GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

Greenplum机器学习工具集和案例

姚延栋
Pivotal 研发技术总监

2017.thegiac.com

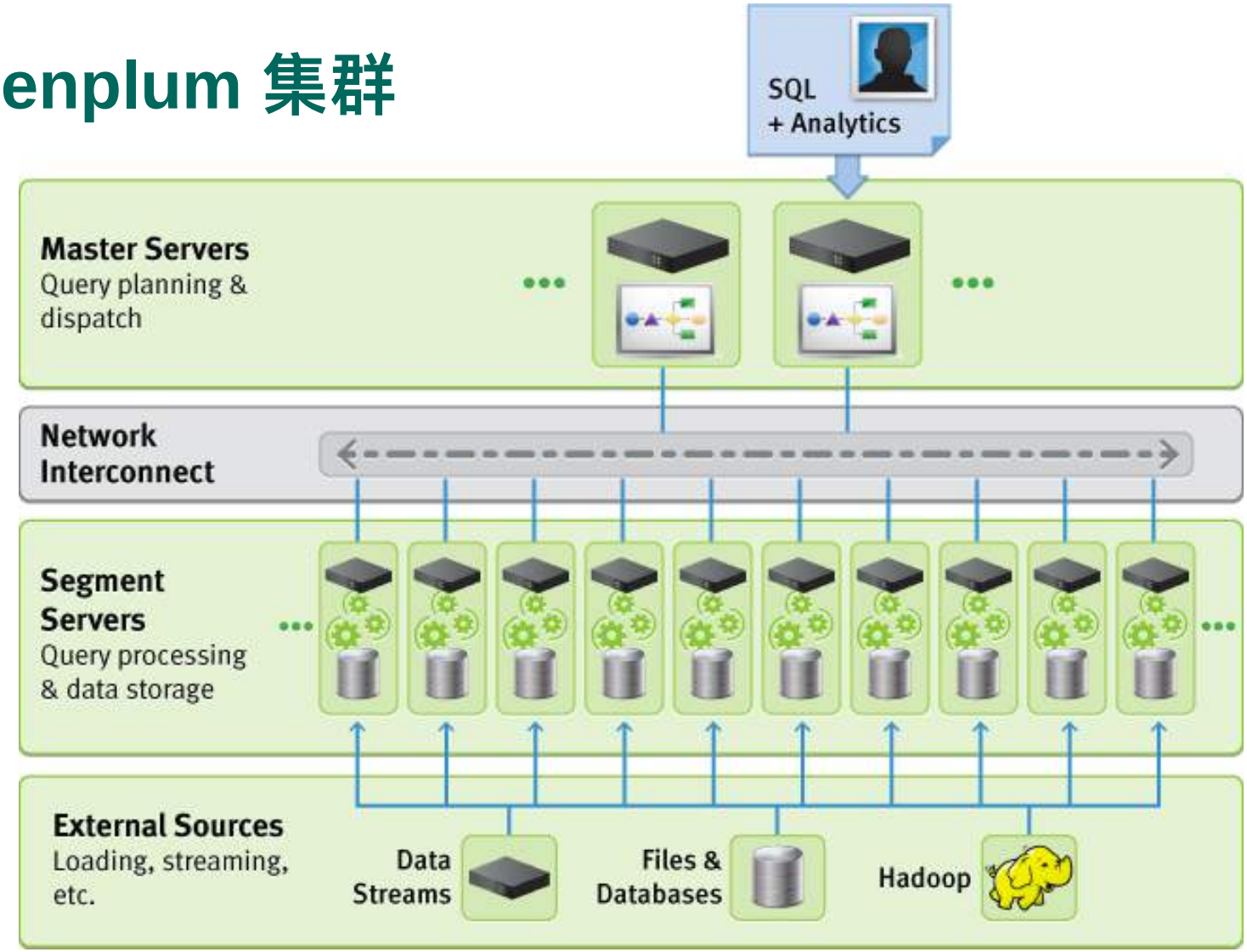
大纲

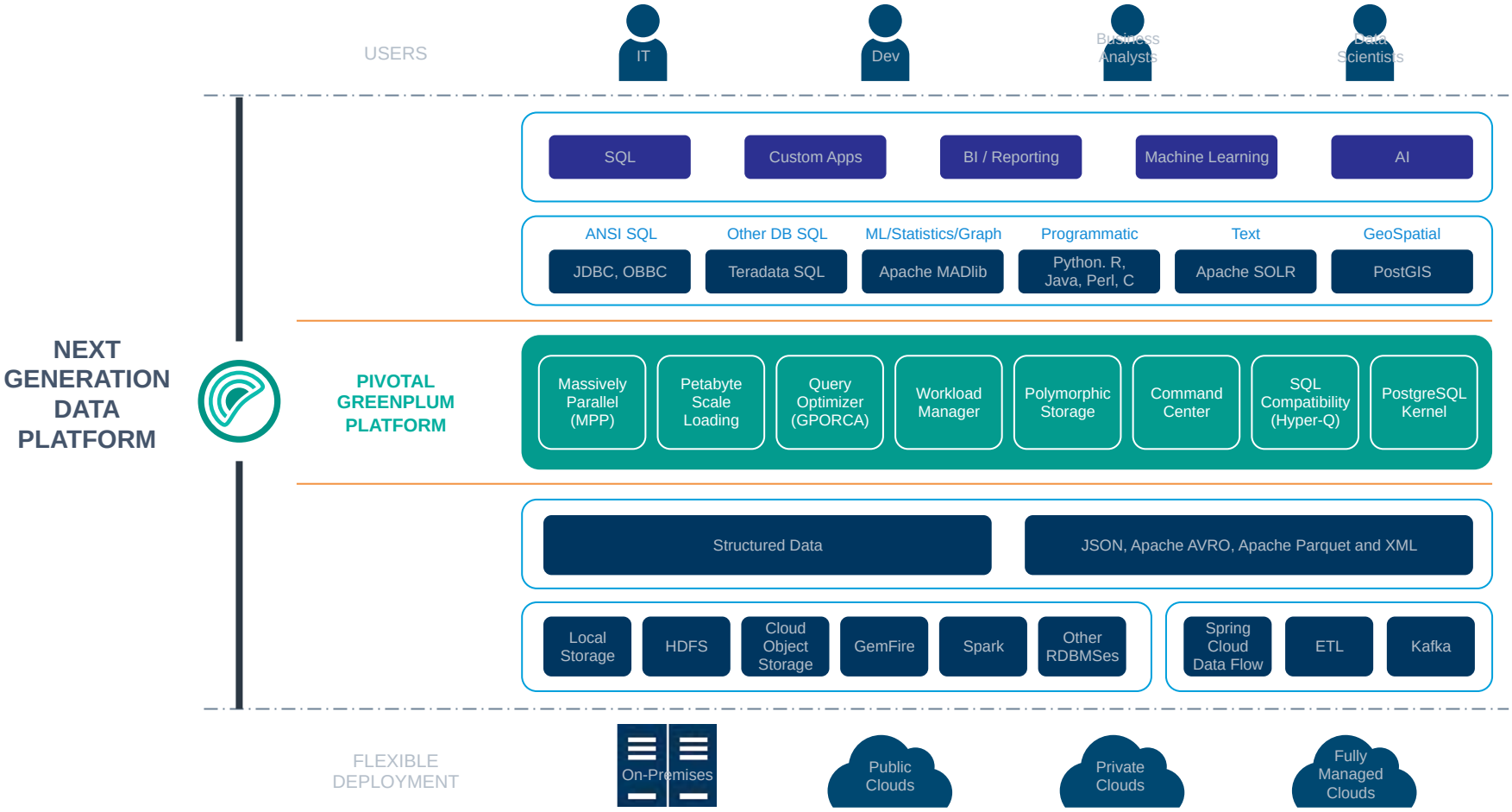
- Greenplum 大数据平台
- Greenplum 机器学习工具
- Greenplum 机器学习案例

Pivotal

Greenplum：新一代开源大数据平台

Greenplum 集群




USERS
IT
Dev
Business Analysts
Data Scientists
SQL
Custom Apps
BI / Reporting
Machine Learning
AI
ANSI SQL
Other DB SQL
ML/Statistics/Graph
Programmatic
Text
GeoSpatial
JDBC, OBBC
Teradata SQL
Apache MADlib
Python. R, Java, Perl, C
Apache SOLR
PostGIS
NEXT GENERATION DATA PLATFORM
PIVOTAL GREENPLUM PLATFORM
Massively Parallel (MPP)
Petabyte Scale Loading
Query Optimizer (GPORCA)
Workload Manager
Polymorphic Storage
Command Center
SQL Compatibility (Hyper-Q)
PostgreSQL Kernel
Structured Data
JSON, Apache AVRO, Apache Parquet and XML
Local Storage
HDFS
Cloud Object Storage
GemFire
Spark
Other RDBMSes
Spring Cloud Data Flow
ETL
Kafka
FLEXIBLE DEPLOYMENT
On-Premises
Public Clouds
Private Clouds
Fully Managed Clouds

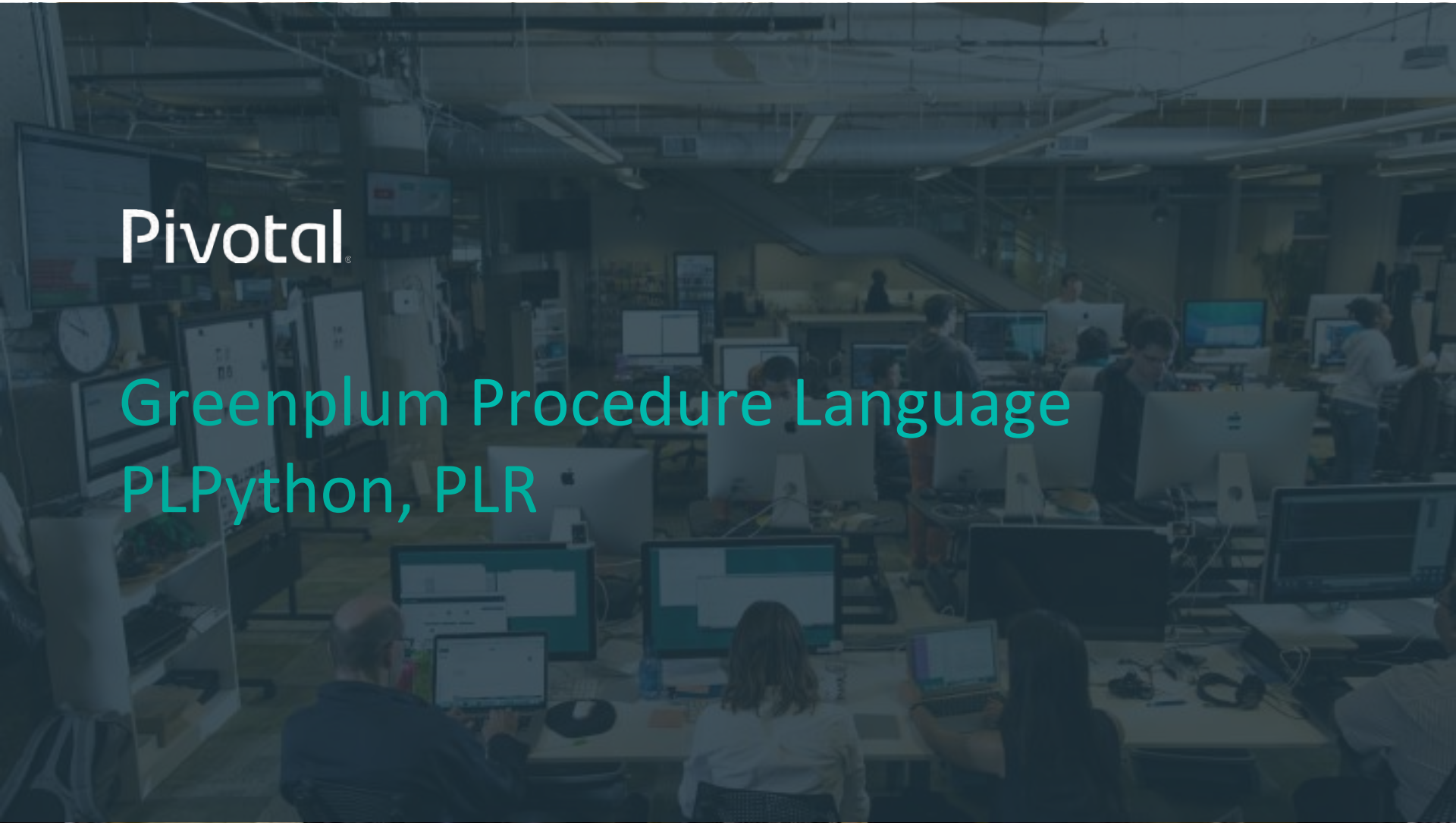
Greenplum 大数据平台

- 一次打包，到处运行：裸机、私有云、公有云
- 各种数据源：**Hadoop**、**S3**、数据库、文件、**Spark**、**Kafka**
- 各种数据格式：结构化、半结构化（**JSON/XML/Hstore)**、非结构化
- 强大内核：**MPP**、优化器、多态存储、灵活分区、高速加载、**PG**内核
- 强大的灵活性、可扩展：**PL/X**、**Extension**、**PXF**、外部表机制
- 完善的标准支持：**SQL**、**JDBC**、**ODBC**
- 集成数据平台：**BI/DW**、文本、**GIS**、图、图像、机器学习
- 开放源代码，持续大力投入
- 敏捷方法学：快速迭代、持续发布、质量内建
- 企业级稳定性，成熟生态系统

Greenplum：机器学习工具集

Greenplum 机器学习工具集

- PL/X：各种语言实现自定义函数（存储过程）
- MADLib: 数据挖掘、统计分析、图（Graph）等算法
- GPText：文本检索和分析
- GeoSpatial：地理信息数据分析
- Image: 图像数据分析



Greenplum Procedure Language
PLPython, PLR

PL/Python 例子

- *CREATE TABLE sales (id int, year int, qtr int, day int, region text)*
  DISTRIBUTED BY (id) ;
  INSERT INTO sales VALUES
  (1, 2014, 1,1, 'usa'),
  (2, 2002, 2,2, 'europe'),
  (3, 2014, 3,3, 'asia'),
  (4, 2014, 4,4, 'usa'),
  (5, 2014, 1,5, 'europe'),
  (6, 2014, 2,6, 'asia'),

- *CREATE OR REPLACE FUNCTION mypytest(index integer) RETURNS text*
  AS $$
  *rv = plpy.execute("SELECT * FROM sales ORDER BY id", 5)*
  region = rv[index]["region"]
  return region
  $$ language plpythonu;

- *SELECT mypytest(2) ;*

Module Name	Description/Used For
Beautiful Soup	Navigating HTML and XML
Gensim	Topic modeling and document indexing
Keras	Deep learning
Lifelines	Survival analysis
lxml	XML and HTML processing
NLTK	Natural language toolkit
NumPy	Scientific computing
Pandas	Data analysis
Pattern-en	Part-of-speech tagging
pyLDAvis	Interactive topic model visualization
PyMC3	Statistical modeling and probabilistic machine learning
scikit-learn	Machine learning data mining and analysis
SciPy	Scientific computing
spaCy	Large scale natural language processing
StatsModels	Statistical modeling
Tensorflow	Numerical computation using data flow graphs
XGBoost	Gradient boosting, classifying, ranking

PL/R 例子

- *CREATE OR REPLACE FUNCTION r_norm(n integer, mean float8, std_dev float8) RETURNS float8[ ] AS*

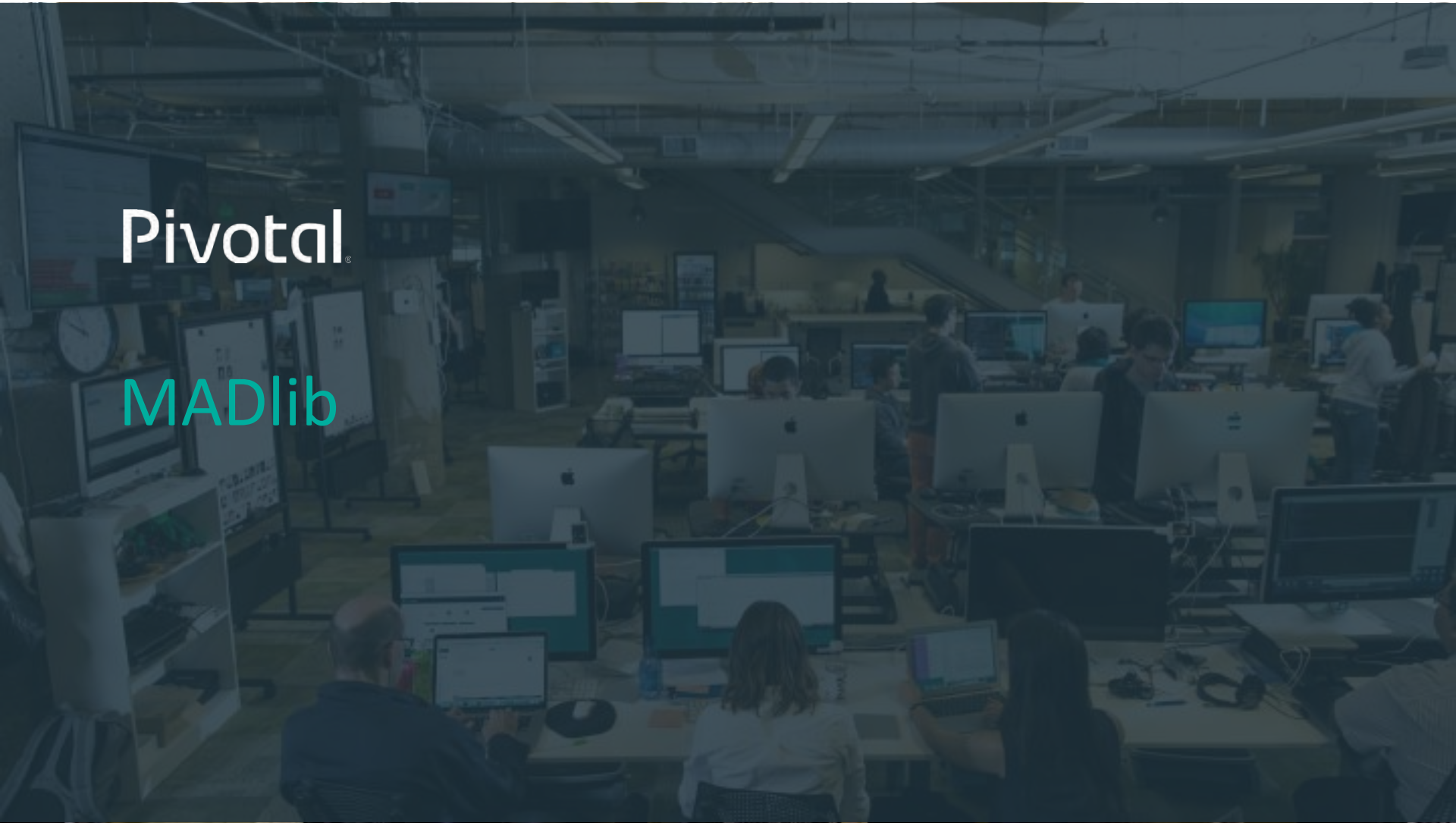
```
$$
  x<-rnorm(n,mean,std_dev)
  return(x)
$$
LANGUAGE 'plr';
```

- *CREATE TABLE test_norm_var*
  AS SELECT id, ***r_norm(10,0,1)*** *as x*
  FROM (SELECT generate_series(1,30:: bigint) AS ID) foo
  DISTRIBUTED BY (id);

abind	gplots	quantreg
adabag	gtable	R2jags
arm	gtools	R6
assertthat	hciust	randomForest
BH	hms	RColorBrewer
bitops	igraph	Rcpp
car	labeling	RcppEigen
caret	lattice	readr
caTools	lazyeval	reshape2
coda	lme4	rjags
colorspace	lmtest	RobustRankAggreg
curl	magrittr	ROCR
data.table	MASS	rpart
DBI	Matrix	RPostgreSQL
dichromat	MCMCPack	sandwich
digest	minqa	scales
dplyr	mts	SparseM
e1071	munsell	stringi
forecast	neuralnet	stringr
foreign	nloptr	survival
gdata	nnet	tibble
ggplot2	pbkrtest	tseries
glmnet	plyr	zoo

适用场景

- 适合模型应用于数据子集的场景，并行执行效率非常高
- 如果节点间数据通讯，使用 MADlib


Pivotal
MADlib

MADlib 基于SQL的数据库内置的机器学习库

Apache上的开源项目

- 发布了 6 个版本
- Apache 顶级项目

MPP系统上的可扩展应用

Pivotal Greenplum

PostgreSQL

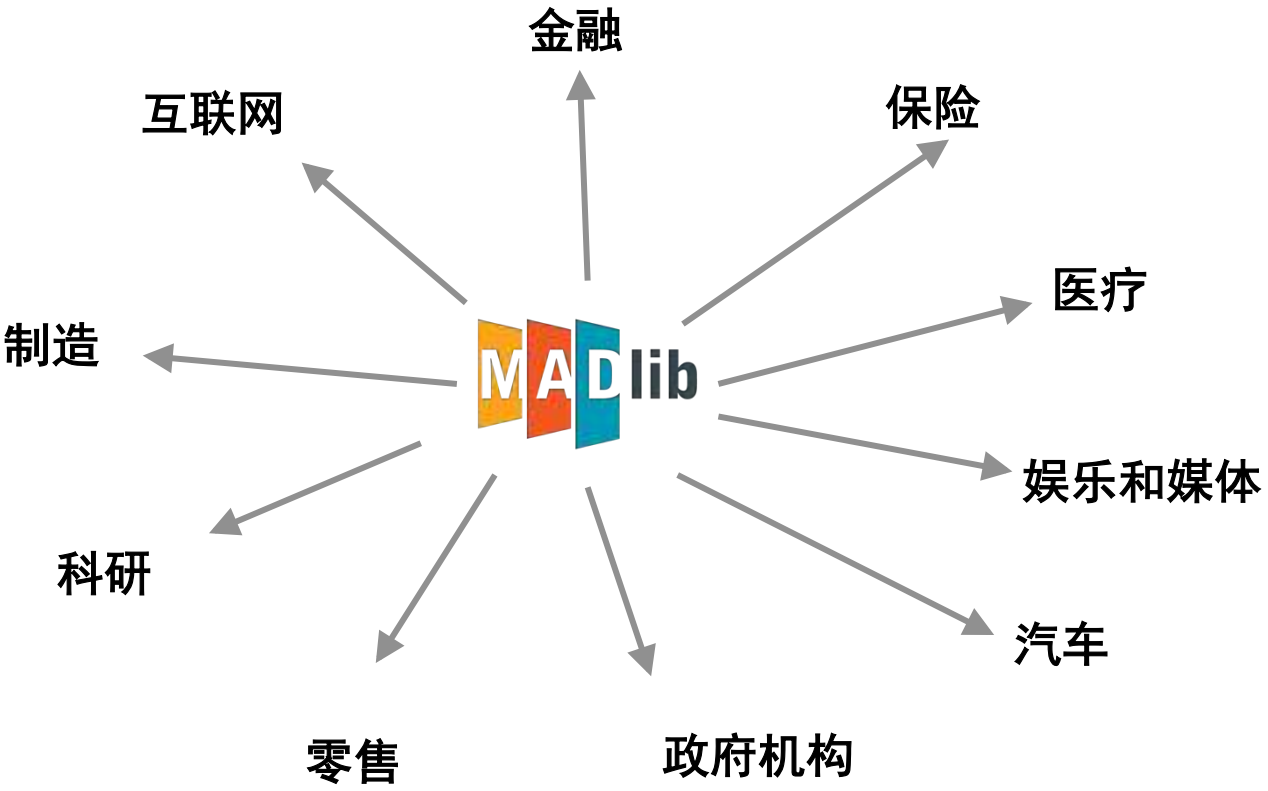
强大的分析能力

- 机器学习
- 图形分析
- 统计分析

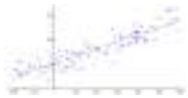
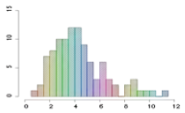
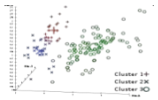
历史回顾

创始于2011年
EMC/Greenplum
Joe Hellerstein from Univ. of California, Berkeley.

MADlib 用户和场景



MADlib 功能

Supervised Learning
Neural Networks
Support Vector Machines (SVM)
Regression Models
- Clustered Variance
- Cox-Proportional Hazards Regression
- Elastic Net Regularization
- Generalized Linear Models
- Linear Regression
- Logistic Regression
- Marginal Effects
- Multinomial Regression
- Naïve Bayes
- Ordinal Regression
- Robust Variance

Tree Methods
- Decision Tree
- Random Forest

Conditional Random Field (CRF)

Unsupervised Learning
Association Rules (Apriori)
Clustering (k-Means)
Topic Modelling (Latent Dirichlet Allocation)

Nearest Neighbors
- k-Nearest Neighbors

Graph
All Pairs Shortest Path (APSP)
Breadth-First Search
Average Path Length
Closeness Centrality
Graph Diameter
In-Out Degree
PageRank
Single Source Shortest Path (SSSP)
Weakly Connected Components

Utility Functions
- Sparse Linear Systems

Path
PMML Export
Sampling
- Random
- Stratified

Sessionize
Term Frequency for Text Analysis

Time Series Analysis
- ARIMA

Data Types and Transformations
Array and Matrix Operations
Matrix Factorization
- Low Rank
- Singular Value Decomposition (SVD)

Norms and Distance Functions
Sparse Vectors
Principal Component Analysis (PCA)
Encoding Categorical Variables
Pivot
Stemming

ics
otive Statistics
rdinality Estimators
- Correlation and Covariance
- Summary

Inferential Statistics
- Hypothesis Tests

Probability Functions

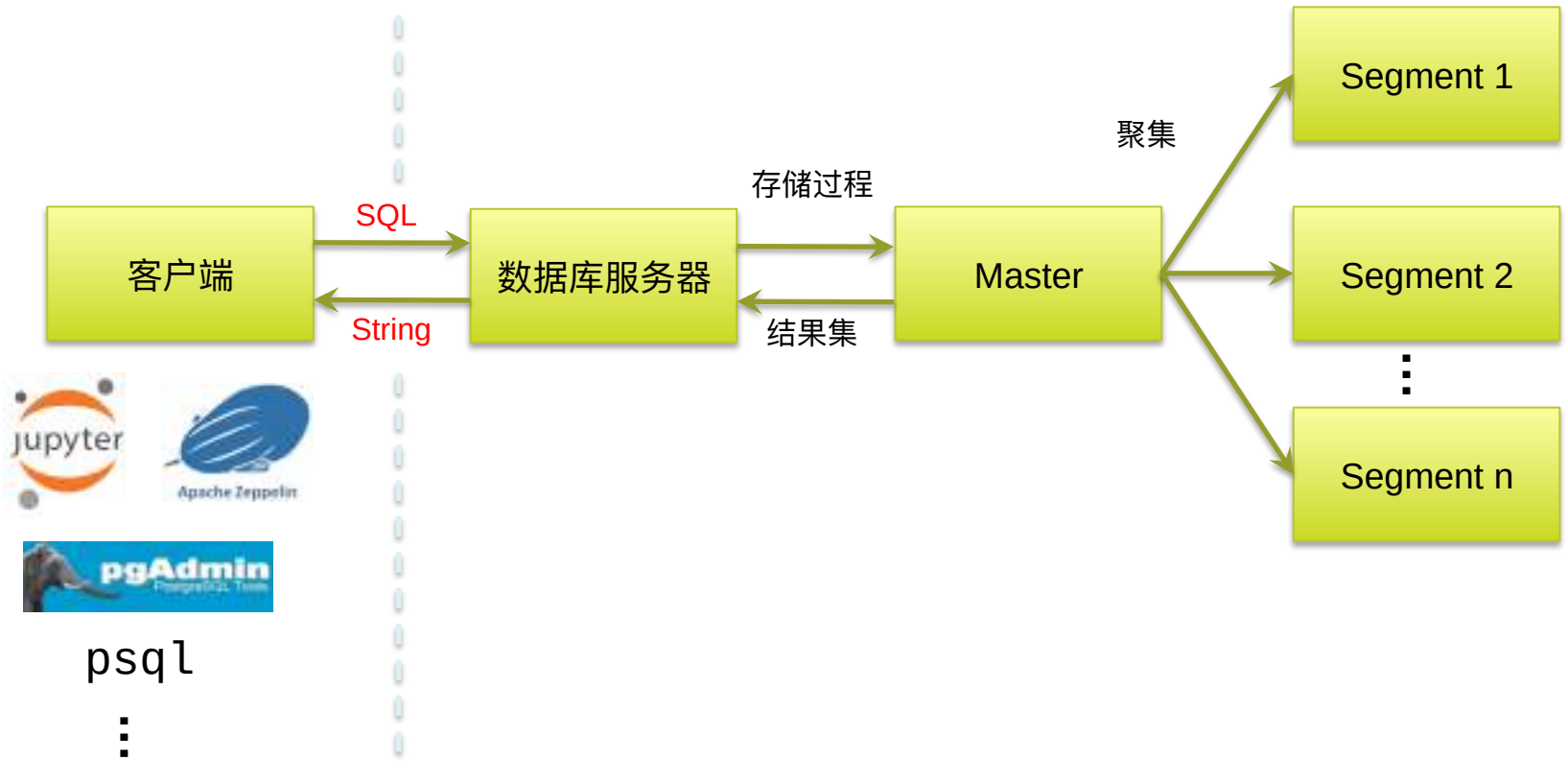
Model Selection
Cross Validation
Prediction Metrics
Train-Test Split

成熟的数据科学学习库

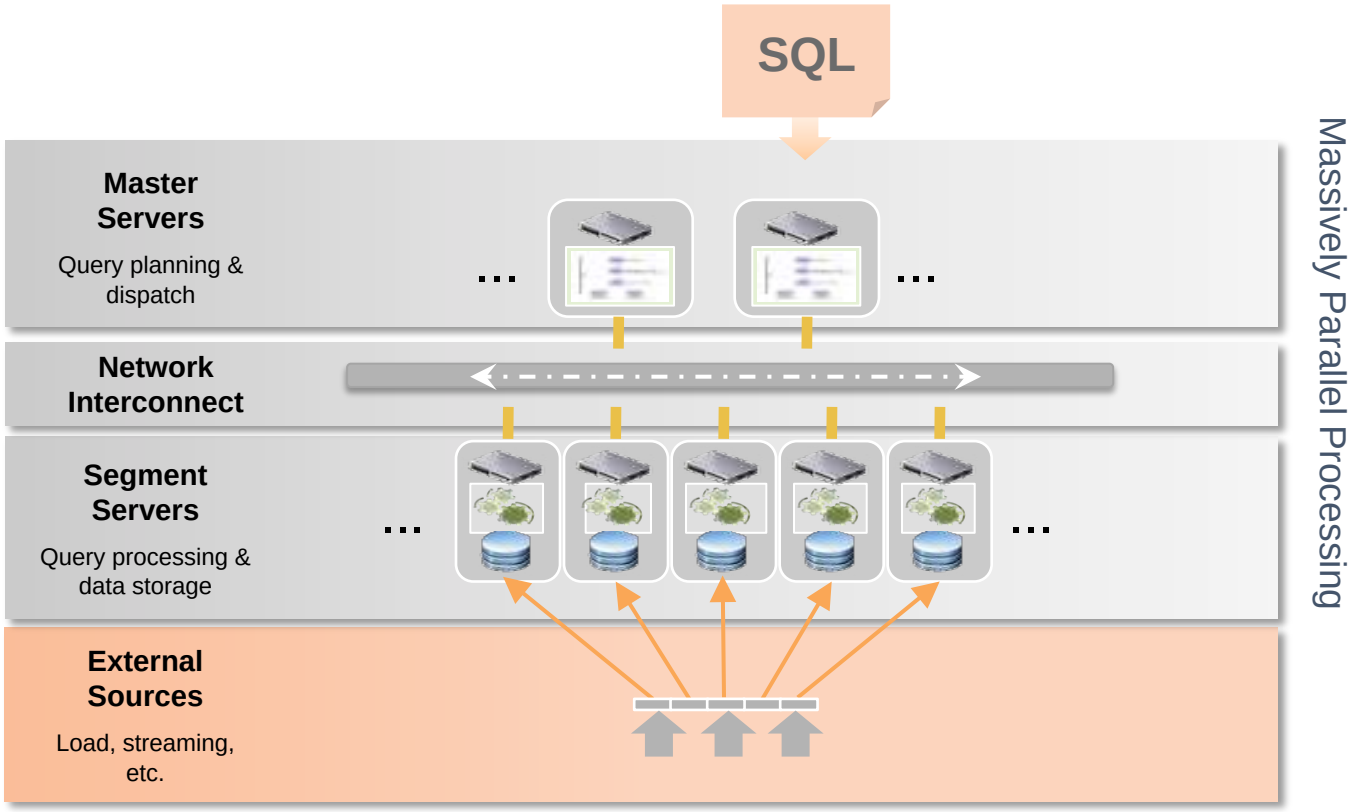
MADlib 特性

- 更好的并行度
  - 算法充分利用 MPP 架构实现并行
- 更好的可扩展性
  - 算法随着数据扩充而线性扩展
- 更高的预测精准度
  - 适用更多数据，而不是抽样
- 顶级 ASF 开源项目
  - 社区驱动开发模式

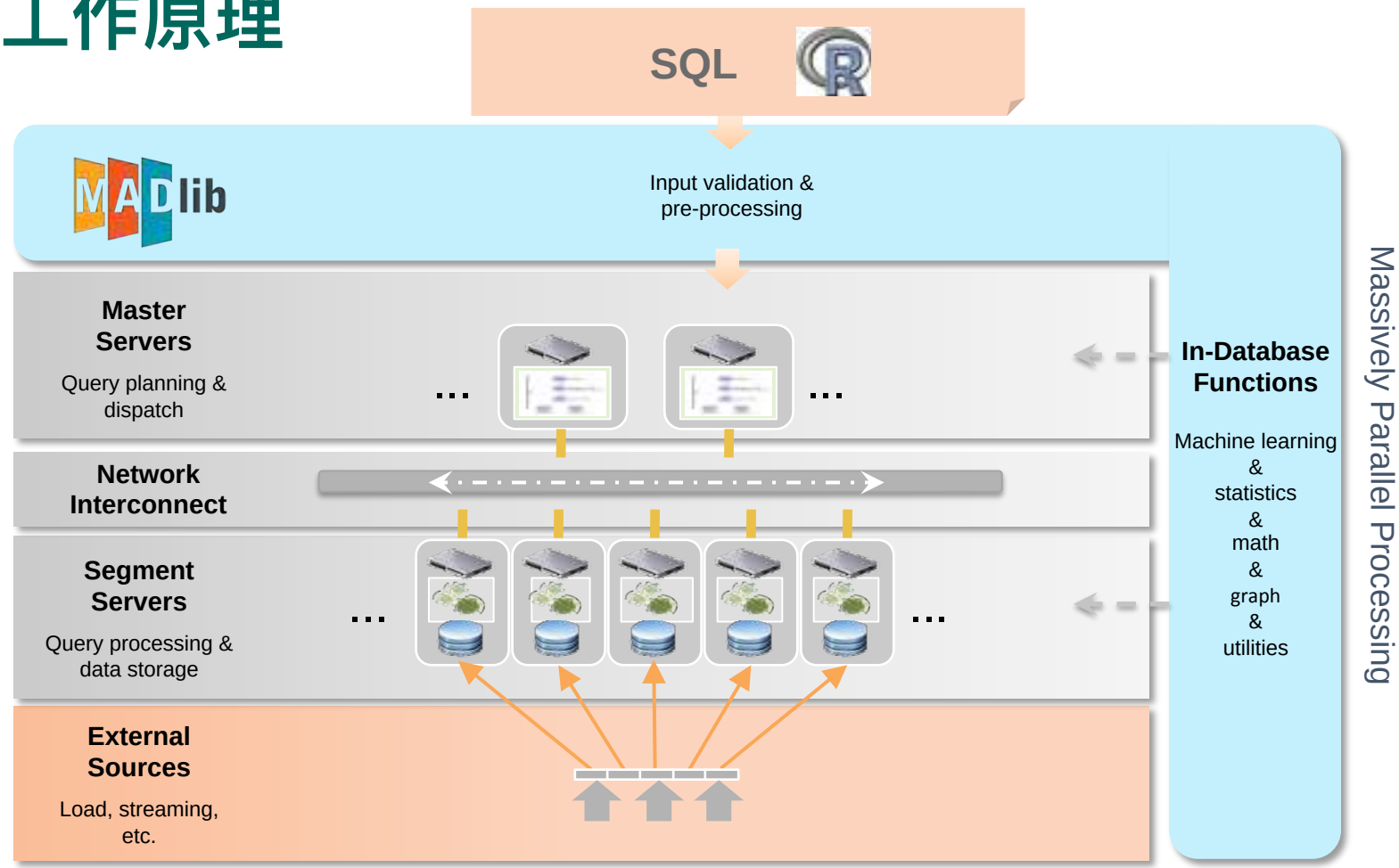
执行流程


客户端
SQL
String
数据库服务器
存储过程
结果集
Master
聚集
Segment 1
Segment 2
Segment n
jupyter
Apache Zeppelin
pgAdmin
psql

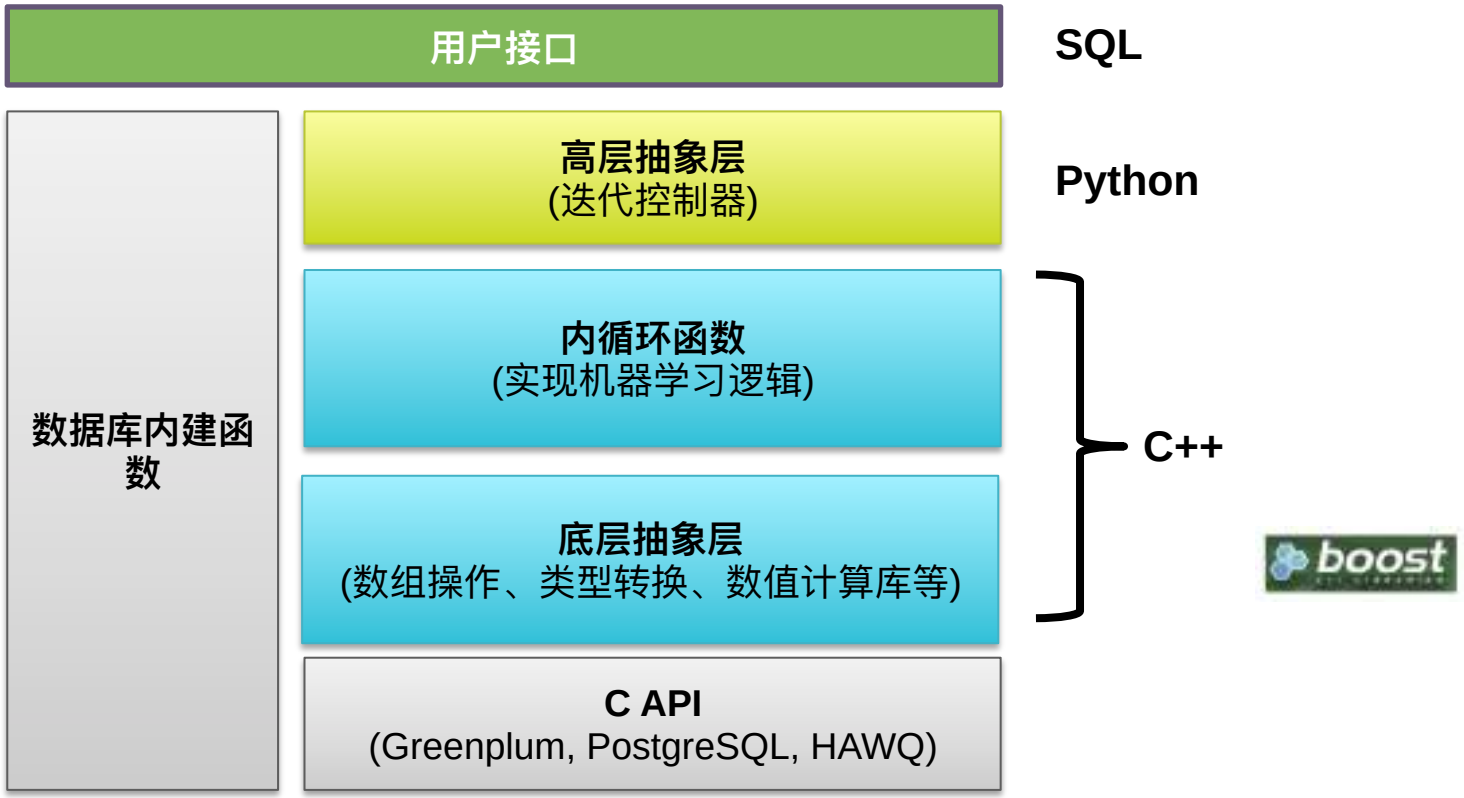
工作原理


SQL
Master Servers
Query planning & dispatch
Network Interconnect
Segment Servers
Query processing & data storage
External Sources
Load, streaming, etc.
Massively Parallel Processing

工作原理


SQL
MADlib
Input validation &
pre-processing
Master
Servers
Query planning &
dispatch
Network
Interconnect
Segment
Servers
Query processing &
data storage
External
Sources
Load, streaming,
etc.
In-Database
Functions
Machine learning
&
statistics
&
math
&
graph
&
utilities
Massively Parallel Processing

MADlib 架构


用户接口
SQL
高层抽象层
(迭代控制器)
Python
内循环函数
(实现机器学习逻辑)
C++
数据库内建函数
底层抽象层
(数组操作、类型转换、数值计算库等)
boost
C API
(Greenplum, PostgreSQL, HAWQ)

示例 - PageRank

- 是一种由搜索引擎根据网页之间相互的超链接计算的技术，而作为网页排名的要素之一，以Google创办人 Larry Page来命名

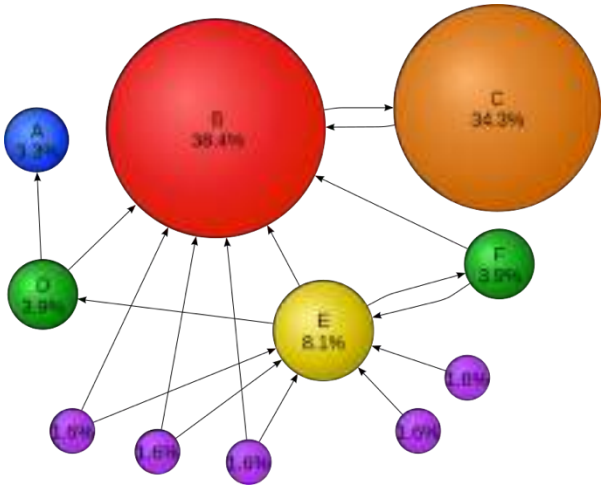


Image from
https://en.wikipedia.org/wiki/PageRank

示例 - PageRank

数据

```
CREATE TABLE vertex(
        id INTEGER
        );
CREATE TABLE edge(
        src INTEGER,
        dest INTEGER,
        user_id INTEGER
        );
```

计算

```
SELECT madlib.pagerank(
                        'vertex',             -- Vertex table
                        'id',                 -- Vertix id column
                        'edge',               -- Edge table
                        'src=src, dest=dest', -- Comma delimted string of edge arguments
                        'pagerank_out');      -- Output table of PageRank
```

示例 - PageRank

计算结果

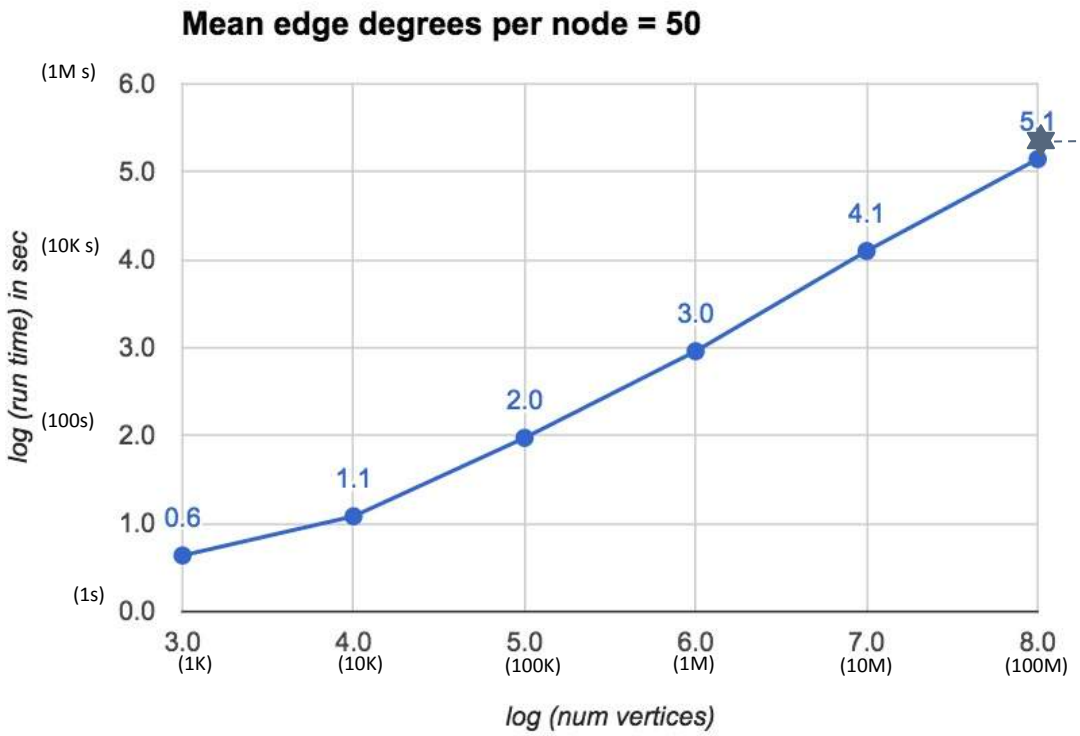
```
SELECT * FROM pagerank_out ORDER BY pagerank DESC;
```

```
 id |      pagerank
----+--------------------
  0 |  0.28753749341184
  3 |  0.21016988901855
  2 |  0.14662683454062
  4 |  0.10289614384217
  1 |  0.10289614384217
  6 |  0.09728637768887
  5 |  0.05258711765692
(7 rows)
```

```
SELECT * FROM pagerank_out_summary;
```

```
 __iterations__
 ---------------+
             16
(1 row)
```

可扩展性 – PageRank 性能



5B edges

每个定点有50个边的图

Greenplum 集群规模:

- 1 主
- 4 数据节点
- 每个节点6个segment

Note: log-log scale

MADlib vs. Spark: 不同的产品，侧重点不同

	MADlib	Spark
算法库		
易用性		需要编程
查询优化		成熟度稍差
内存和流处理	通过 Gemfire	
SQL 语法支持		需要提升
磁盘数据		不是核心焦点
并发性能		不是核心焦点
大数据关联		不是核心焦点

用户案例 1
Greenplum + MADlib 助力邮件营销

背景

客户

- 某大型跨国多元化传媒和娱乐公司

问题

- 邮件广告点击预测模型不够精准，需要更好的邮件营销策略
- 现有数据分析流程繁琐，速度慢，有很多手动步骤，易出错

数据科学解决方案

- 简化Data 流程
- 在Madlib上重新建模和预测
- 实现流程全自动化

数据和技术预览

数据源

- 客户数据
  - 购买
  - 预定
  - 营销
  - 在线注册
  - 网页浏览历史
  - 地理信息数据
  - 业务部门信息
  - 网站用户信息
- TB 级别数据
- 1000+ 特征

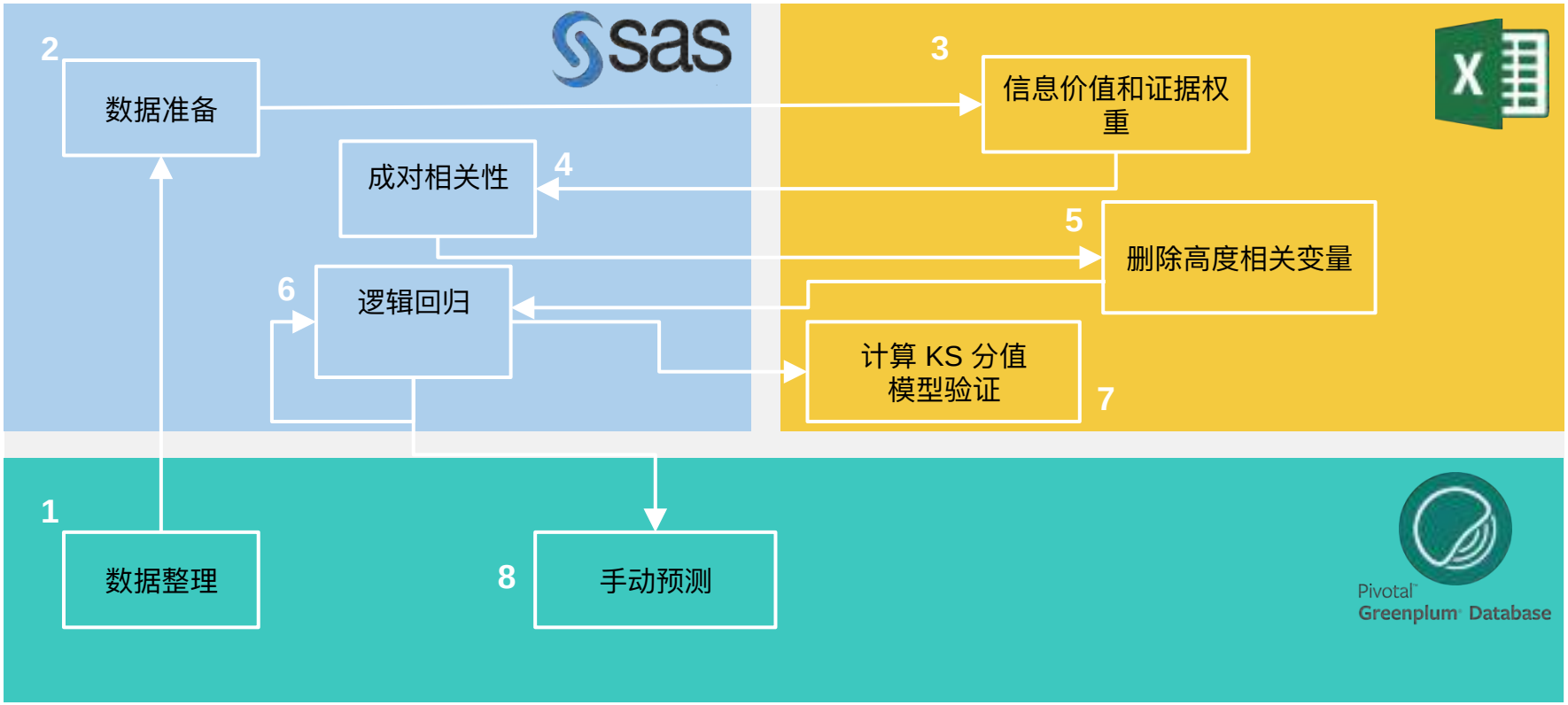
平台

Pivotal™
Greenplum® Database

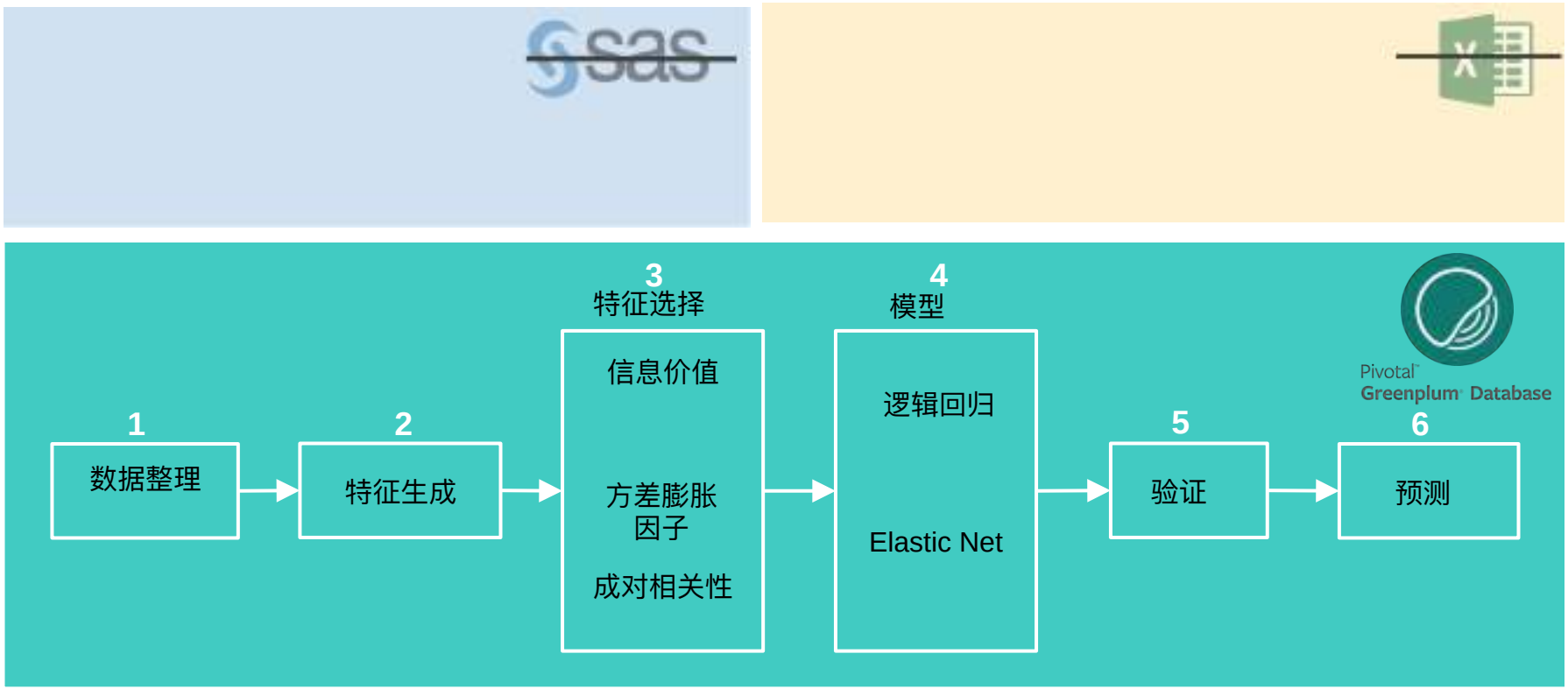
建模工具

PL/pgSQL

原始工作流程


sas
2
数据准备
3
信息价值和证据权重
成对相关性
4
5
删除高度相关变量
6
逻辑回归
计算 KS 分值
模型验证
7
1
数据整理
8
手动预测
Pivotal
Greenplum Database

改进后的 in-database 流程



工作流程优化

	之前	之后	性能提升
数据编辑/整理	• 181 行代码 • 75 分钟	• 116 行代码 • 8 分钟	9.35x
特征编辑	• 439 特征 • 4,517 行代码 • 100 分钟	• 934 特征 • 1,438 行代码 • 30 分钟	多 495 个特征，快 3.33x
信息价值	• ~450 个变量，~30分钟计算结果并写入 excel	• 在 GPDB 中花 58 秒计算 ~200 个变量的IV	13.7x/变量
建模	• < 50 个变量，运行一次逻辑回归迭代需要 ~30 分钟	• 376 个变量，运行一次逻辑回归迭代需要 ~1.86 分钟	~16x/迭代

建模结果

原始模型

- 模型精确度 = 99.7%
- 真正率(True Positive Rate) = 0%

该模型善于预测不会点击邮件的用户，但是无法预测会点击邮件的用户

改良后的模型

- 模型精确度= 62.8%
- 真正率 = 66%

该模型更善于预测会点击邮件的用户，这样是用户真正关心的，能为公司带来价值的用户群体

商业影响

改良前	改良后
X 对数据集的探索有限	✓ 在Greenplum里充分探索了数据集
X 对Pivotal产品线不熟悉	✓ 在Greenplum上充分利用了MADlib和PL/X
X 在SAS和Excel上有很多手动流程	✓ 在Greenplum内部实现了流程自动化
X 代码复杂冗余，很多数据类型转换	✓ 代码更精简，更便于维护的代码
X 原始模型预测效果不理想	✓ 新模型能够更精准地预测目标客户

Pivotal

用户案例2
基于API日志的金融产品用户分析

背景

客户

- 某大型跨国金融服务公司
- 移动应用 API 分析

问题

- 更好地理解不同种类的用户
- 更好地了解用户与APP的交互
- 对实时API请求进行分类和安全检测
- 数据量大，现有数据分析团队缺乏技能

数据科学解决方案

- 使用Madlib进行聚类分析，建立会话识别模型和主题模型
- 建立scoring pipeline, 对新访问的安全性进行评估
- 使用可视化工具对结果进行更好地呈现

数据和技术概览

数据源

- 数据
  - API 访问日志
  - 客户数据
- 45 天区域数据
- 50亿条数据
- 上百万订购者

平台

Pivotal Greenplum Database

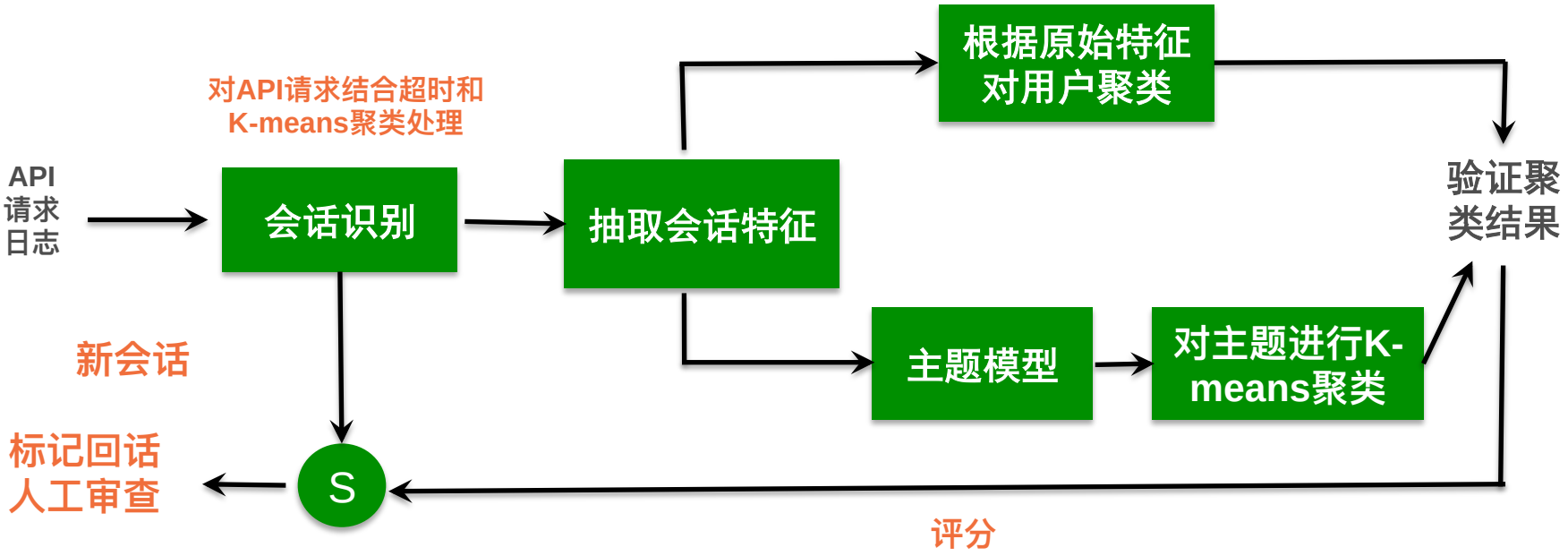
建模工具

MADlib PL/R, PL/PYTHON, PDLTools

可视化

tableau

建模过程

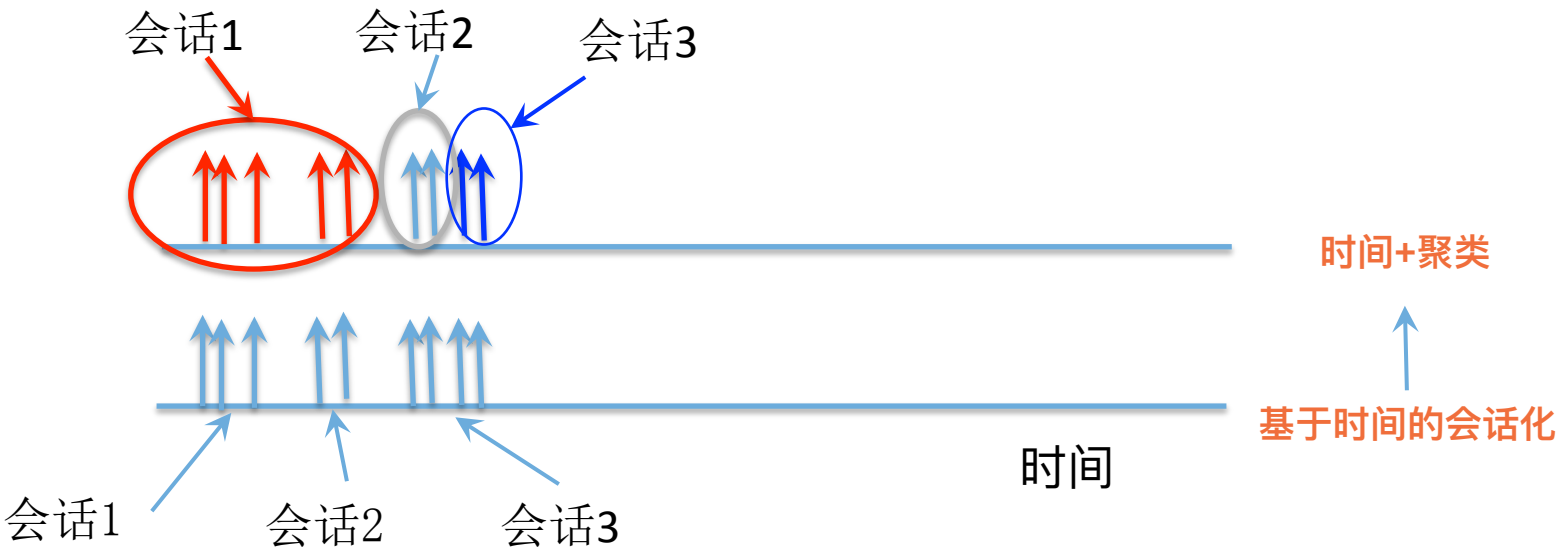

对API请求结合超时和
K-means聚类处理
API
请求
日志
会话识别
抽取会话特征
根据原始特征
对用户聚类
验证聚
类结果
新会话
主题模型
对主题进行K-
means聚类
标记回话
人工审查
S
评分

建模过程

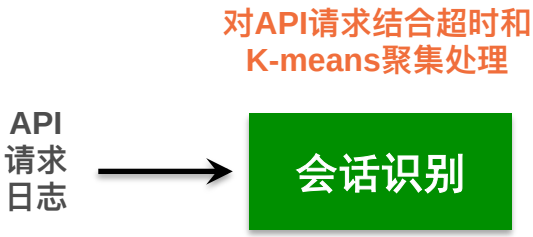
对API请求结合超时和
K-means聚类处理

API
请求
日志

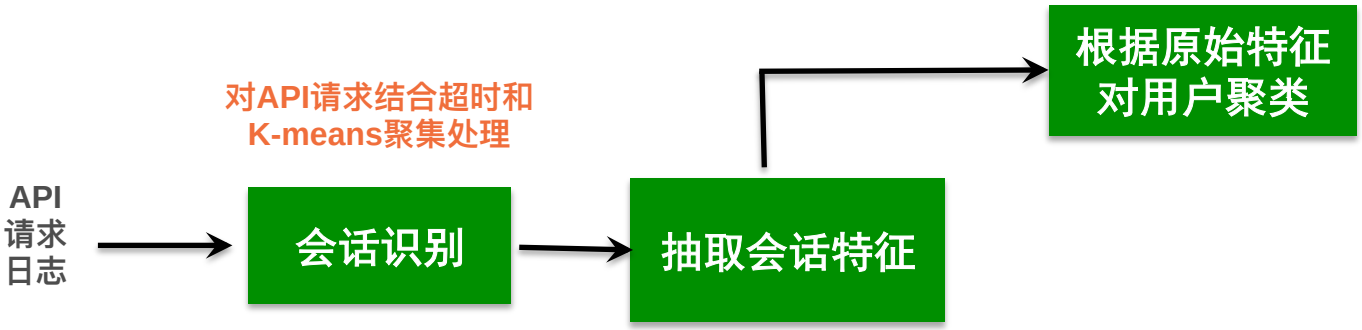
对API请求进行会话化



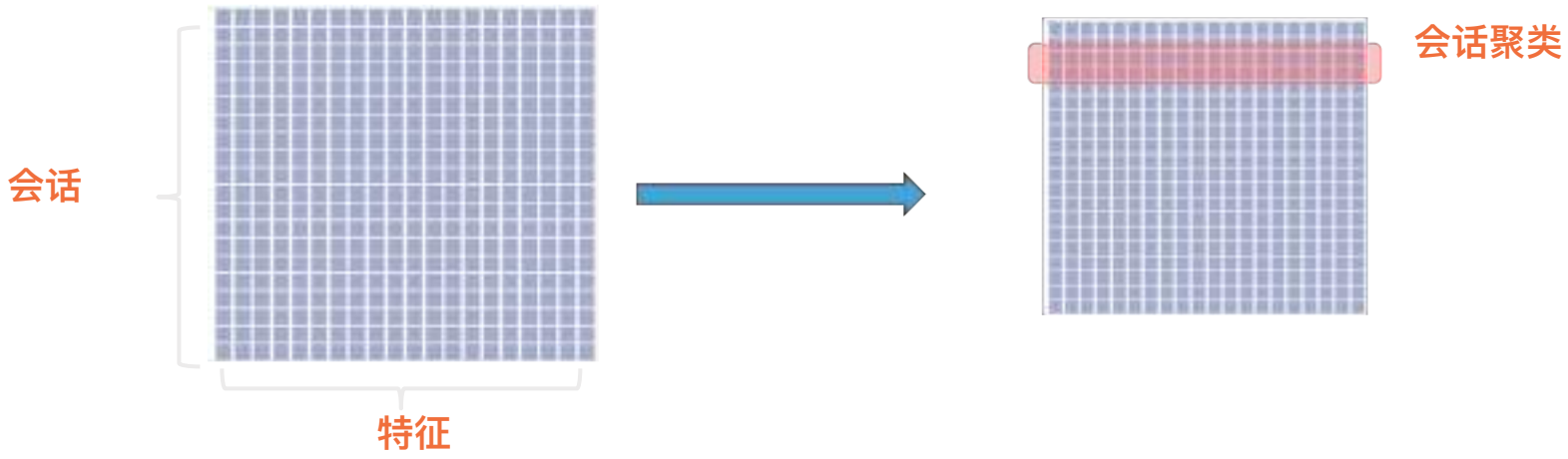
建模过程



建模过程



K-means 聚类示例

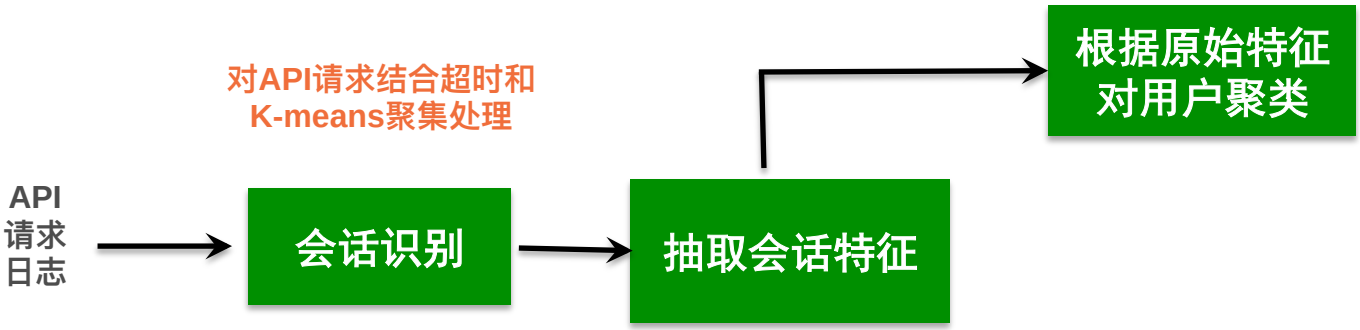



```
kmeans_random( rel_source,
               expr_point,
               k,
               fn_dist,
               agg_centroid,
               max_num_iterations,
               min_frac_reassigned
             )
```

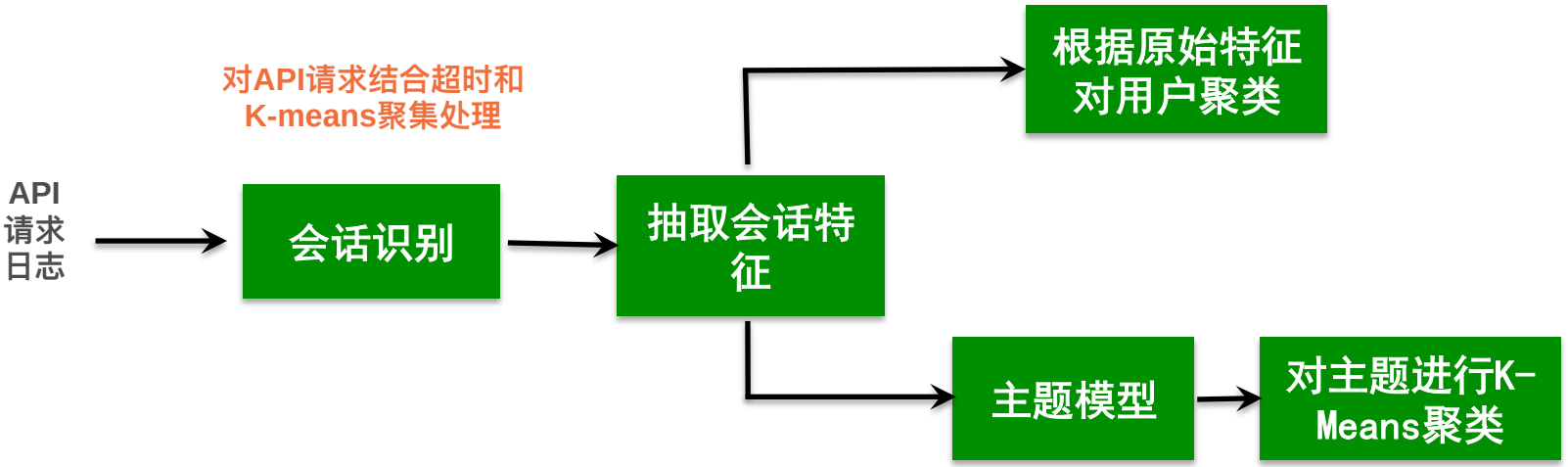
MADlib

```
kmeanspp( rel_source,
          expr_point,
          k,
          fn_dist,
          agg_centroid,
          max_num_iterations,
          min_frac_reassigned
        )
```

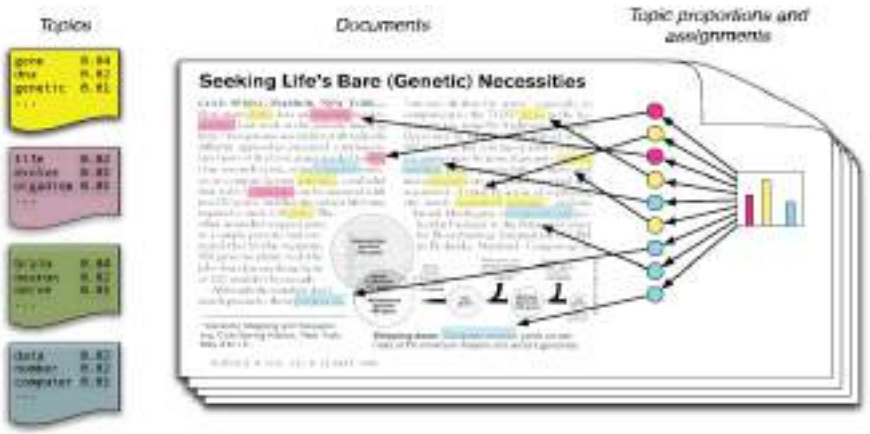
建模过程



建模过程


对API请求结合超时和
K-means聚集处理
API
请求
日志
会话识别
抽取会话特征
根据原始特征
对用户聚类
主题模型
对主题进行K-
Means聚类

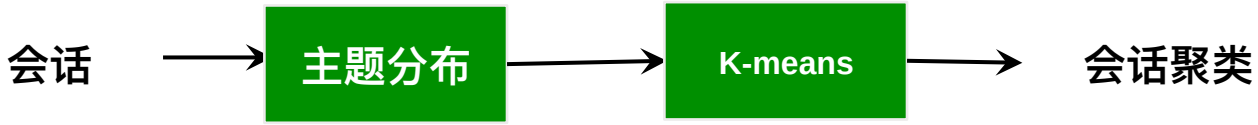
主题模型：Latent Dirichlet Allocation (LDA)



- 一篇文章中反复出现在一起的几个词语往往在描述同一个主题
- 一篇文章往往含有多个主题，且每个主题所占的比例各不相同
- LDA 自动分析每个文档，统计文档内的词语，根据统计的信息来断定当前文档含有哪些主题，以及每个主题所占的比例各为多少。

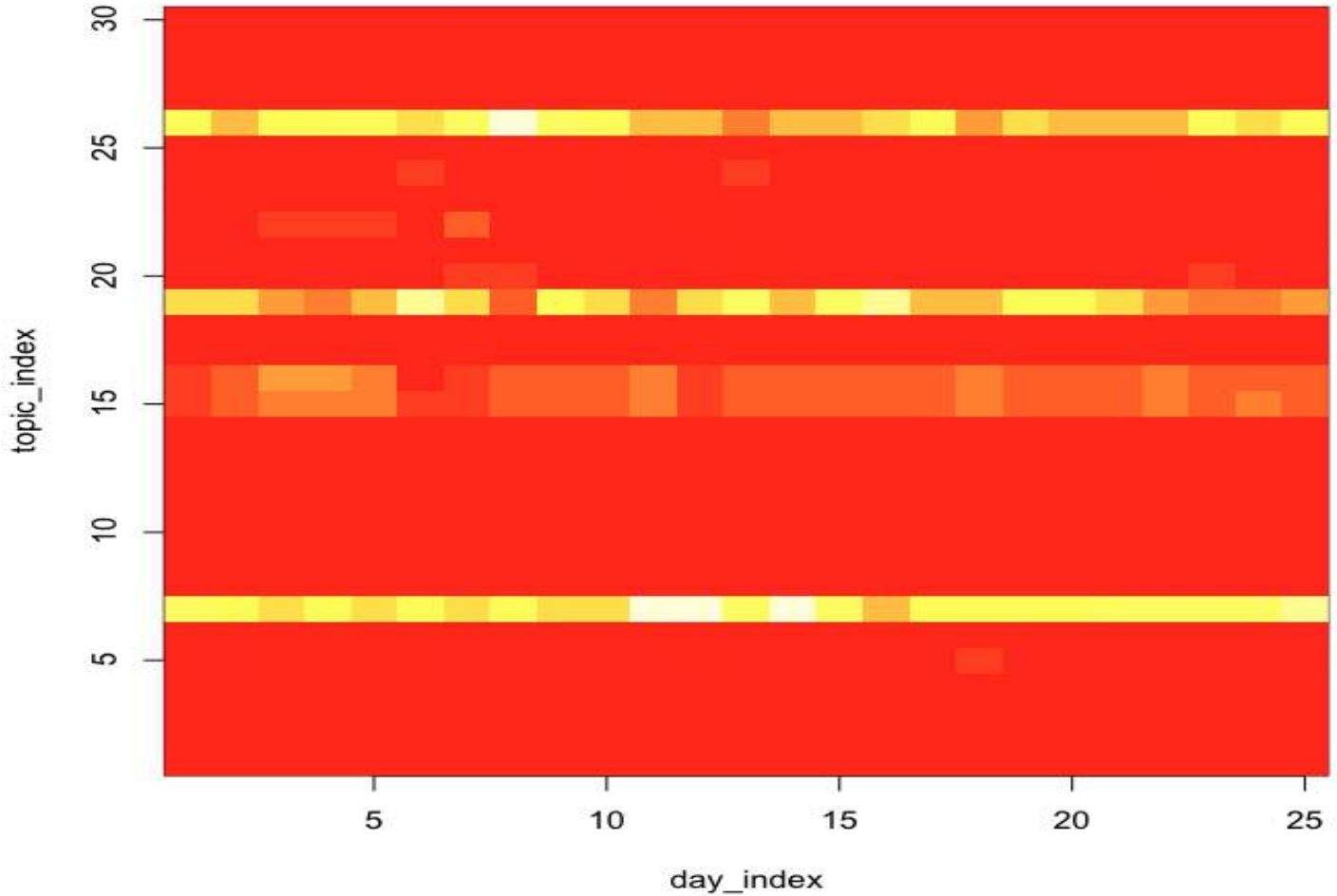
基于LDA特征的聚类


```
lda_train( data_table,
           model_table,
           output_data_table,
           voc_size,
           topic_num,
           iter_num,
           alpha,
           beta
         )
```

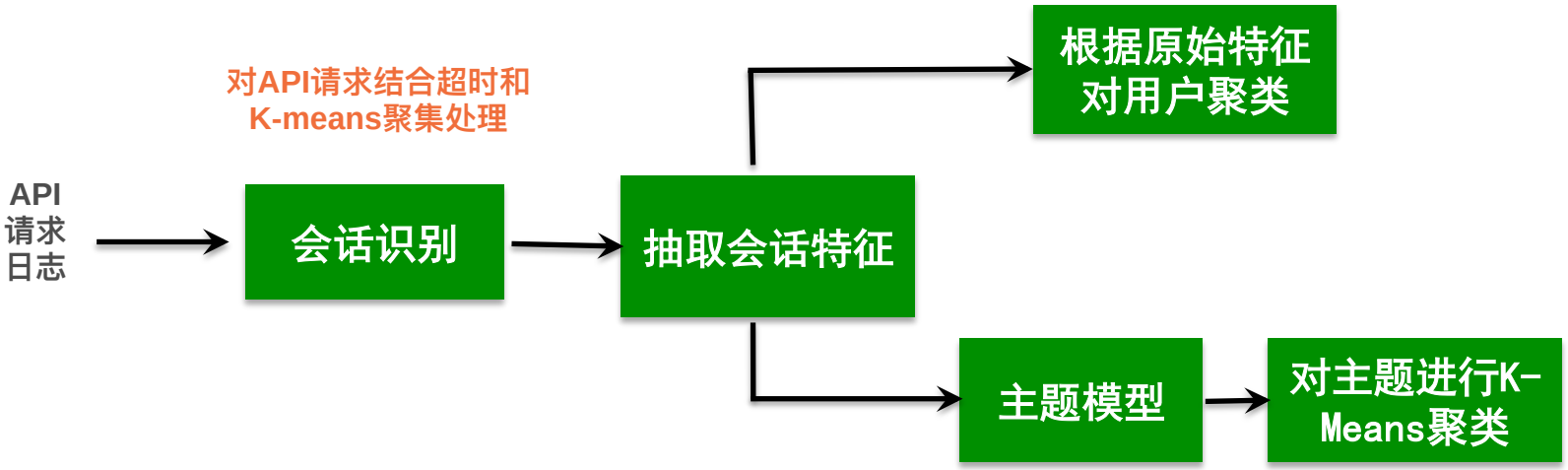


- 把每个会话看成一篇文章
- 百万级别文档
- 词汇量 8000+

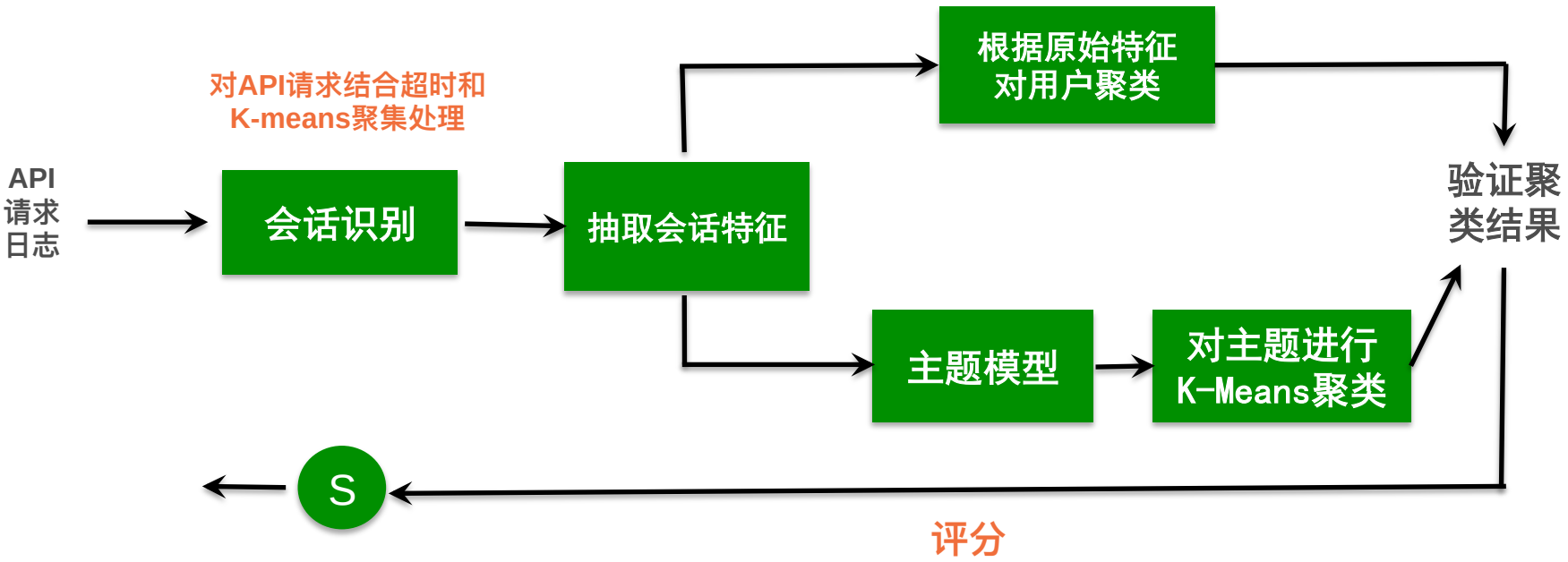
某个特定用户在一个月内都关注什么?



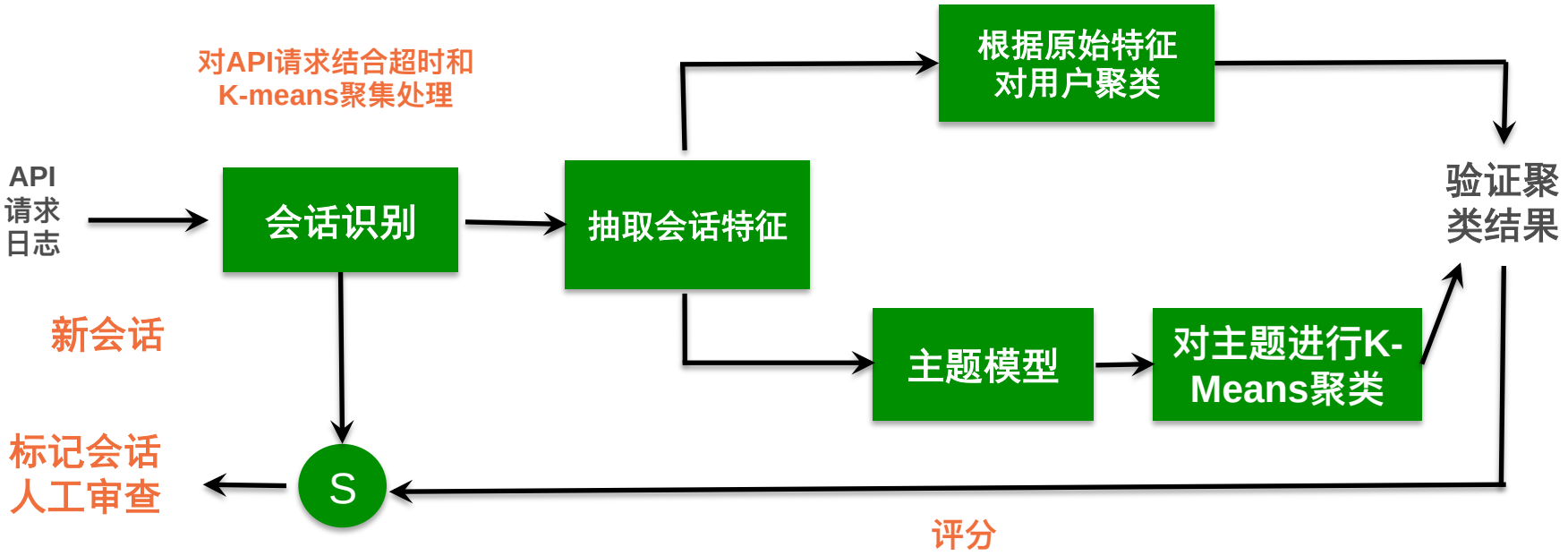
建模过程


对API请求结合超时和
K-means聚集处理
API
请求
日志
会话识别
抽取会话特征
根据原始特征
对用户聚类
主题模型
对主题进行K-
Means聚类

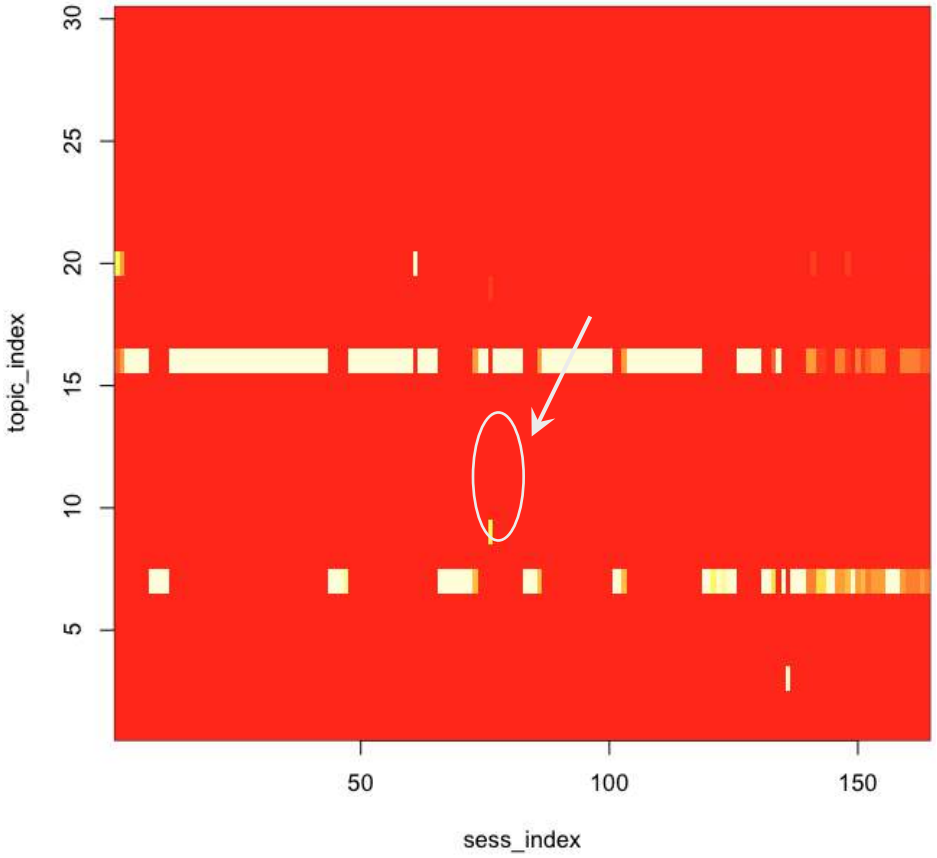
建模过程

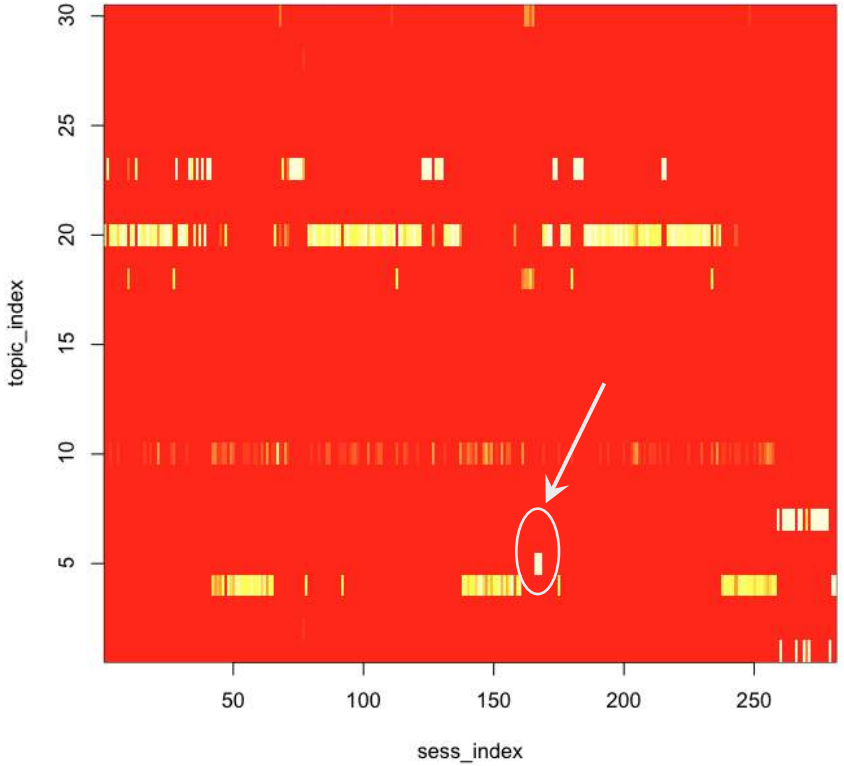

对API请求结合超时和
K-means聚集处理
API
请求
日志
会话识别
抽取会话特征
根据原始特征
对用户聚类
主题模型
对主题进行
K-Means聚类
验证聚
类结果
S
评分

建模过程

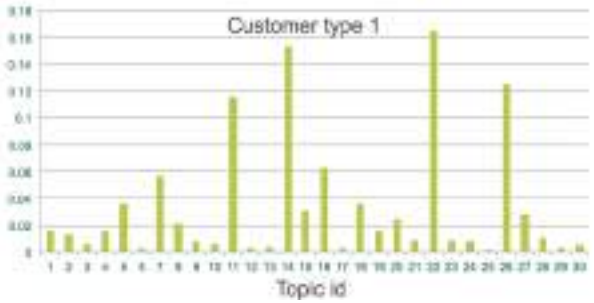

对API请求结合超时和
K-means聚集处理
根据原始特征
对用户聚类
API
请求
日志
会话识别
抽取会话特征
验证聚
类结果
新会话
主题模型
对主题进行K-
Means聚类
标记会话
人工审查
S
评分

主题分布热力图检测异常



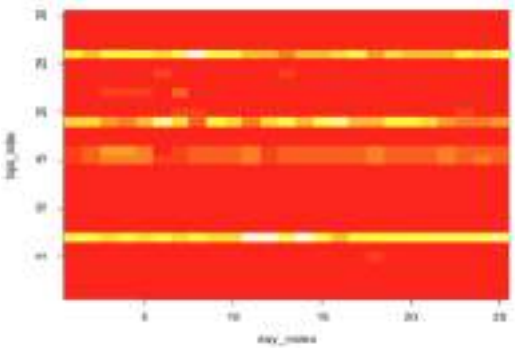


建模效果示例



股票分析师

固定收益分析师



客户行为热图

案例优化总结

改良前	改良后
X 在R上对data sample进行分析，DCA闲置	✓ 使用Greenplum+MADlib对大数据集进行了更充分的分析
X 没有良好的用户分类体系	✓ 建立了两套模型对典型用户进行聚类分析，对用户群体和用户习惯有了更深入的了解，制定相应的营销策略
X 不能高效监测可疑Session	✓ 建立了可疑Session实时评分体系
X 考虑转换到Teradata	✓ 决定增加Greenplum Cluster数量

Pivotal Greenplum: 开源大数据高级分析平台