



Engenharia de Dados e Engenharia de Machine Learning

como fazer projetos de Data Science usando um
processo sustentável

Fabiane Bizinella Nardon
Chief Data Scientist
@fabianenardon

Harvard Business Review

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

SUMMARY SAVE SHARE COMMENT TEXT SIZE PRINT \$9.95 BUY COPIES

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join.

<https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>

Forbes Science

OS DOIS RITMOS DA LAVA-JATO
Ourtiles: 107 condenados
Brasília: nenhum

veja

DE MÃOS DADAS COM A INTELIGÊNCIA ARTIFICIAL

Longe dos cenários futurísticos da ficção científica, ela já faz parte do presente. Mas em que medida pode servir ao ser humano e, ao mesmo tempo, ameaçá-lo?

EXCLUSIVO Artigo de Yuval Noah Harari, autor de *Homo Deus: uma Breve História do Amanhã*

What it takes to end an ERD epidemic
Diala brain cancer through brain networks
Spine surgery in a new way

INSIDE: A 14-PAGE SPECIAL REPORT ON FINANCIAL TECHNOLOGY

The Economist

How to fix America's inner cities
The self-service economy
Time to open up Indonesia
Inside the anti-bribery business
Why humans cause heartbreaks

Artificial Intelligence

The promise and the peril

MAR 9TH - 15TH 2015

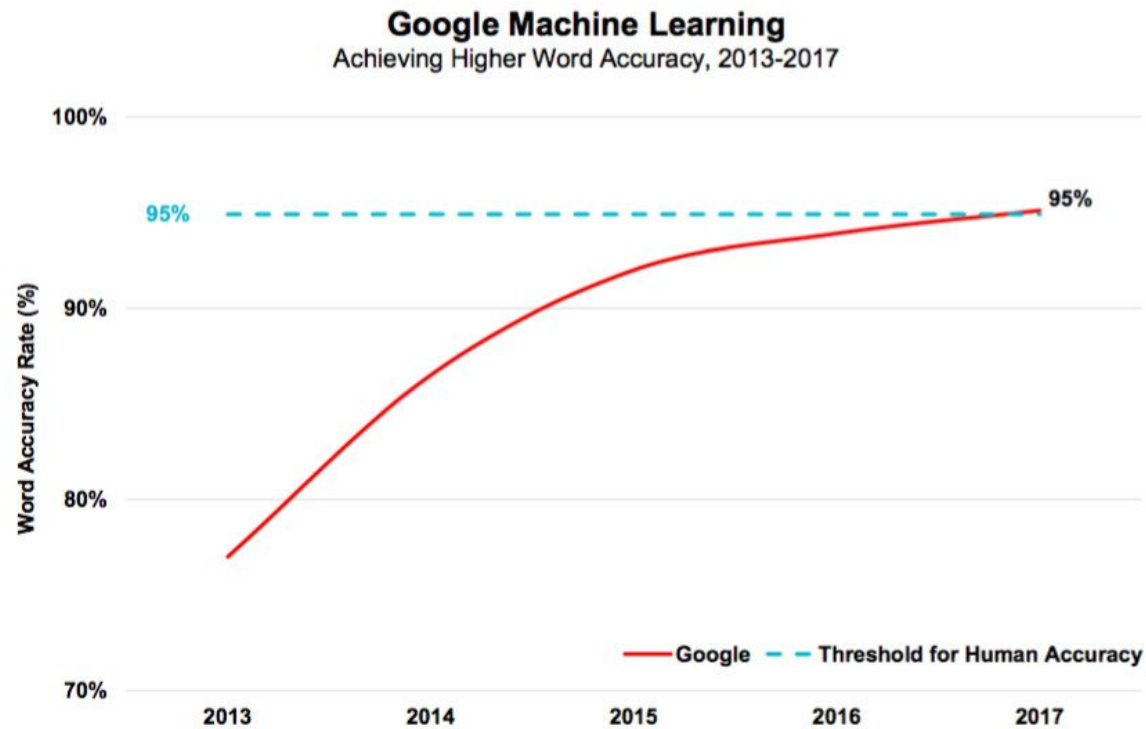
Worldwide ex UK

THE NEW ARTIFICIAL INTELLIGENCE

THE FUTURE OF NANOBOTS
DETECTING CANCER
KILLER ROBOTS?

THE NEW SPECIAL

...Voice-Based Platform *Back-Ends* = Voice Recognition Accuracy Continues to Improve



**KLEINER
PERKINS**

Source: Google (5/17)
Note: Data as of 5/17/17 and refers to recognition accuracy for English language. Word error rate is evaluated using real world search data which is extremely diverse and more error prone than typical human dialogue.



ARTIFICIAL INTELLIGENCE · BIG DATA · DATA SCIENCE · FEATURES

WHY 96% OF ENTERPRISES FACE AI TRAINING ISSUES

DON ROEDNER · JULY 30, 2019

A recent survey of over 225 enterprise Data Scientists, AI technologists, and machine learning (ML) projects, suggests that AI is still in its early days for AI technology.

The AI market is projected to become a \$190 billion industry by 2025 (according to Markets and Markets), and global spending on cognitive and AI systems is projected to reach \$150 billion in 2029, an increase of 44.0% over the amount spent in 2025. Research suggests AI is advanced and on the move, already being adopted by enterprises and ready to make an impact on how we live and work.

bmc blogs

AI Ops | BMC Beat | Cloud | DevOps | Experience | ITSM | Mainframe | Workload Automation

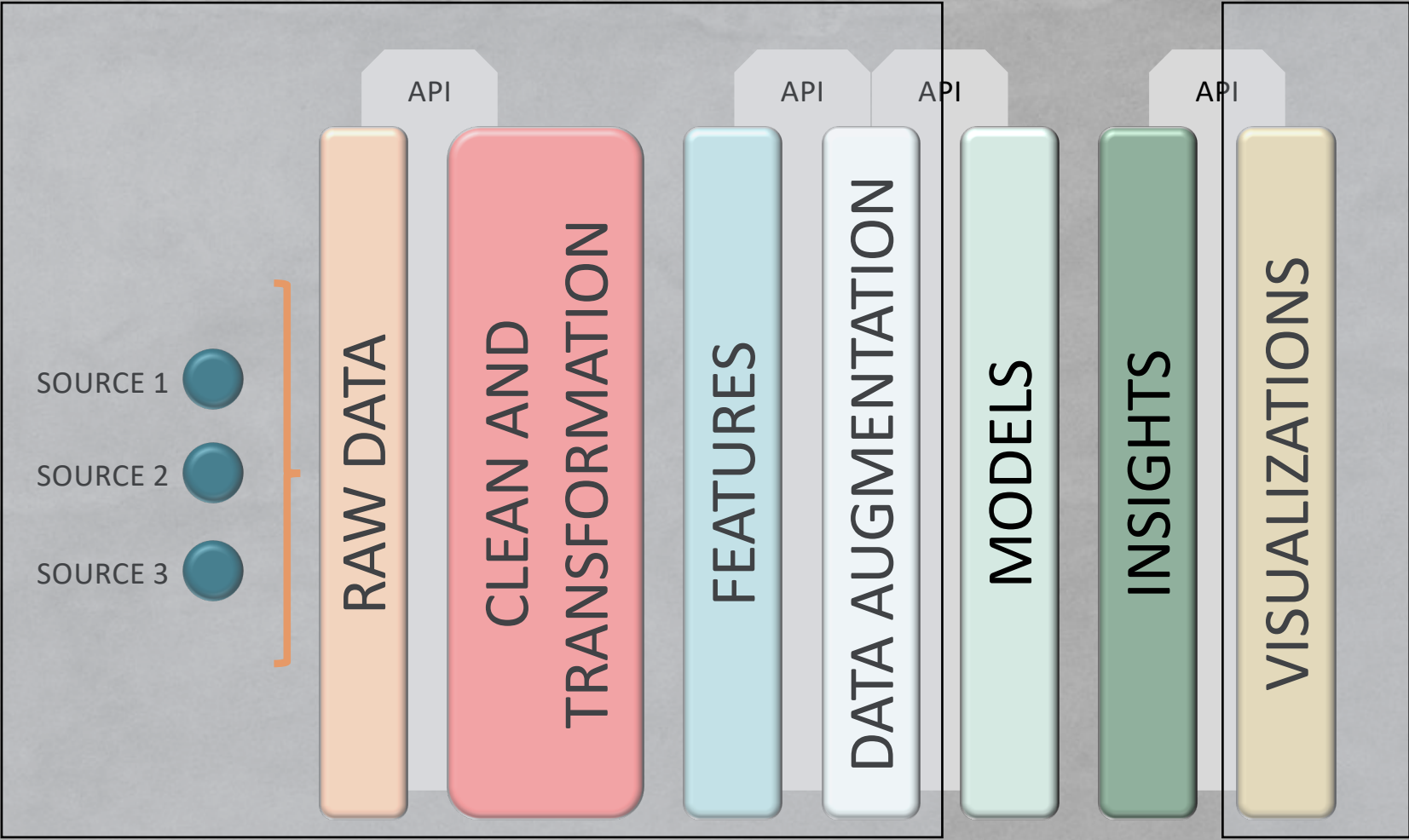
Machine Learning & Big Data Blog

Why does Gartner predict up to 85% of AI projects will “not deliver” for CIOs?

Once companies opt for an AI project, confusion can wreak havoc. A general lack of understanding around all things AI means you may not have enough data, or that data may not be suitable to the project you're considering. If your data isn't good, your algorithms can't be tested correctly – maybe you're using the wrong algorithms for what you're trying to solve. Any amount of understanding can fuel poor team management, but the more misunderstanding, the more likely the team is just wasting time.

10 CHALLENGES THAT DATA SCIENTISTS WILL FACE

Data Science Pipeline



90% do tempo de um projeto de
Data Science
é gasto em Engenharia de Dados

<https://www.kdnuggets.com/2019/03/most-impactful-ai-trends-2018-rise-ml-engineering.html>

machine learning



161,300 repository results

Sort: Best match ▾

data engineering



3,325 repository results

Sort: Best match ▾

feature engineering



1,010 repository results

Sort: Best match ▾

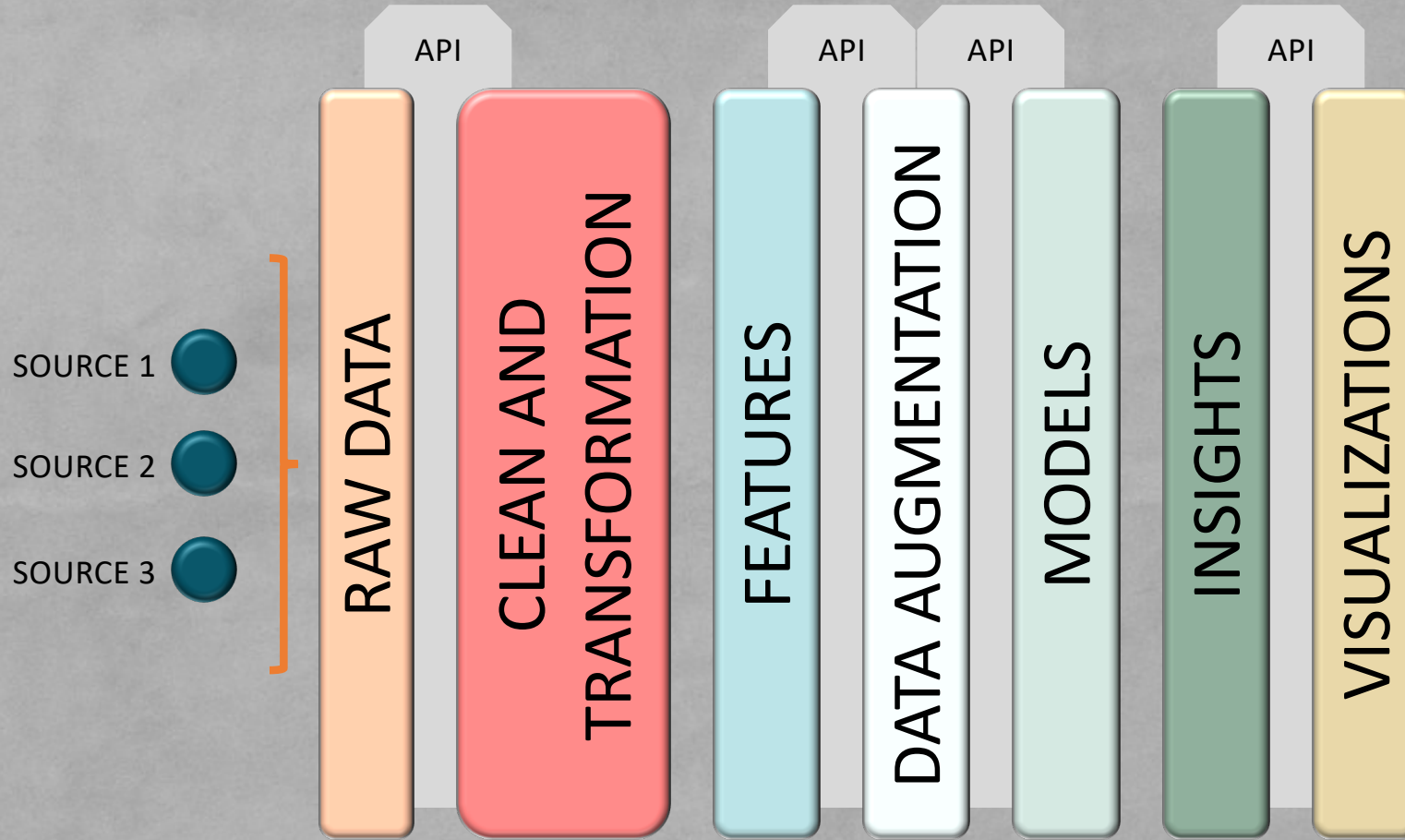
data lake



731 repository results

Sort: Best match ▾

Data Science Pipeline



ESCALA

3.5 bilhões de novos registros
4.680 pipelines executados

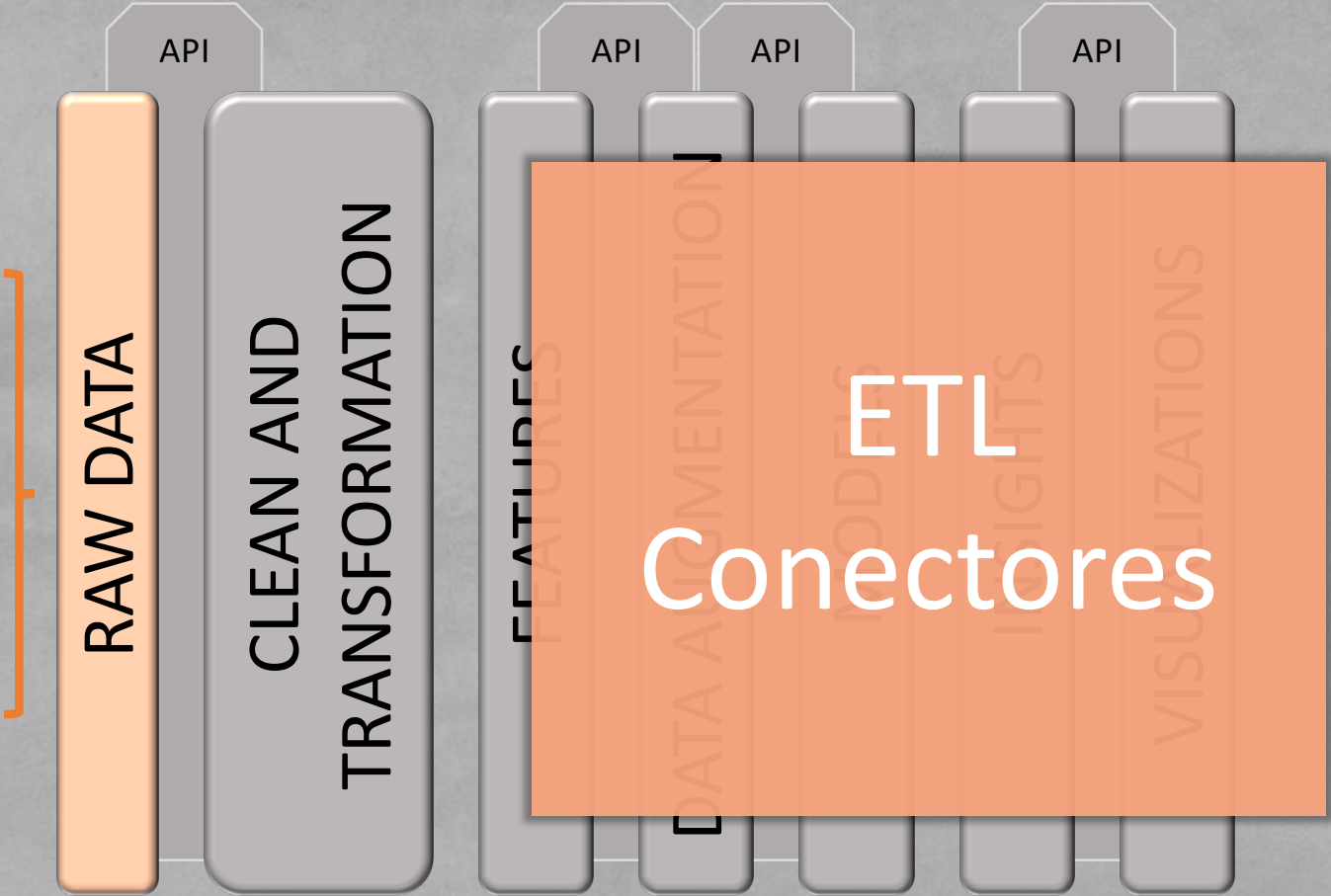
em um dia

Teoria do Mundo Perfeito

1. Não existe Big Data nem código legado
2. Depois que você faz a experimentação, o trabalho acabou
3. Existe um Data Lake com todos os dados que você precisa

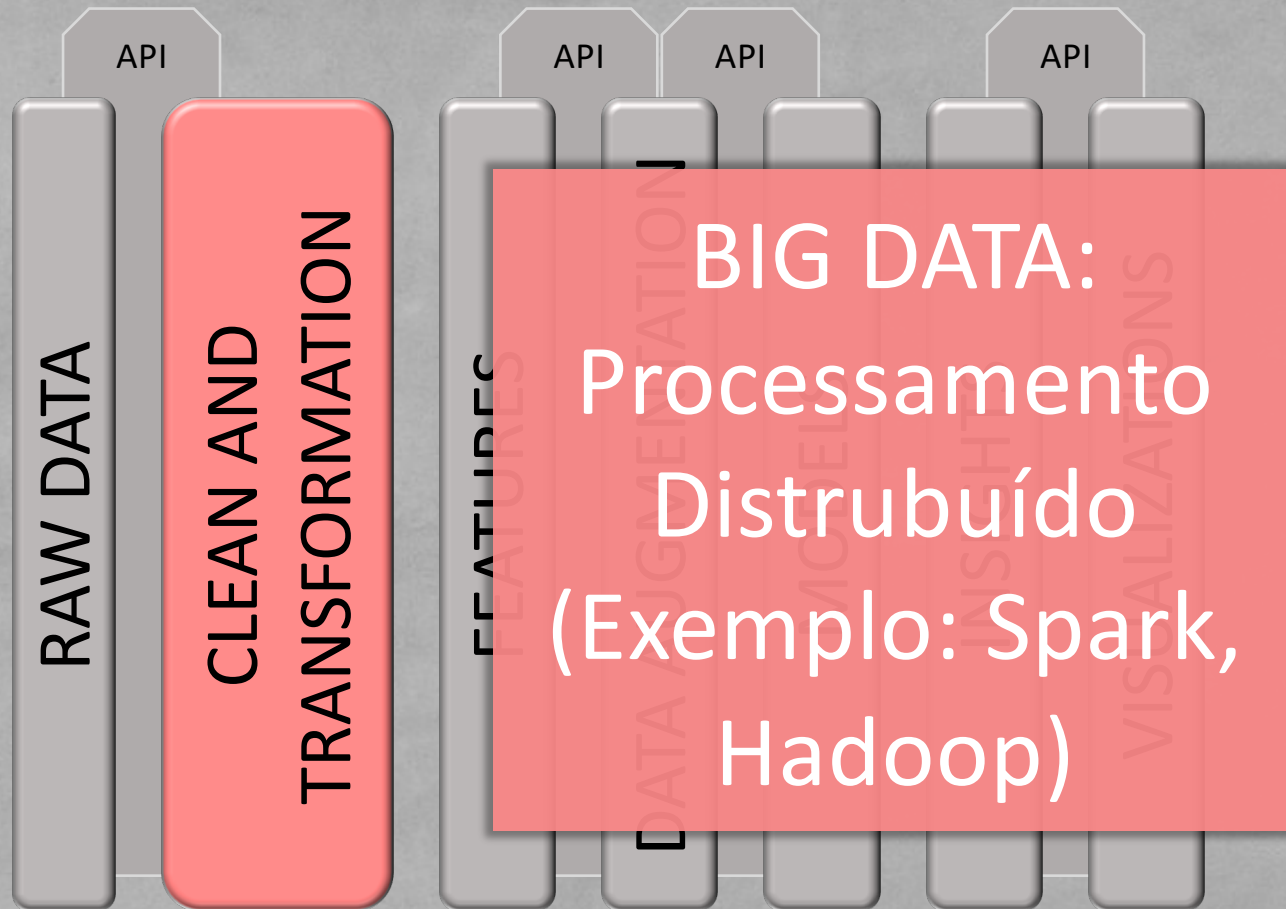
Data Science Pipeline

- SOURCE 1 ●
- SOURCE 2 ●
- SOURCE 3 ●



Data Science Pipeline

- SOURCE 1
- SOURCE 2
- SOURCE 3



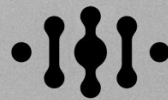
> Spark SQL

```
val records = spark.read.option("header", "true")
    .csv("example-database.csv").toDF

records.createOrReplaceTempView("records")

val transformed = spark.sql(
    "SELECT ucase(name) as name from records"
)

transformed.write
    .format("com.databricks.spark.csv")
    .option("header", "true")
    .save("newrecords.csv");
```



Scheduler

Spark Cluster

SOURCE 1



SOURCE 2



SOURCE 3



```
Exploration of geo data

Create the Spark Context:
import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder().appName("Spark SQL basic example")
    .config("spark.driver.memory", "1024m")
    .config("spark.recoincubation.address", "localhost").getOrCreate()

// More configuration is used for the job on cluster. If running on the cloud or other env, you can change to:
// val spark = SparkSession.builder().appName("Spark SQL basic example").getOrCreate()

import org.apache.spark.sql.SparkSession
spark = org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession.builder()

Read a local data file with geo data and create a Data Frame:
val df = spark.read.option("header", "true")
    .option("inferSchema", "true").load("src/main/resources/geo-data.csv")
df: org.apache.spark.sql.DataFrame = [lat: string, longitude: string ... 5 more fields]

Show the data retrieved from the file:
df.show
```

Pipeline



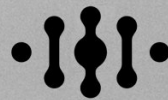
Data Lake

```
> code  
> lake
```



> Spark SQL

```
val records =  
    spark.read.option("header", "true")  
              .csv("example-database.csv").toDF  
  
val filtered =  
    records.filter("type = 'android'")  
  
filtered.write  
    .format("com.databricks.spark.csv")  
    .option("header", "true")  
    .save("newrecords.csv");
```

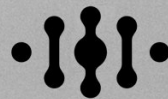


> Spark SQL + Meu Código

```
val geoHash = new my.udf.SparkGeoHash();

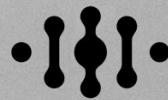
spark.udf.register("geoHash", geoHash,
                    DataTypes.StringType);

val recordsWithFunction =
    filtered.withColumn("generatedGeoHash",
                        callUDF("geoHash", col("latitude"),
                                col("longitude"), lit(12)))
    .select(col("id"), col("latitude"),
            col("longitude"),
            col("geo_hash"),
            col("generatedGeoHash"))
```



> **Plugins**

- > Tipos de dados com semântica
- > Transformações
- > Agregações



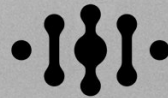
> Tipos de Dados com Semântica

```
val records = spark.read.option("header","true")  
    .csv("example-database.csv").toDF
```

```
records.printSchema
```

```
root
```

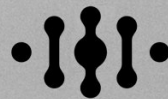
```
 |-- id: string (nullable = true)  
 |-- timestamp: string (nullable = true)  
 |-- type: string (nullable = true)  
 |-- latitude: string (nullable = true)  
 |-- longitude: string (nullable = true)  
 |-- horizontal_accuracy: string (nullable = true)  
 |-- geo_hash: string (nullable = true)
```



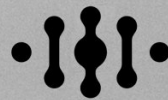
> Tipos de Dados com Semântica

```
val schema = StructType(Array(  
  StructField("ad_id", StringType, true),  
  StructField("utc_timestamp", LongType, true),  
  StructField("id_type", StringType, true),  
  StructField("latitude", DoubleType, true),  
  StructField("longitude", DoubleType, true),  
  StructField("horizontal_accuracy", DoubleType, true),  
  StructField("geo_hash", StringType, true)  
))
```

```
val records = spark.read.schema(schema)  
  .option("header", "true")  
  .csv("example-database.csv").toDF
```



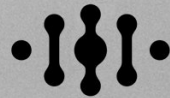
```
CpfSparkType.java
1 public class CpfSparkType
2     implements SparkType<String, CpfType> {
3     @Override
4     public String serialize(CpfType dataType) {
5         return dataType.value();
6     }
7
8     @Override
9     public CpfType deserialize(String sparkData) {
10        return new CpfType(sparkData);
11    }
12
13    @Override
14    public DataType sqlType() {
15        return DataTypes.StringType;
16    }
17 }
```



> Tipos de Dados com Semântica

```
val schema = StructType(Array(
  StructField("id", StringType, true),
  StructField("cpf",
    new CpfSparkType().sqlType(), true)
))

val records = spark.read.schema(schema)
  .option("header", "true")
  .csv("cpfs.csv")
```



Tipo de dados + semântica = muda o jogo

Detectores de tipo = +inteligência

Privacidade + LGPD

Validação

Feature engineering automatizada



Teoria do Mundo Perfeito

1. Não existe Big Data nem código legado
2. Depois que você faz a experimentação, o trabalho acabou
3. Existe um Data Lake com todos os dados que você precisa

EXPERIMENTAÇÃO

MIND THE GAP

PRODUÇÃO

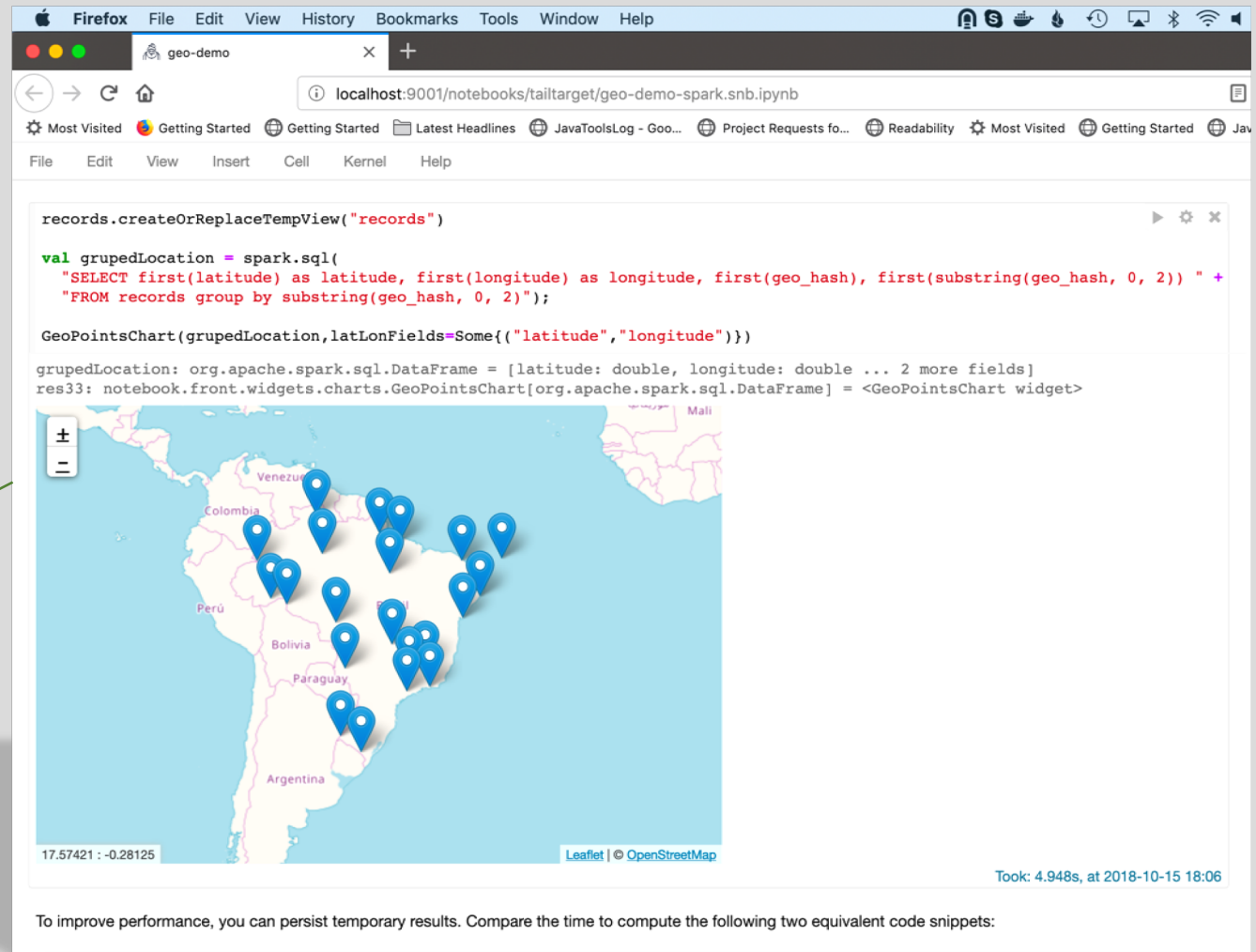
Experimentação:

- Amostras
- Notebooks
- Execução interativa

Produção:

- Big Data
- Schedule
- Execução em lote
- Log

code lake
+
data lake
(amostras)



The screenshot shows a Jupyter Notebook interface in a Firefox browser. The notebook contains the following code and output:

```
records.createOrReplaceTempView("records")

val grupeLocation = spark.sql(
  "SELECT first(latitude) as latitude, first(longitude) as longitude, first(geo_hash), first(substring(geo_hash, 0, 2)) " +
  "FROM records group by substring(geo_hash, 0, 2)");

GeoPointsChart(grupeLocation, latLonFields=Some(("latitude", "longitude")))

grupeLocation: org.apache.spark.sql.DataFrame = [latitude: double, longitude: double ... 2 more fields]
res33: notebook.front.widgets.charts.GeoPointsChart[org.apache.spark.sql.DataFrame] = <GeoPointsChart widget>
```

The output is a map of South America with several blue location pins. The map shows countries including Venezuela, Colombia, Perú, Bolivia, Paraguay, and Argentina. The map is powered by Leaflet and OpenStreetMap. The coordinates shown at the bottom left are 17.57421, -0.28125. The execution time is noted as 4.948s at 2018-10-15 18:06.

To improve performance, you can persist temporary results. Compare the time to compute the following two equivalent code snippets:

The screenshot shows a Jupyter Notebook interface in a Firefox browser. The notebook contains the following code:

```
records.createOrReplaceTempView("records")

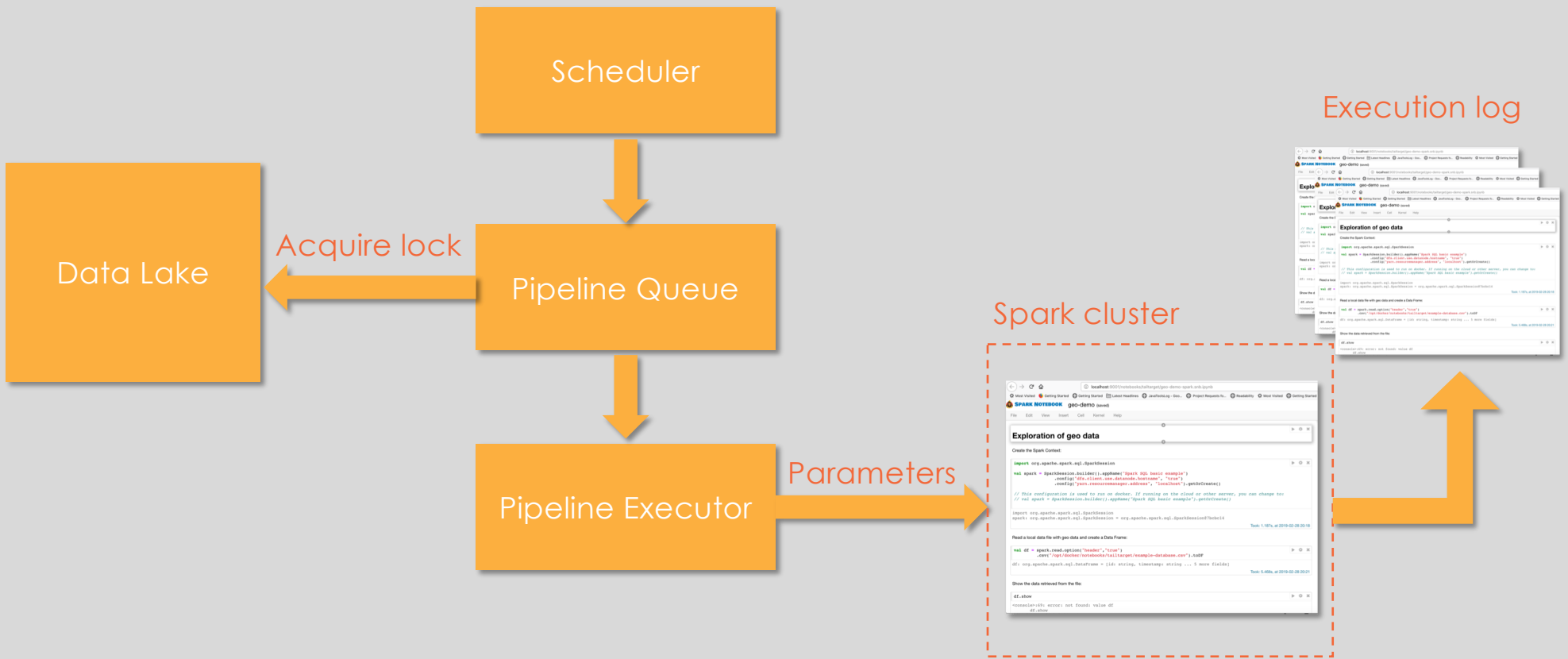
val grupeLocation = spark.sql(
  "SELECT first(latitude) as latitude, first(longitude) as longitude, first(geo_hash), first(substring(geo_hash, 0, 2)) " +
  "FROM records group by substring(geo_hash, 0, 2)");

GeoPointsChart(grupeLocation, latLonFields=Some(("latitude", "longitude")))
grupeLocation: org.apache.spark.sql.DataFrame = [[latitude: double, longitude: double ... 2 more fields]
res33: notebook.front.widgets.charts.GeoPointsChart[org.apache.spark.sql.DataFrame] = <GeoPointsChart widget>
```

Below the code, a map of South America is displayed with several blue location pins. The map shows countries like Venezuela, Colombia, Peru, Bolivia, Paraguay, and Argentina. The map is powered by Leaflet and OpenStreetMap. The coordinates are 17.57421, -0.28125. The map was taken on 2018-10-15 at 18:06.

To improve performance, you can persist temporary results. Compare the time to compute the following two equivalent code snippets:

+ Parâmetros + Scheduler



Scheduler

Data Lake

Acquire lock

Pipeline Queue

Pipeline Executor

Parameters

Spark cluster

Execution log

```
SPARK NOTEBOOK geo-demo (new)
File Edit View Insert Cell Kernel Help

Exploration of geo data

Create the Spark Context
spark = org.apache.spark.sql.SparkSession.builder().appName("spark sql basic example").master("yarn://localhost:8030").getOrCreate()

// This configuration is used to run on docker. If running on the cloud or other server, you can change it.
// see spark = SparkSession.builder().config("spark.yarn.maximizeResourceAllocation", "true").getOrCreate()

spark = org.apache.spark.sql.SparkSession.builder().appName("spark sql basic example").getOrCreate()

Read a local data file with geo data and create a Data Frame
val df = spark.read.option("header", "true").option("inferSchema", "true").load("file:///docker/notesbook/target/example-datалансе.csv").createTempView("example")

Show the data retrieved from the file
df.show
```

Teoria do Mundo Perfeito

1. Não existe Big Data nem código legado
2. Depois que você faz a experimentação, o trabalho acabou
3. Existe um Data Lake com todos os dados que você precisa

PROCURA-SE

CADASTRO
CLIENTES

HISTÓRICO
DE
VENDAS

DADOS DE
CLIMA

CLICKS EM
CAMPANHAS

SCORE DE
CRÉDITO

WEBSITE
ANALYTICS

DADOS DE
ESTOQUE

CEP


















>catálogo
>de datos

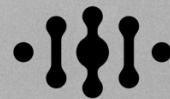
26 Datastores

[Criar novo datastore](#)



Recarregar  Visualização  

CEPs Brasil 2019-08-29 	Emma_Testes_Yp 2020-04-07 	LOG_BAIRRO 2019-10-22 
LOG_CPC 2019-10-22 	LOG_FAIXA_BAIRRO 2019-11-20 	LOG_FAIXA_CPC 2019-10-22 
LOG_FAIXA_LOCALIDADE 2019-11-20 	LOG_FAIXA_UF 2019-11-20 	LOG_FAIXA_UOP 2019-10-22 
LOG_GRANDE_USUARIO 2019-11-20 	LOG_LOCALIDADE 2019-11-20 	LOG_LOGRADOURO 2019-11-20 
LOG_NUM_SEC 2019-10-22 	LOG_UNID_OPER 2019-12-16 	LOG_VAR_BAI 2019-10-22 



Nome
CEPs Brasil (CEPsBrasil)

Somente Leitura?
Não

Diretório
1/dataReceptor/DataReceptorOfCEPsBrasil

Chave Primária
cep

Tipo de Persistência
BIG_DATA

Atualizado em
29/08/2019

Criado em
29/08/2019

Última escrita
29/08/2019

Esquema

Nome	Tipo	Anonimizada	Aumentada
cep 	STRING		
tipologradouro	STRING		
logradouro	STRING		
bairro	STRING		
cidade	STRING		
uf	STRING		

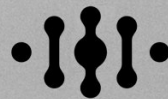
Amostra(s)

Nome
Sample of CEPs Brasil *



> Amostras

```
val records =  
    spark.read.option("header", "true")  
              .csv("example-database.csv").toDF  
  
records.createOrReplaceTempView("records")  
  
val sample = spark.sql(  
    "SELECT * FROM records "+  
    "TABLESAMPLE (25 PERCENT) "  
)
```



Precísamos
falar sobre
LGPD



DADOS PII

BASES LEGAIS

DATA LINEAGE



(555) 555-1234

PII

Detector de Tipos de Datos

da2066c24702746781ff4



(555) 555-1234

PII

Detector de Tipos de Dados

da2066c24702746781ff4

Base Legal

- Consentimento
- Proteção ao crédito

DATA LINEAGE

CLIENTES

```
SPARK NOTEBOOK Credit Score Calculation new1
File Edit View Insert Cell Kernel Help
import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder().appName("Spark SQL basic example")
    .config("dfs.client.use.datanode.hostname", "true")
    .config("yarn.resourcemanager.address", "localhost").getOrCreate()

import org.apache.spark.sql.SparkSession
spark.org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession66f0ab3fd

val values = List(List("1", "One"), List("2", "Two"), List("3", "Three"), List("4", "4"))
import spark.sqlContext
val df = values.map {
  (id, value) => (id, value)
}.toDF("id", "value")
df.printSchema
df.show(false)

root
 |-- id_1 string (nullable = true)
 |-- value_1 string (nullable = true)
-----
[id_1,value_1]
-----
```

May, 5th, 2019

Scores
De
Crédito

```
SPARK NOTEBOOK Good Players Calculation new1
File Edit View Insert Cell Kernel Help
import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder().appName("Spark SQL basic example")
    .config("dfs.client.use.datanode.hostname", "true")
    .config("yarn.resourcemanager.address", "localhost").getOrCreate()

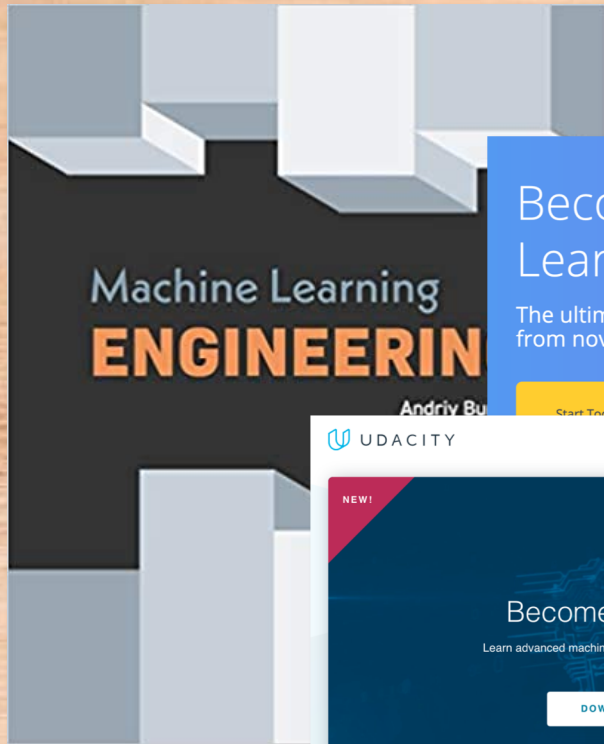
import org.apache.spark.sql.SparkSession
spark.org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession66f0ab3fd

val values = List(List("1", "One"), List("2", "Two"), List("3", "Three"), List("4", "4"))
import spark.sqlContext
val df = values.map {
  (id, value) => (id, value)
}.toDF("id", "value")
df.printSchema
df.show(false)

root
 |-- id_1 string (nullable = true)
 |-- value_1 string (nullable = true)
-----
[id_1,value_1]
-----
```

May, 6th, 2019


Bons
Pagadores



Become a Machine Learning Engineer

The ultimate guide to career transformation—go from novice to professional with a step-by-step path.

Start Today

A circular inset image showing a woman with dark hair, wearing a teal long-sleeved shirt, sitting at a desk and working on a computer. She is looking towards the camera.

UDACITY Programs ▾ Careers ▾ For Enterprise ▾ | Sign In GET STARTED

NEW!

NANODEGREE PROGRAM

Become a Machine Learning Engineer

Learn advanced machine learning techniques and algorithms -- including how to package and deploy your models to a production environment.

DOWNLOAD SYLLABUS ENROLL NOW

09 DAYS 11 HRS 17 MIN 32 SEC

▶

Engenharia de Machine Learning

1. Implantar modelos em produção
2. Catálogo de modelos
3. Versionamento e rastreamento de modelos



Machine Learning

Run training in cell READ_1



You are online



MACHINE_LEARNING_1

MODEL_1

Choose type of execution Training Prediction

Run test on READ_1










Method Choose * Implementation Choose * +Add


Method	Stage	Custom Parameters	
FEATURE	VectorAssembler	inputCols, outputCol	
MACHINE_LEARNING	KMeans	featuresCol, predictionCol	


Save model?

No data available, please press run to execute the cell






-  Home
-  Data Lake
-  Pipelines
-  Models
-  Schedule
-  Executions
-  Dashboards
-  Marketplace
-  Integrations

 Models


 Account: Demo
Customer: 2


4 Models




Reload  Visualization  

AptosSBC_LR
2020-08-18 

London Bike - kmeans
2020-08-21 

US Census Income
2020-08-24 

teste-modellineage
2020-08-26 



- Home
- Data Lake
- Pipelines
- Models
- Schedule
- Executions
- Dashboards
- Marketplace
- Integrations



Models > London Bike - kmeans



Account: Demo
Customer: 2

Executed by pipeline
London Bike - kmeans

Last written at
2020-08-26

Created at
2020-08-21

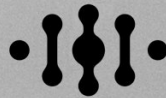
Updated em
2020-08-26

Stages

Implementation Class	Parameters
org.apache.spark.ml.feature.VectorAssembler	outputCol : feature inputCols : distance_from_city_center, duration, num_trips
org.apache.spark.ml.clustering.KMeans	featuresCol : feature k : 4 predictionCol : prediction

Metrics

k : 4 trainingCost : 2011028850.9182916 silhouette : 0.9852745744646104
clusterCenters : [[4.38405851599234,1311.901797195024,49.38569604086846],[7.340377184173547,146561.53846153847,13],[4.592711138639755,45405.24844720497,23],[6.123645553776085,13545.683495035795,32.529411764705884]]



Precísamos
falar sobre
Model
Lineage

- Target Audience & Insights Lab
- Home
- Data Lake
- Pipelines
- Models
- Schedule
- Executions
- Dashboards
- Marketplace
- Integrations

Models > London Bike - kmeans > Model Lineage

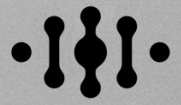
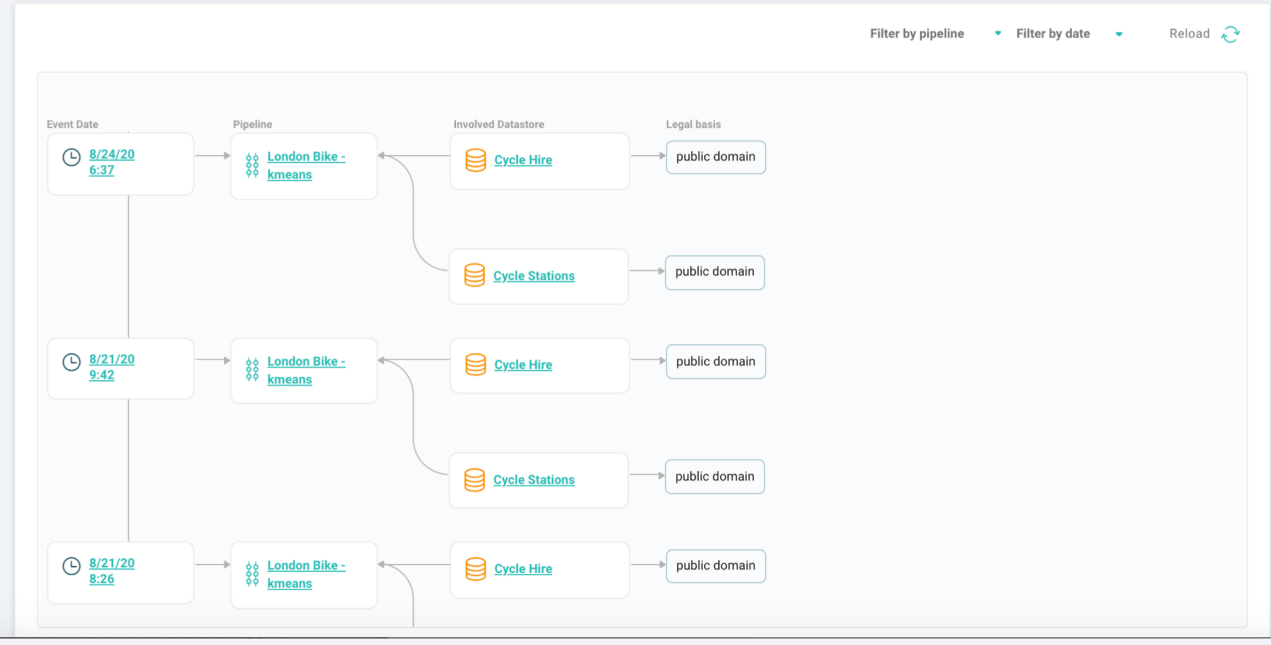
Account: Demo
Customer: 2

Name: London Bike - kmeans [View Info](#)

Updated at: 2020-08-26

Created at: 2020-08-21

Last written at: 2020-08-26





Engenharia de Dados e Engenharia de Machine Learning

como fazer projetos de Data Science usando um
processo sustentável

Fabiane Bizinella Nardon
Chief Data Scientist
@fabianenardon

Recursos:

Frame photo created by rawpixel.com - www.freepik.com

Photo by [Shirly Niv Marton](#) on [Unsplash](#)

Photo by [Tim Gouw](#) on [Unsplash](#)

Image by https://pixabay.com/users/www_darkworkx_de-2044000

Image by [TeroVesalainen](#) from [Pixabay](#)

Image by [MR1313](#) from [Pixabay](#)