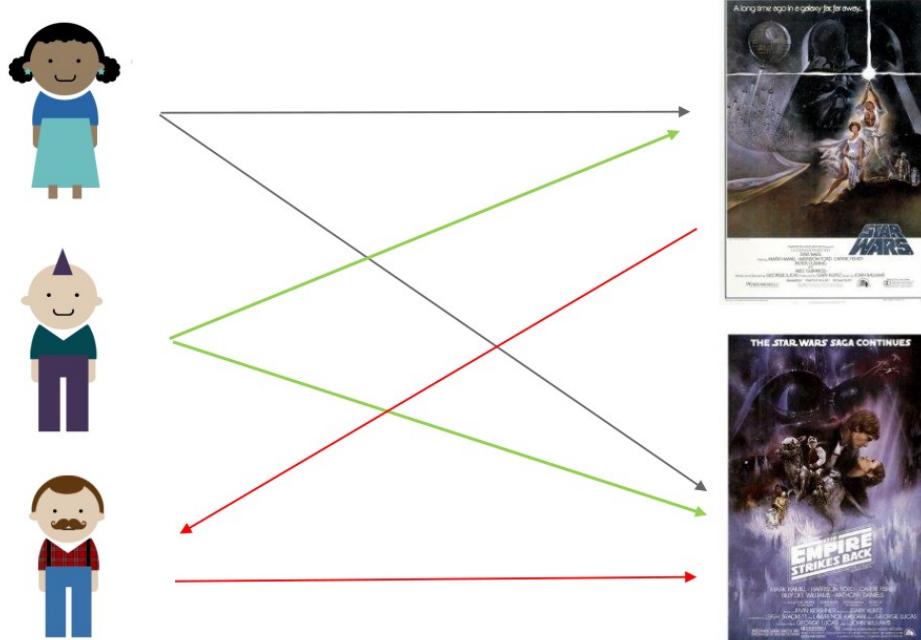
Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Vamos fazer isso

- Em seguida, usaremos o Python para criar "semelhanças de filme" reais usando o Conjunto de dados MovieLens.
 - Além de ser importante para a filtragem colaborativa baseada em itens, esses resultados são valiosos em si mesmos - pense que "as pessoas que gostaram do X também gostaram de Y"
- São dados do mundo real e encontraremos problemas do mundo real
- Então, usaremos esses resultados para criar recomendações de filmes para indivíduos



Exemplo



K-Nearest Neighbor

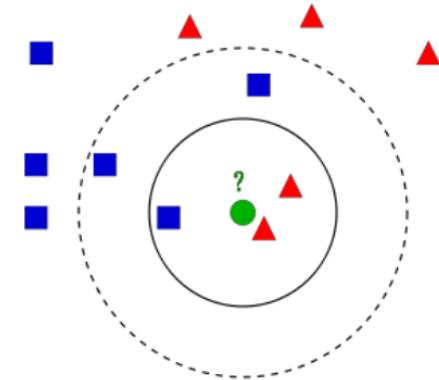


K-Nearest Neighbor

Usado para classificar novos pontos de dados com base em "distância" para dados conhecidos

Encontre os K vizinhos mais próximos, com base na sua métrica de distância

Deixe que todos votem na classificação



É realmente simples

- Embora seja um dos modelos de aprendizado de máquina mais simples que existe, ainda se qualifica como “aprendizado supervisionado”.
- Mas vamos fazer algo mais complexo com isso
- Semelhanças cinematográficas baseadas apenas em metadados!

Customers Who Watched This Item Also Watched



Exemplo



Modelos de Escolha Discreta



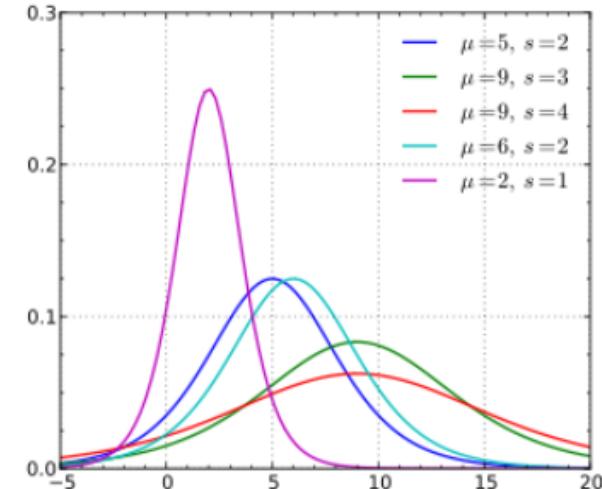
Modelos de Escolha Discreta

- Prevêm algumas escolhas entre alternativas discretas
 - Eu pego o trem, ônibus ou carro para trabalhar hoje? (Escolha Multinomial)
 - Para qual faculdade eu vou? (Multinomial)
 - Vou trair meu cônjuge? (Binário)
- As alternativas devem ser finitas, exaustivas e mutuamente exclusivas



Modelos de Escolha Discreta

- Usa algum tipo de regressão nos atributos relevantes
 - Atributos das pessoas
 - Variáveis das alternativas
- Geralmente usa modelos Logit ou Probit
 - Regressão Logística, Modelo Probit
 - Baseado em alguma função de utilidade você define
 - Similar - um usa a distribuição logística, o Probit usa distribuição normal. Logística parece muito com normal, mas com caudas mais gordas (maior curtose)



Limpando seus dados



Limpando seus dados

- Muito do seu tempo como um cientista de dados será gasto preparando e “limpando” seus dados
- Outliers
- Dados ausentes
- Dados maliciosos
- Dados errados
- Dados irrelevantes
- Dados inconsistentes
- Formatação



Garbage In, Garbage Out

- Olhe seus dados! Examine!
- Questione seus resultados!
 - Sempre faça isso - não apenas quando você não tenha um resultado que você goste!



Vamos analisar alguns dados de log da web.

- O que eu quero: as páginas mais populares no meu site de notícias sem fins lucrativos.
- Quão difícil isso pode ser?



Exemplo



Apache Spark

Introdução à MLLib



Alguns recursos do MLLib

- Extração de características
 - Freqüência a termo / Freqüência de documento inversa útil para pesquisa
- Estatísticas básicas
 - Qui-quadrado, correlação de Pearson ou Spearman, min, max, média, variância
- Regressão Linear, Regressão Logística
- Máquinas de vetores de suporte
- Classificador Naïve Bayes
- Árvores de decisão
- K-significa clusters
- Análise de componentes principais, decomposição de valores singulares
- Recomendações usando Mínimos Quadrados Alternados



Exemplo



