data® expert.

Data Science & Machine Learning

CRISP-DM Deployment

Consultor: Murilo Mendonça

Murilo Mendonça

ML Engineer
@Ambev Tech

Eng. Mecânico

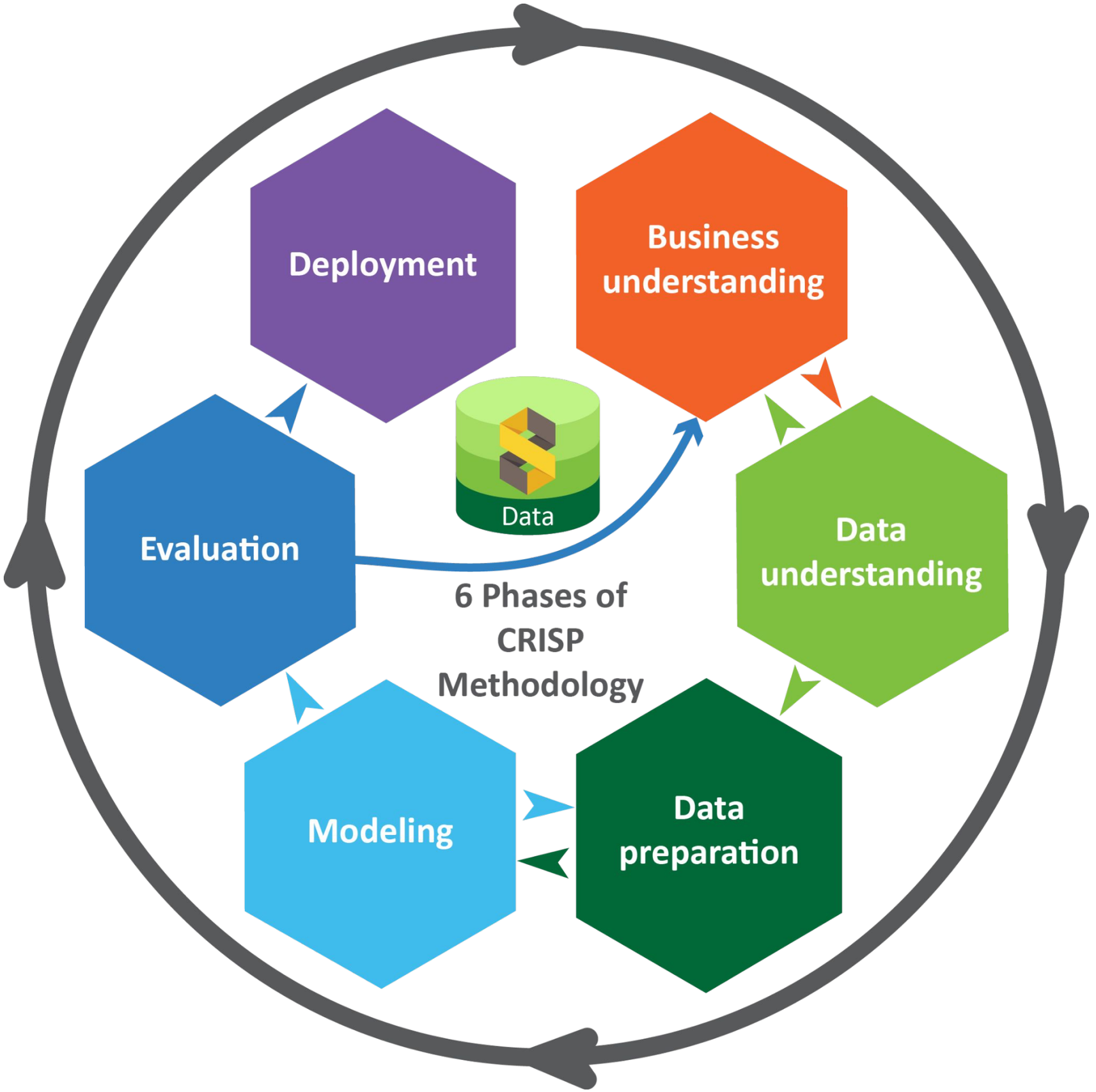
Fraunhofer Institute

Lactec

Ambev

Ambev Tech

Muri Tech


Deployment
Business understanding
Data
Evaluation
Data understanding
6 Phases of CRISP Methodology
Modeling
Data preparation

Tópicos do Módulo:

01

Ambiente de produção

02

Planejando o deployment - Batch

03

Planejando o deployment - API

04

Planejando o monitoramento

05

Planejando a manutenção

06

Produzindo o report final

07

Produzindo uma apresentação final

data expert.®

Data Science & Machine Learning

02. Ambiente de Produção

Consultor: Murilo Mendonça

O que Veremos Nesta Aula:

01

Definição

02

Testes

03

Serialização de modelos

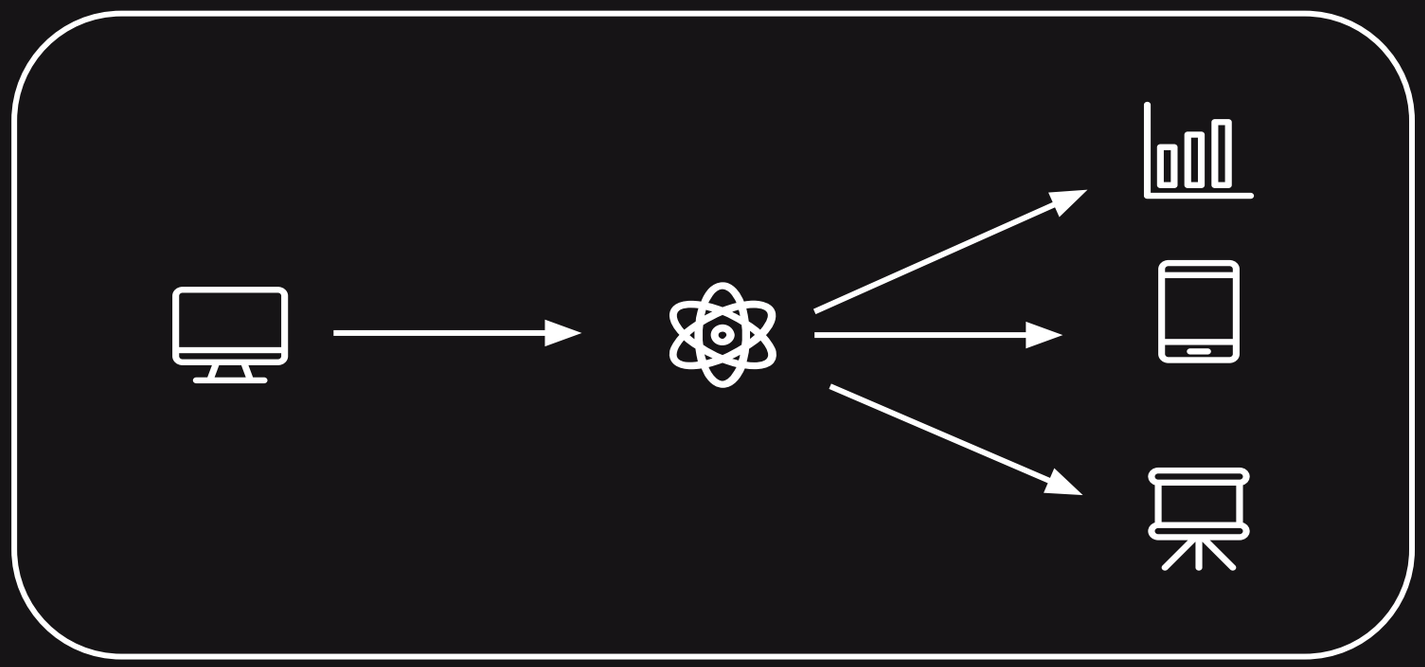
O que é?

Usuário:
é o que ele enxerga e interage

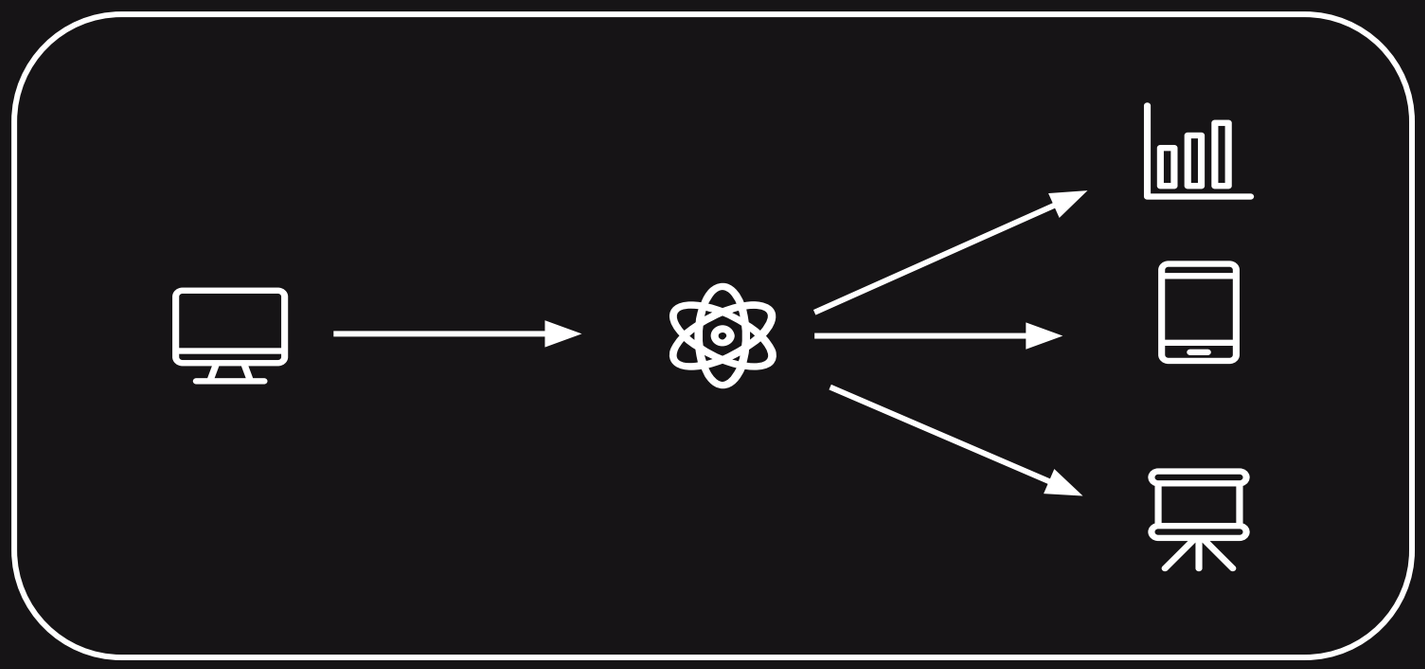
Desenvolvedor:
é uma réplica do seu ambiente de trabalho

Como?

NONPROD



PROD



Por que?

Controle de Versão

Governança

Entregas Faseadas

Testes

Unitários

Integrados

Teste Unitário

```python
def custom_sum(number):
    return number + 10
```

```python
def test_custom_sum():
    number = 10

    actual_result = custom_sum(number)
    expected_result = 20

    assert isinstance(actual_result, int)
    assert actual_result == expected_result
```

Teste Unitário

```
============================= test session starts ==============================
platform darwin -- Python 3.7.12, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
rootdir: /Users/murilomendonca/Documents/repos/dnc-classes
collected 1 item

test_custom_sum.py .                                                     [100%]

============================== 1 passed in 0.01s ===============================
```

Teste Integrado

```python
def test_load_model(connection_string):

    loaded_model = MyClass().load_model(connection_string)

    assert isinstance(loaded_model, mlflow.pyfunc.PythonModel)
```

```python
def test_make_predictions(loaded_model, data):

    predictions = MyClass().make_predictions(loaded_model, data)

    assert isinstance(predictions, list)
    assert predictions[0] == 1
```

Serialização

"Ok, treinei meu modelo, as métricas estão ok. "

E agora?

Serialização

`pickle` — Python object serialization

Source code: Lib/pickle.py

cloudpipe / **cloudpickle** Public

<> Code Issues 59 Pull req

uqfoundation / **dill** Public

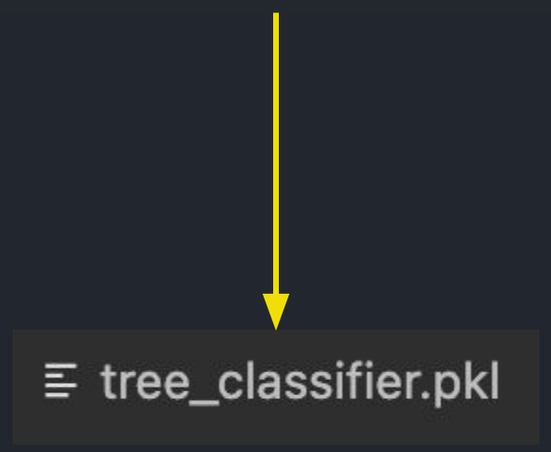
<> Code Issues 136

mlflow™

Serialização

```python
# Train model
model = DecisionTreeClassifier(max_depth=max_depth)
model.fit(X, y)
```

```python
# Save model to disk
with open("tree_classifier.pkl", "wb") as file:
    pickle.dumps(model, file)
```



Serialização


```
????sklearn.tree._classes??DecisionTreeClassifier???)??}?(?     criterion??gini??splitter??best??       max_depth?K
?min_samples_split?K?min_samples_leaf?K?min_weight_fraction_leaf?G?max_features?N?max_leaf_nodes?N?random_state?N?
min_impurity_decrease?G?min_impurity_split?N?class_weight?N?    ccp_alpha?G?n_features_in_?K?n_features_?K?
n_outputs_?K?classes_??numpy.core.multiarray??_reconstruct????numpy??ndarray???K??Cb???R?(KK??h?dtype????
i8??????R?(K?<?NNNJ????J????Kt?b?C?t?b?
n_classes_?h?scalar???h'C???R??max_features_?K?tree_??sklearn.tree._tree??Tree???KhhK??h??R?(KK??h'?C?t?bK
??R?}?(h        K?
node_count?K?nodes?hhK??h??R?(KK??h$?V56?????R?(K?|?N(?
left_child??right_child??feature??      threshold??impurity??n_node_samples??weighted_n_node_samples?t?}?(hJh$?
i8??????R?(Kh(NNNJ????J????Kt?bK??hKhUK??hLhUK??hMh$?f8?????R?(Kh(NNNJ????J????Kt?bK??hNh\K ??hOhUK(??hPh\K0???uK8KK
t?b?B???@VUUUUU?????b@?????????????????????????2I@?7?dY@?????????????????????????h?WH??7?6K@?????????????????????????
???<b?ƥ?.G@?t?b?values?hhK??h??R?(KKKK??h\?CxI@I@I@I@I@?H@@?7?F@?t?bub?_sklearn_version??0.24.2?ub.
```

Serialização

tree_classifier.pkl


```python
def load_model():
    with open("tree_classifier.pkl", "rb") as file:
        model = pickle.load(file)
    return model
```

Onde Armazenar?

File System

Cloud Storage

Docker image

Banco de dados

Resumo

Definição

Testes

Serialização

data expert.®

Data Science & Machine Learning

03. Planejando o Deployment (BATCH)

Consultor: Murilo Mendonça

O que Veremos nesta Aula:

01

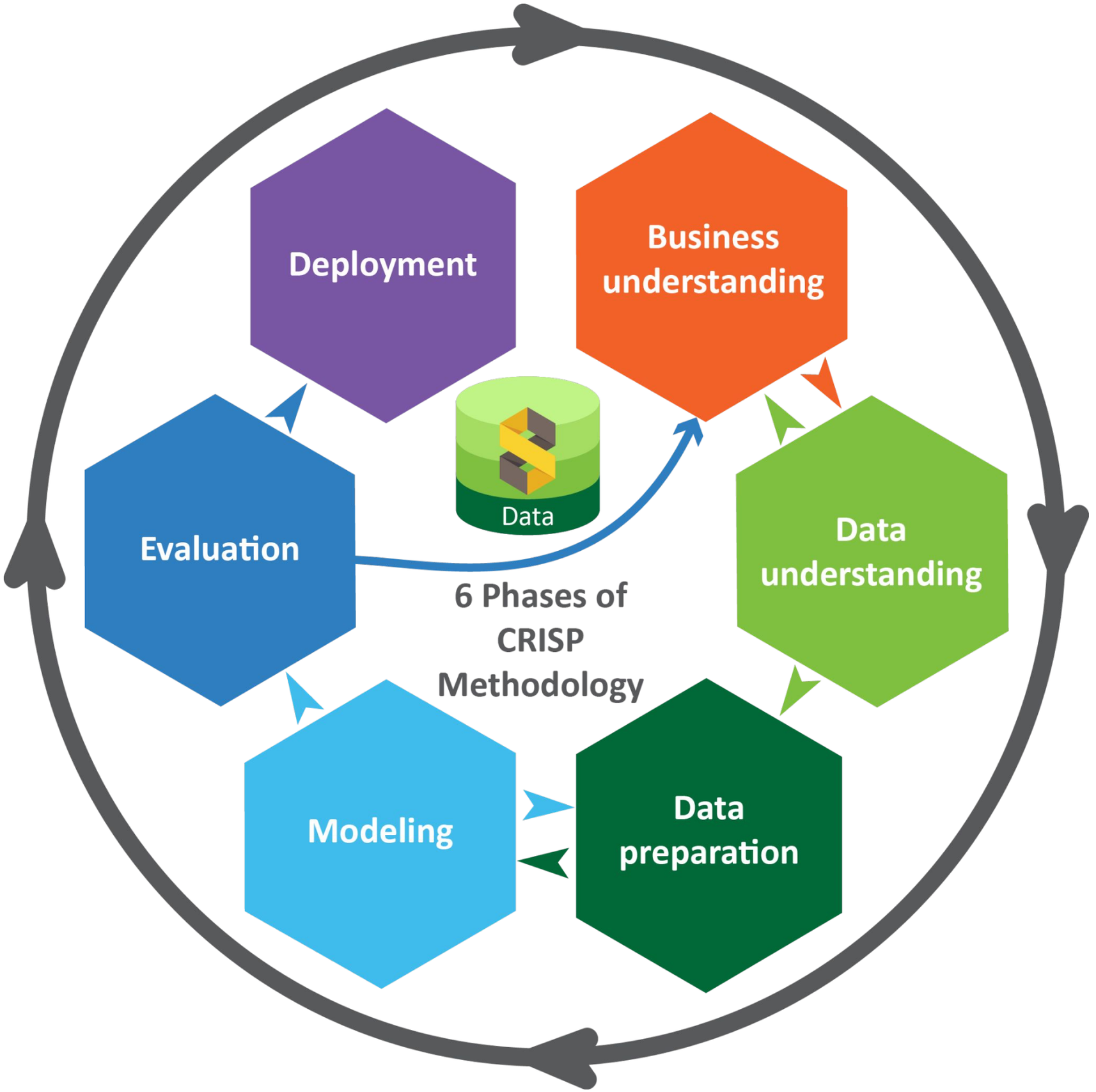
Pipelines

02

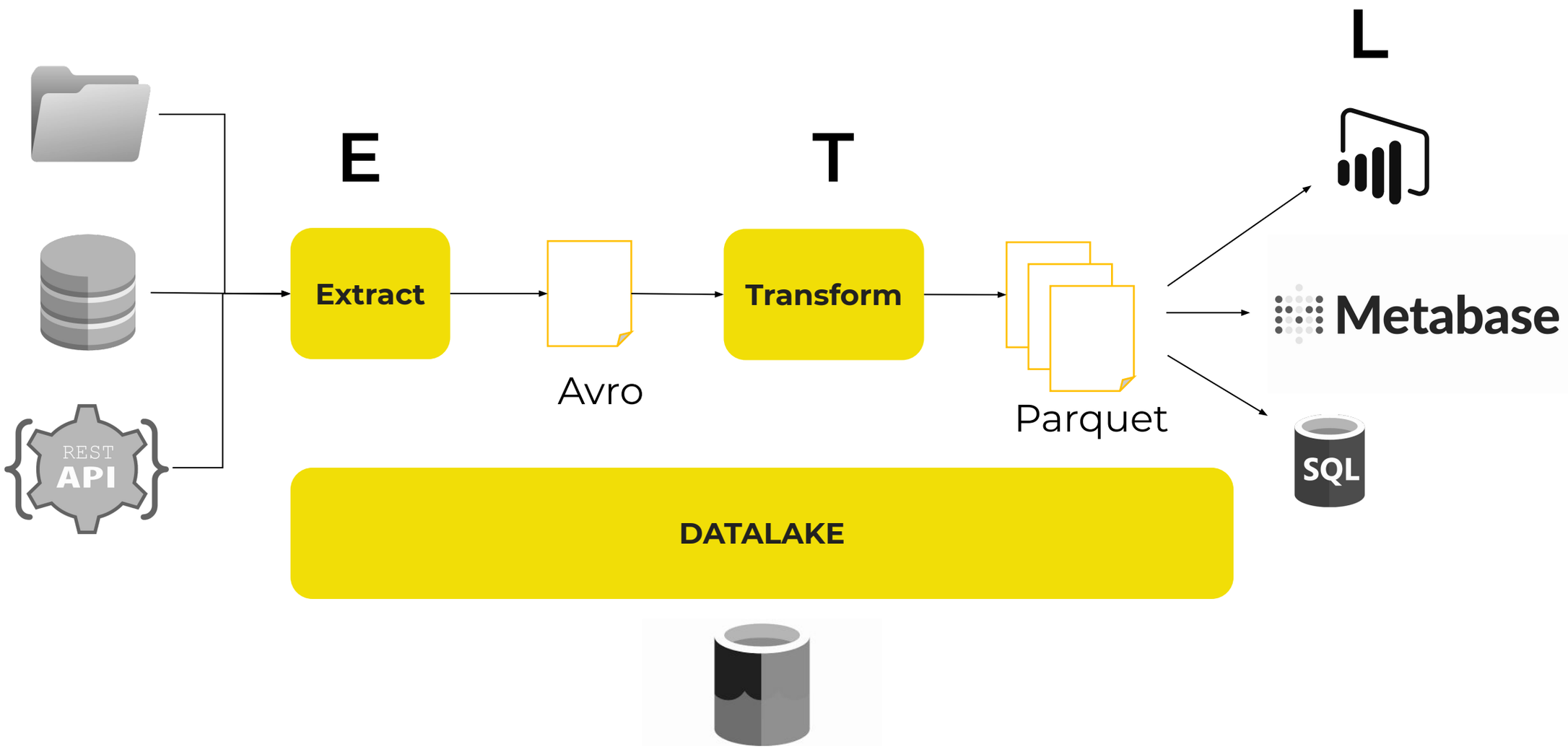
Spark + Mlflow

03

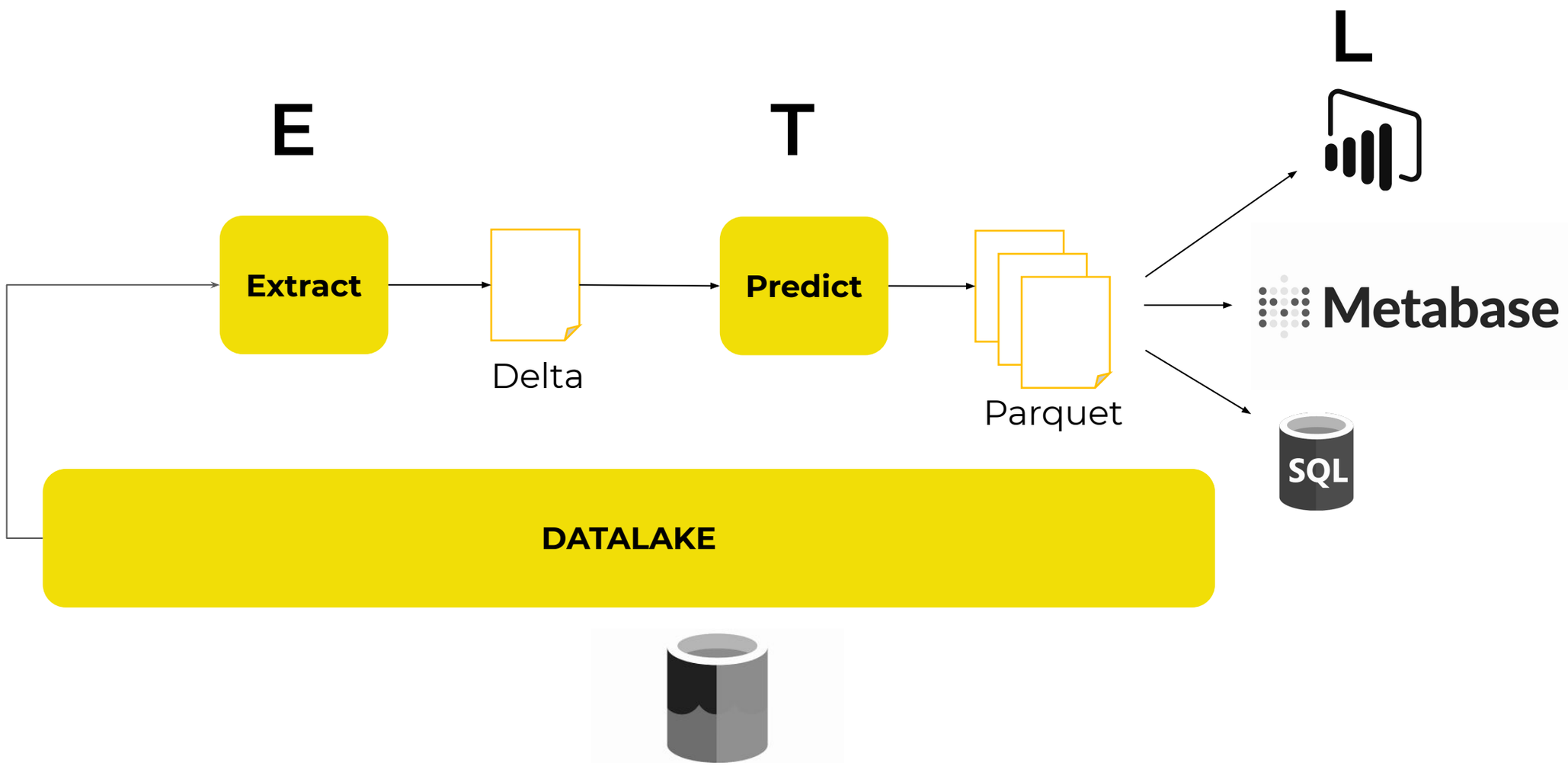
Periodicidade e Orquestração


Deployment
Business understanding
Data
Evaluation
Data understanding
6 Phases of CRISP Methodology
Modeling
Data preparation

Pipeline de Dados



Pipeline ML



MLflow + Spark

Ferramenta de análise, treinamento e processamento de dados **em escala**

Plataforma de gerenciamento do ciclo de vida de modelos

Mlflow + Spark

```
logged_model = 'runs:/<run_id>/model'

# Load model as a Spark UDF.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model)

# Predict on a Spark DataFrame.
columns = list(df.columns)
df.withColumn('predictions', loaded_model(*columns)).collect()
```

Periodicidade

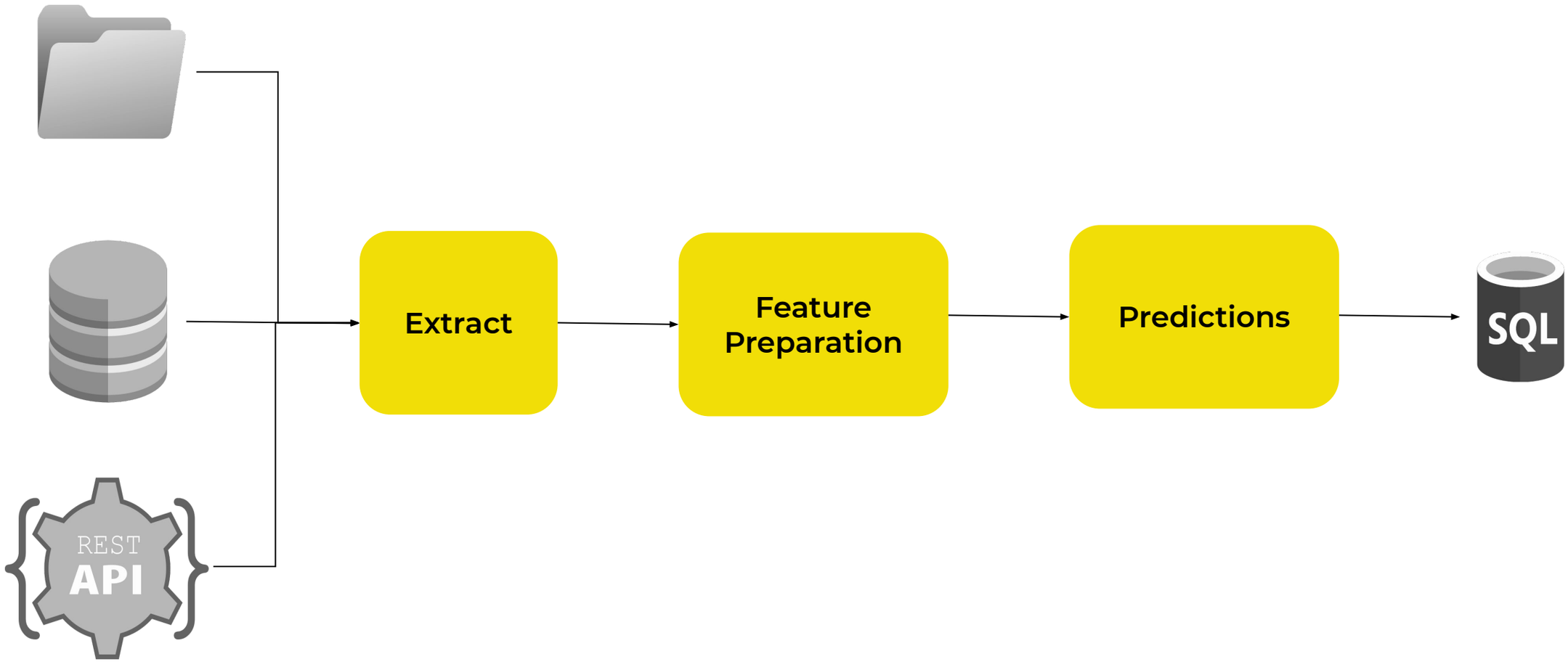
De quanto em quanto tempo?

Onde?

Quais são as etapas do seu pipeline?

DAG

Directed Acyclic Graph



Orquestração

Código

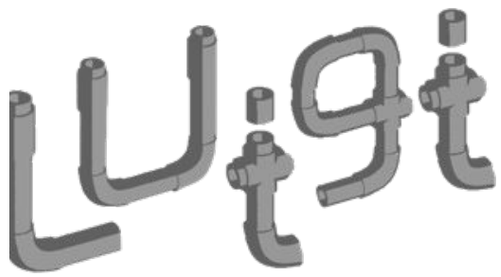
Crontab

Ferramentas de mercado

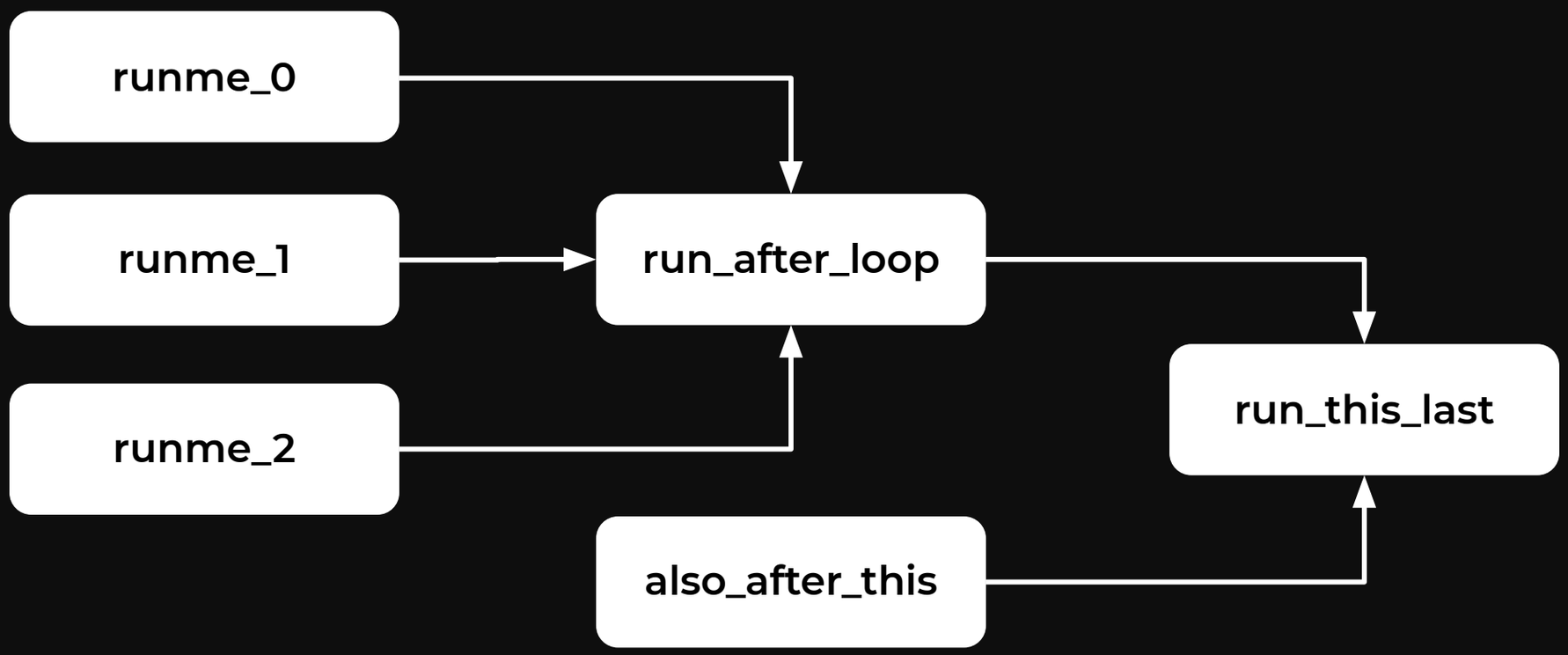
Orquestração

Kubeflow

Apache
Airflow



Orquestração



Resumo

01

Pipelines

02

Spark + Mlflow

03

Periodicidade e Orquestração

data expert.®

Data Science & Machine Learning

04. Planejando o Deployment (API)

Consultor: Murilo Mendonça

O que Veremos nesta Aula:

01

API

02

Complexidade

03

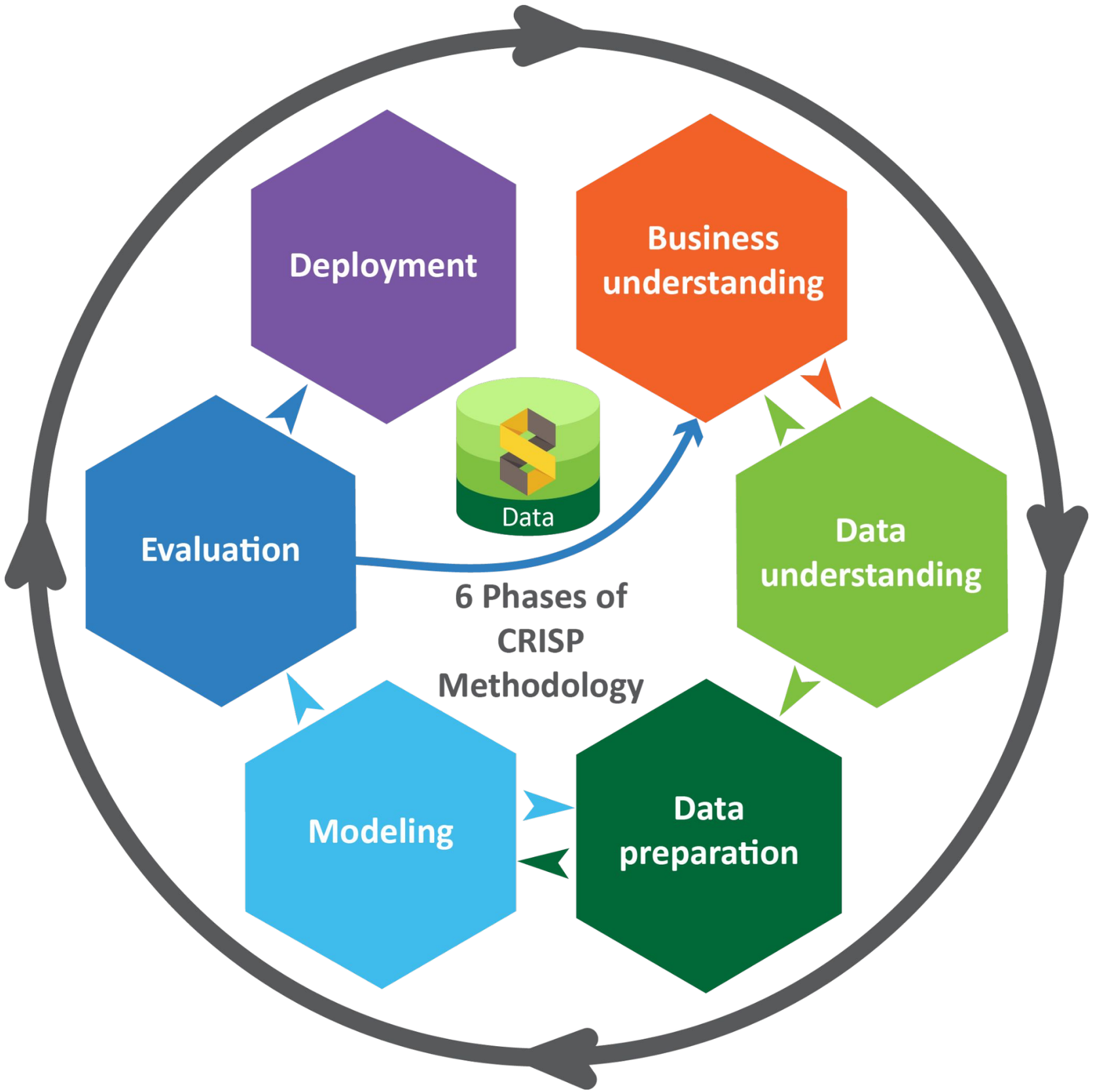
Mini-Batch

04

Ambiente de execução

05

Métricas


Deployment
Business understanding
Data
Evaluation
Data understanding
6 Phases of CRISP Methodology
Modeling
Data preparation

API

API

PREDICTIONS
[True]

POST
[1,2,3,4,5,6]

MODELO

```json
{
    "model": "iris_flower_prediction",
    "data": {
        "SEPAL_LENGTH": [5.3],
        "SEPATL_WIDTH": [6.7],
        "PETAL_LENGTH": [2.1],
        "PETAL_WIDTH": [2.3]
    }
}
```

Body ▾ 200 OK 3.72 s 256 B

Pretty | Raw | Preview | Visualize | JSON ▾

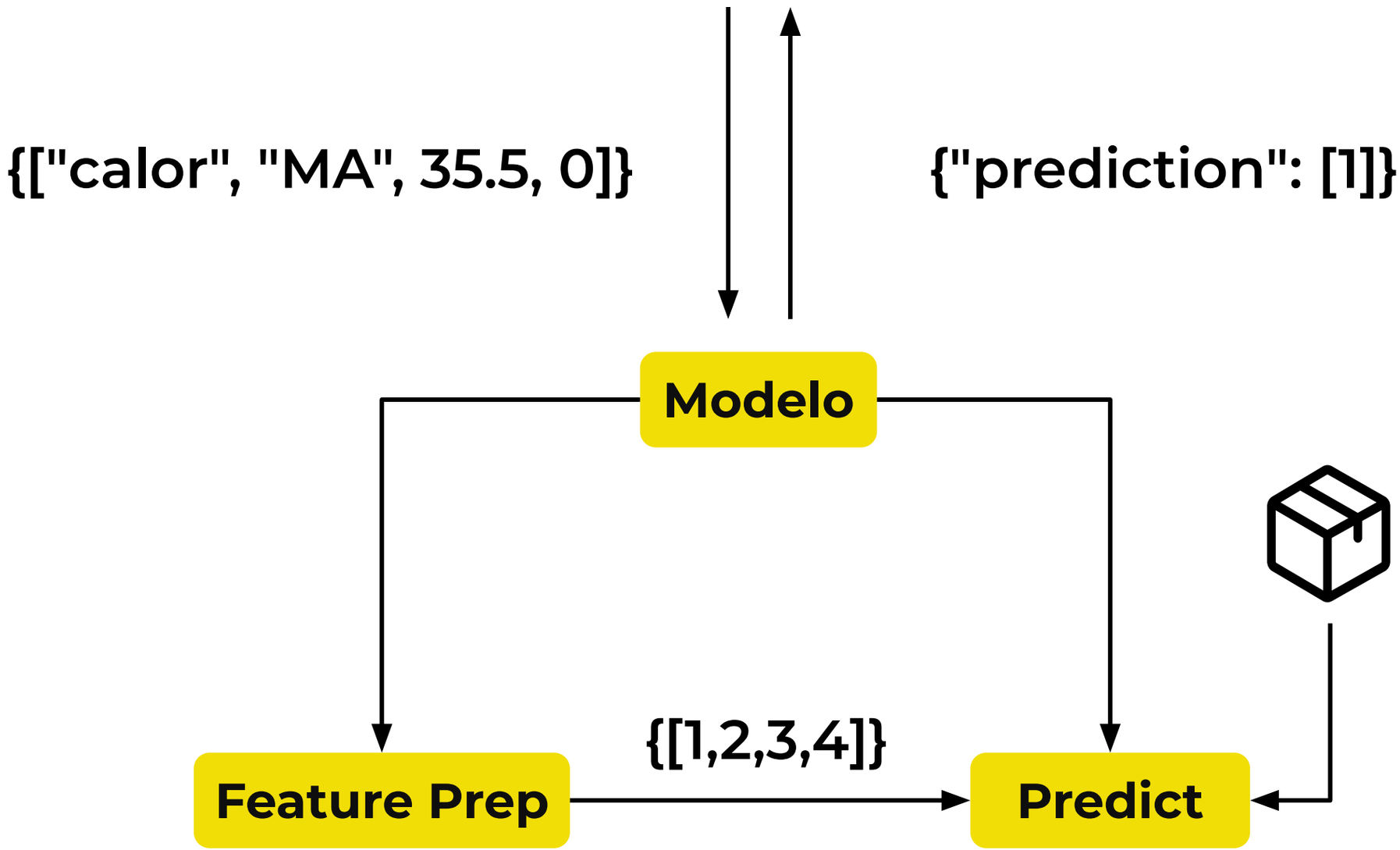
```json
{
    "predictions": [
        2
    ]
}
```

Complexidade

- Intermitência da aplicação
- Contrato
- Documentação
- Autenticação
- Infraestrutura
- Performance

É de uma API que você precisa?

Mini-batch


{["calor", "MA", 35.5, 0]}
{"prediction": [1]}
Modelo
Feature Prep
{[1,2,3,4]}
Predict

Ambiente

Orçamento

Trade-off

Requerimento de projeto

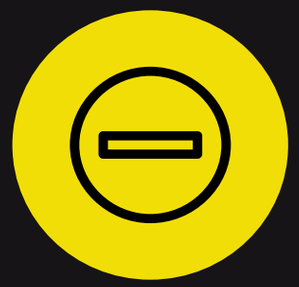
Ambiente de Execução

- Máquina virtual
- Serviços gerenciados
  - Databricks
  - Sagemaker, Azure ML, etc.
  - Algorithmia, Seldon, etc.
- Serverless
  - Functions
  - Container Instances
- Swarm ou Kubernetes

Métricas de App

Latência

Tráfego



Erros

Saturação

Métricas de ML

Definir as métricas de sucesso (F1, RMSE, etc.)

Como medir?

Métricas de negócio (RoI, tempo médio, etc.)

Qualidade de dados

Resumo

- API
- Complexidade
- Mini-Batch
- Ambiente de execução
- Métricas

data expert.®

Data Science & Machine Learning

05. Planejando o Monitoramento

Consultor: Murilo Mendonça

O que Veremos nesta Aula:

01

Métricas de ML

02

Métricas de aplicação

03

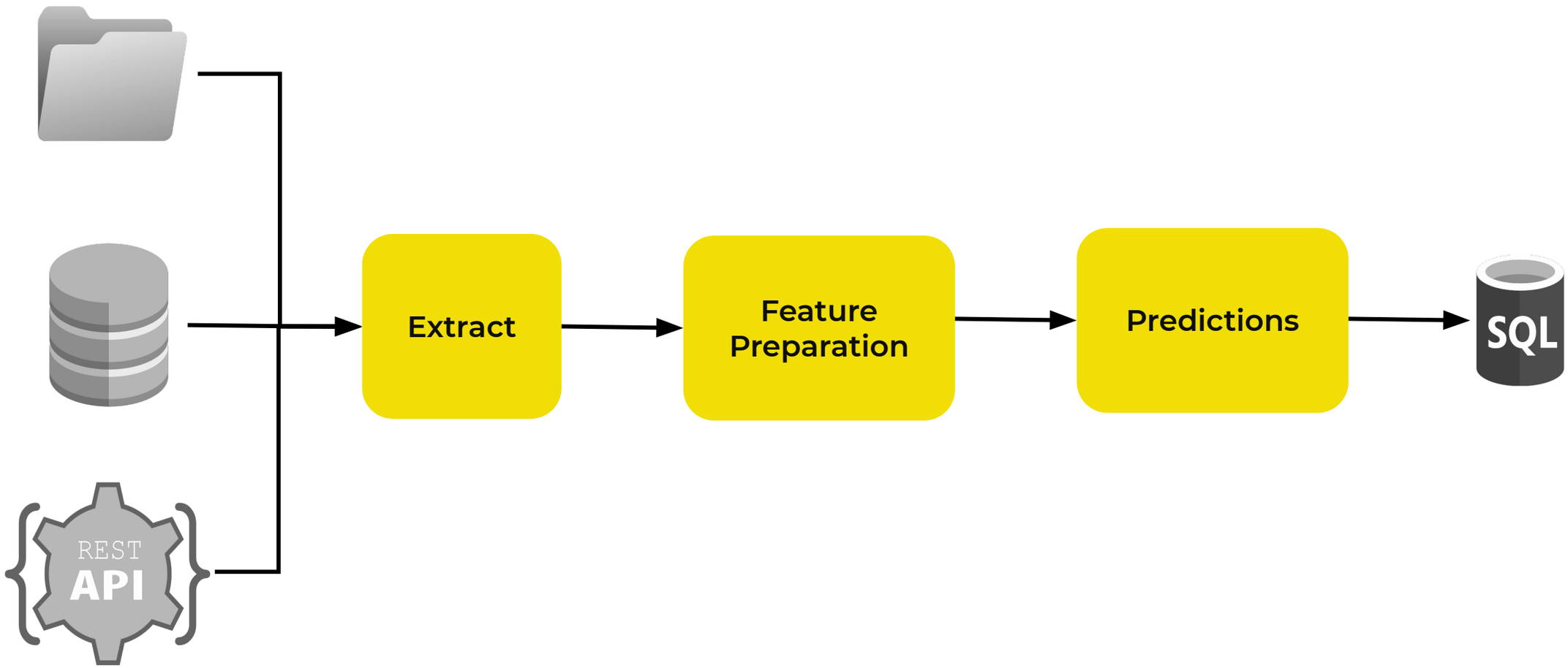
Ferramentas de mercado

04

Caso real

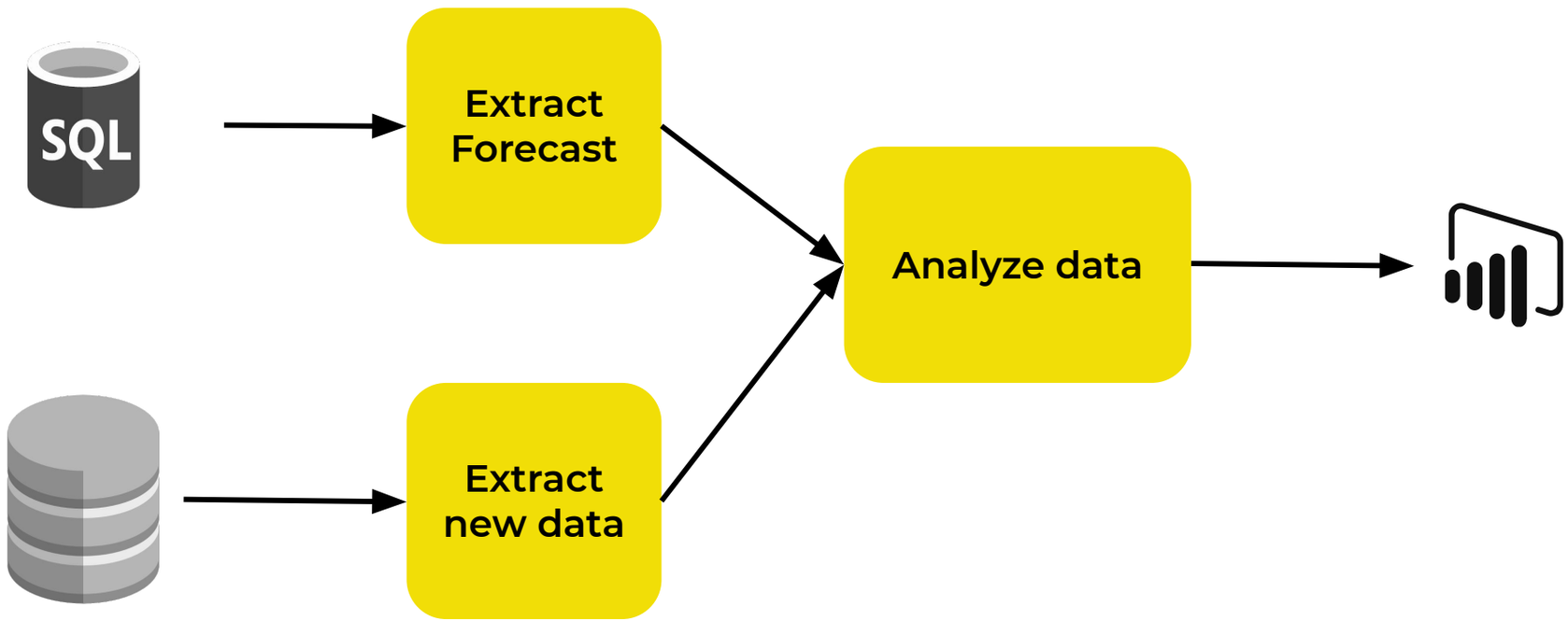
Métricas de ML

Prediction DAG

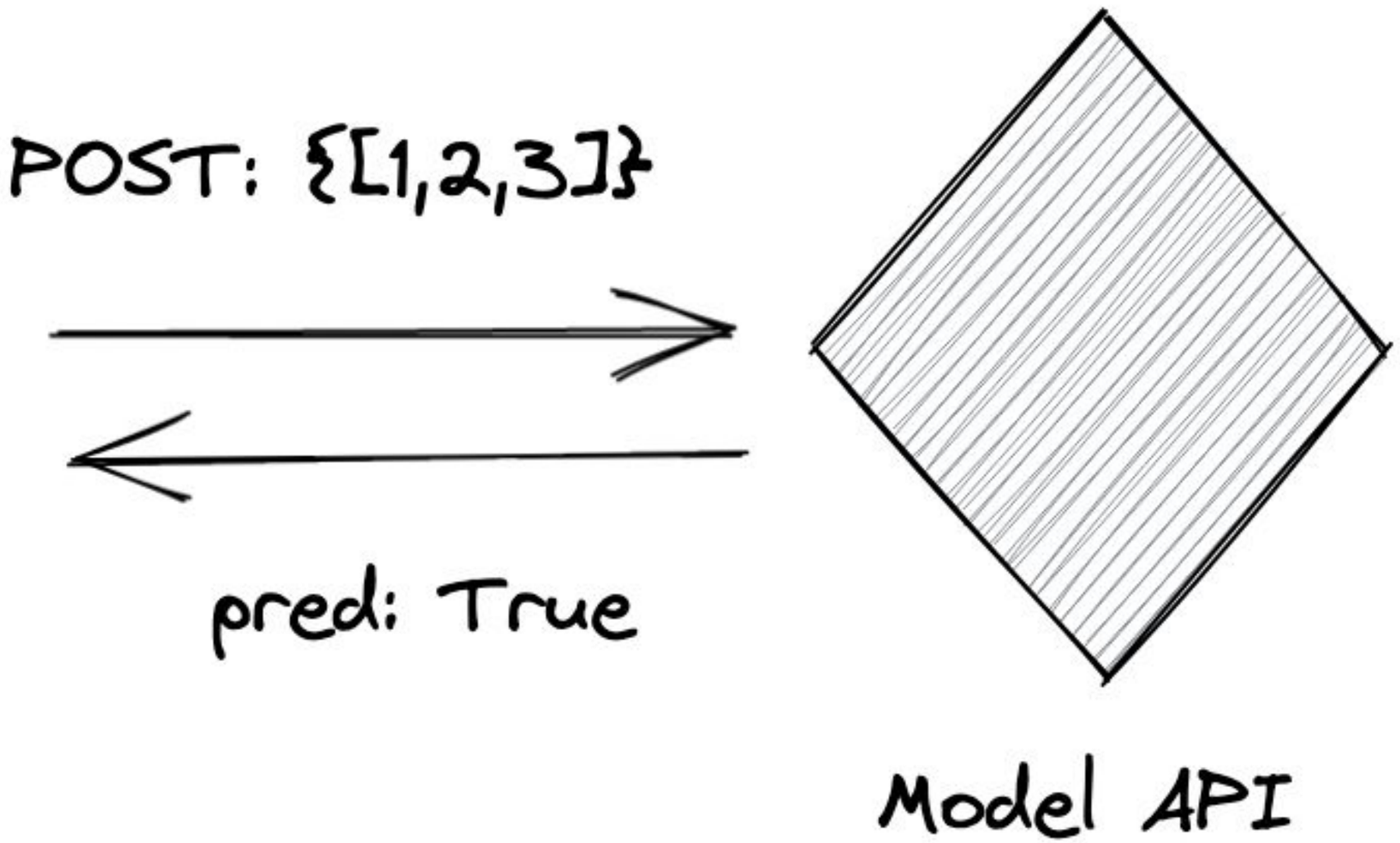


Métricas de ML

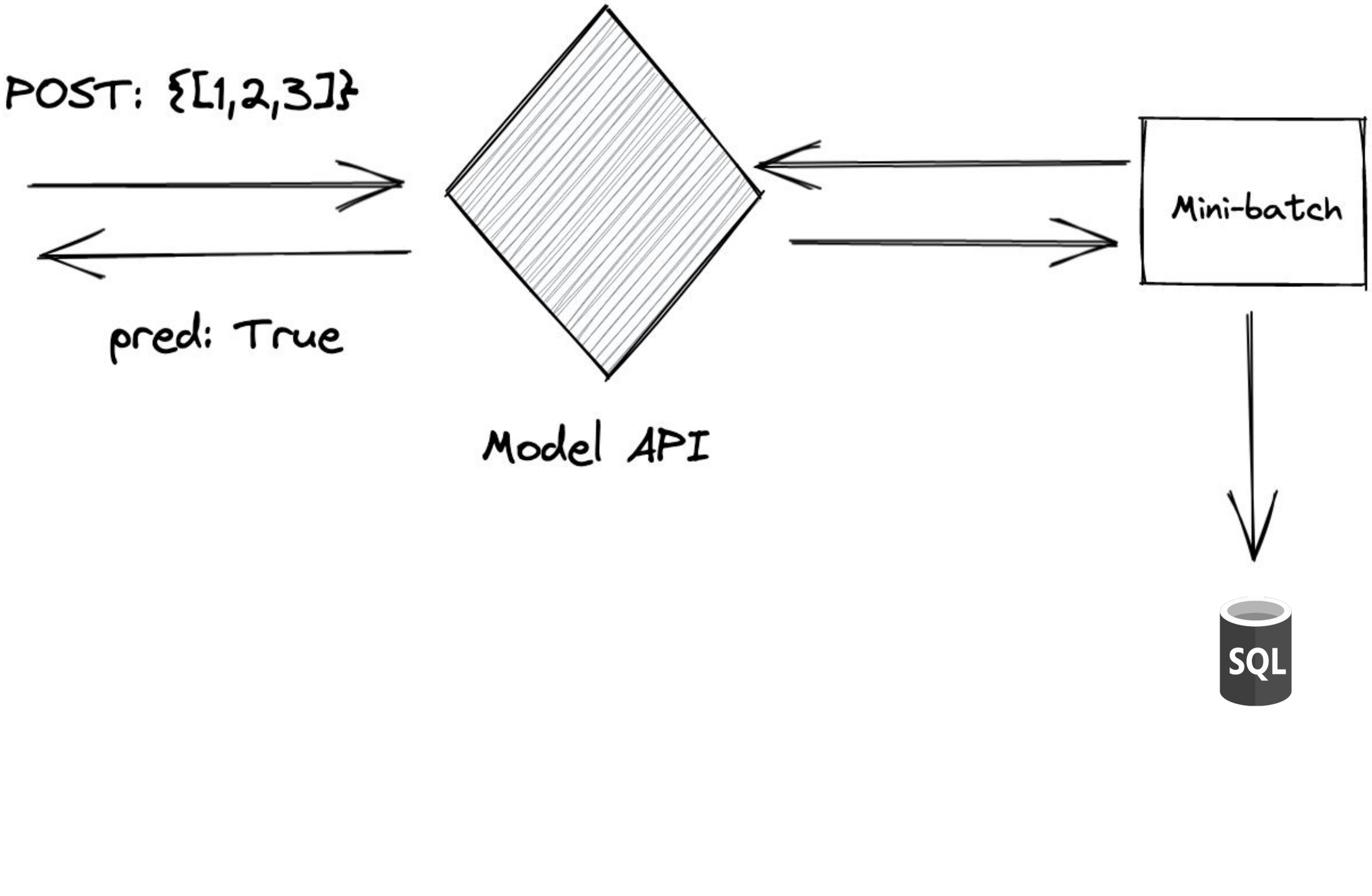
Metrics DAG



Métricas de ML



Métricas de ML



Métricas de App

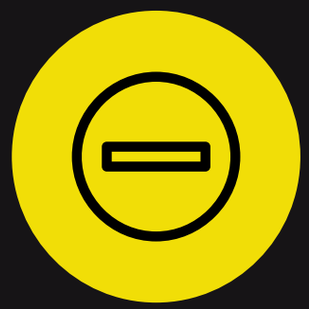
Tivemos um pico de latência às 4:00am.

O que mais aconteceu?

Métricas de App

Latência

Tráfego



Erros

Saturação

Métricas de App

Latência

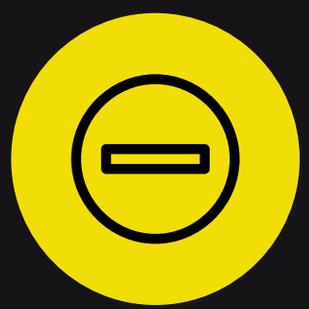
- 500ms
- Timeout
- Percentil

Métricas de App

Tráfego

- Request
- Request/Second
- Request/Response

Métricas de App



Erros

- 3xx - Redirect
- 4xx - Bad Request
- 5xx - Internal Error

Métricas de App

Saturação

- CPU Utilization
- Memory Usage
- Avaible pods, nodes etc.

Métricas de App

Google Site Reliability Engineering

Home | Spotlight | Resources | Books | Mobaa | Classroom | Careers

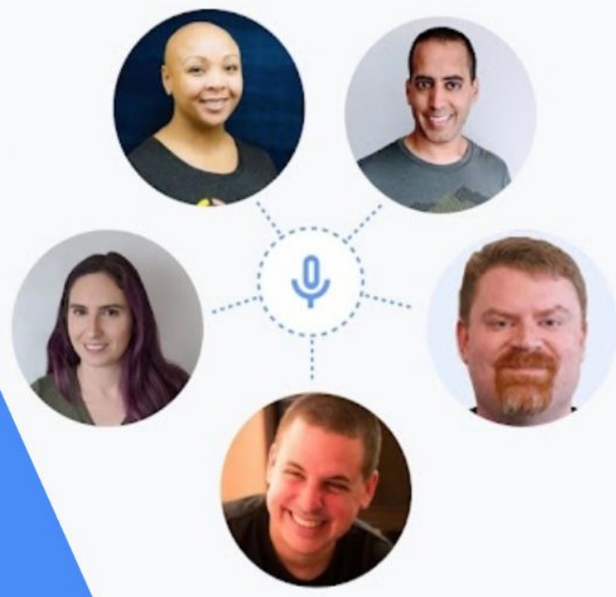
What is Site Reliability Engineering (SRE)?

SRE is what you get when you treat operations as if it's a software problem. Our mission is to protect, provide for, and progress the software and systems behind all of Google's public services — Google Search, Ads, Gmail, Android, YouTube, and App Engine, to name just a few — with an ever-watchful eye on their availability, latency, performance, and capacity.

SRE Experts

Hear veteran Googlers describe their experiences as SREs: how their backgrounds led them to their current roles, and what their day-to-day work looks like.

Read more →



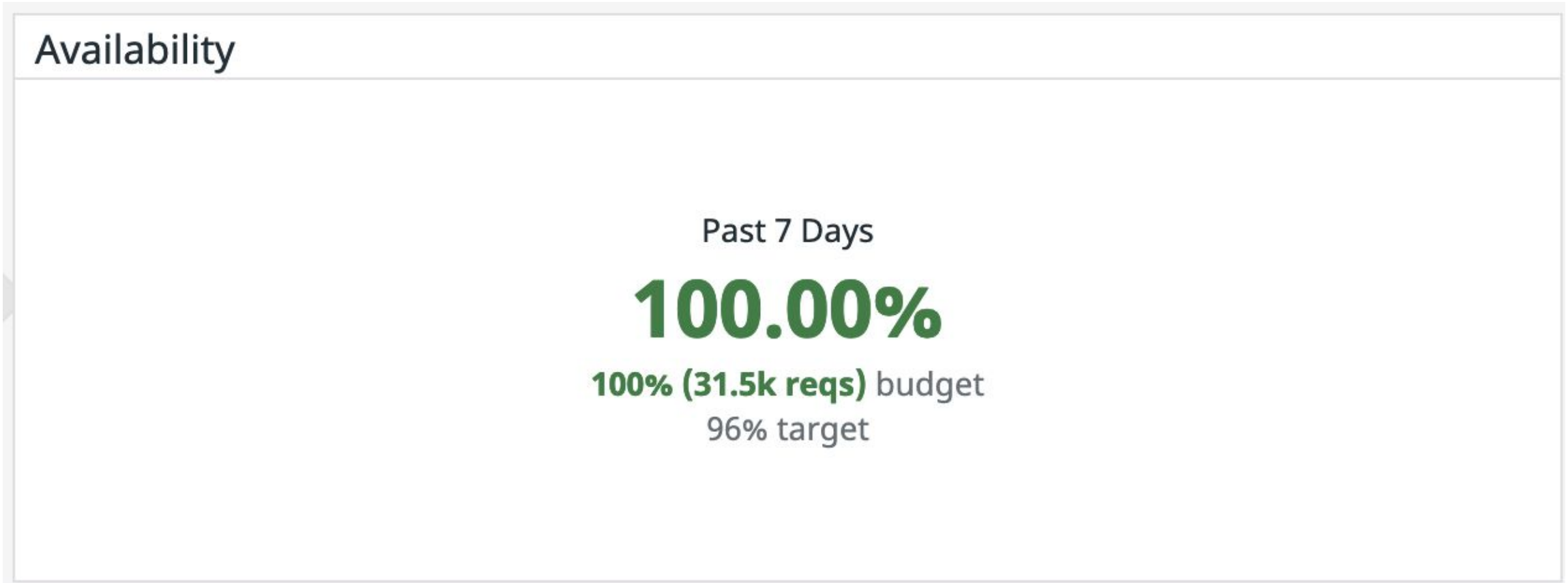
Métricas de App

Service Level Objetive

Service Level Agreement

Service Level Indicator

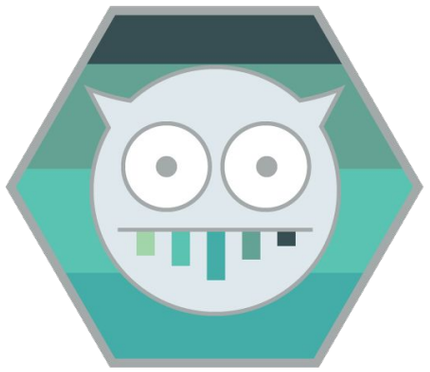
Métricas de App

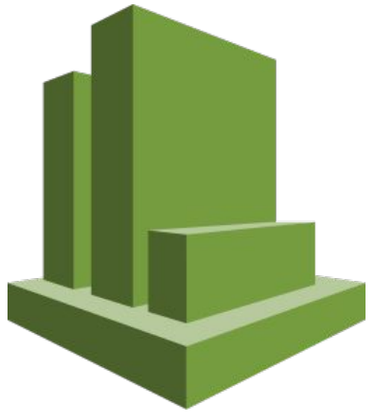


Métricas de App

RESPONSE CODE	↓ COUNT	AVG:DURATION	PC95:DURATION	PC99:DURATION
200	995	4.32s	7.31s	12.57s
200	91	8.86s	14.01s	22.43s
200	36	10.47s	18.06s	26.69s
400	6	521ms	821ms	821ms

Ferramentas





General / Model Metrics

Last 30 minutes | 30s

continuous: feature_0_value | discrete: feature_1_value | interval: 2m | duration: auto

Alerts

KL Divergence (30m)

K-S Test (30m)

feature_1_value

Baseline distribution over training set

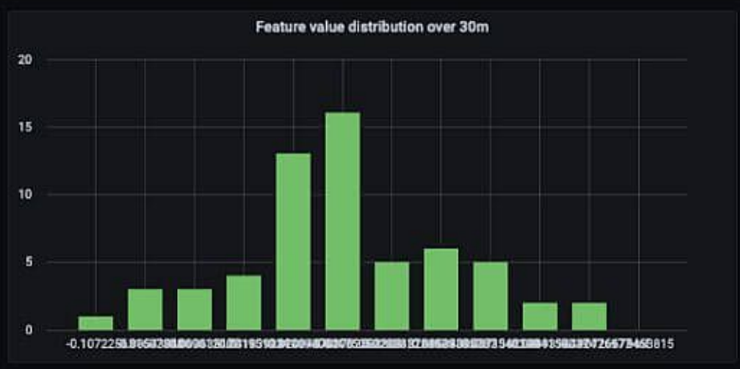
Feature value distribution over 30m

KL Divergence over 30m

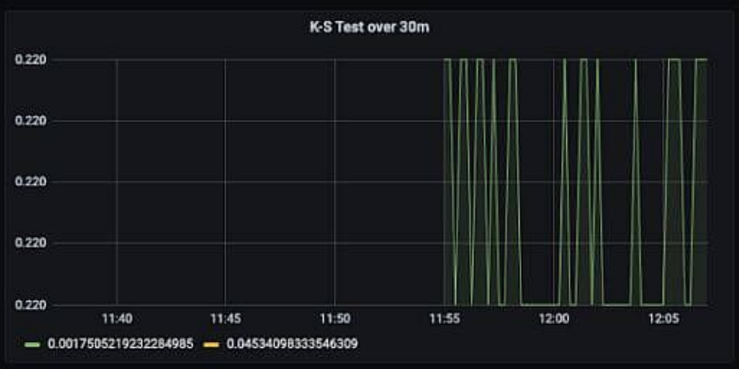
feature_0_value

Baseline distribution over training set

Feature value distribution over 30m

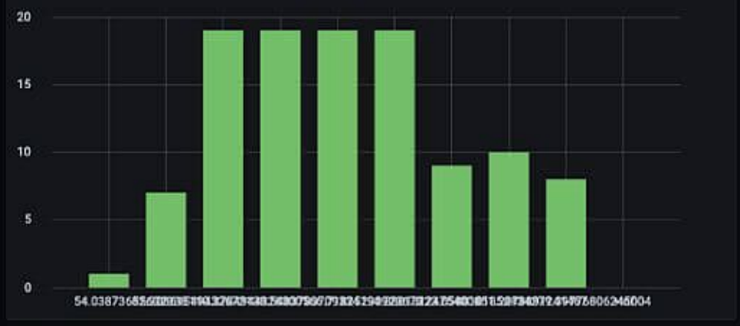


K-S Test over 30m

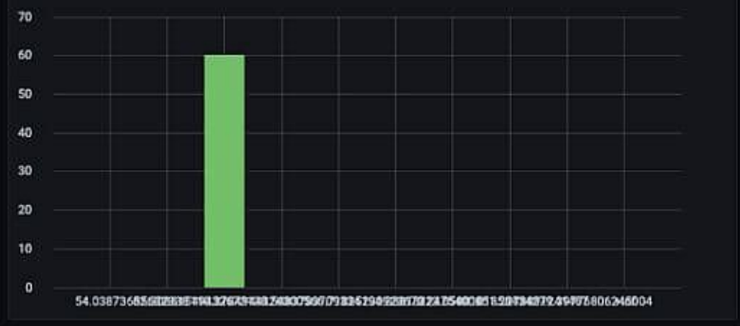


Regression

Baseline distribution over test set



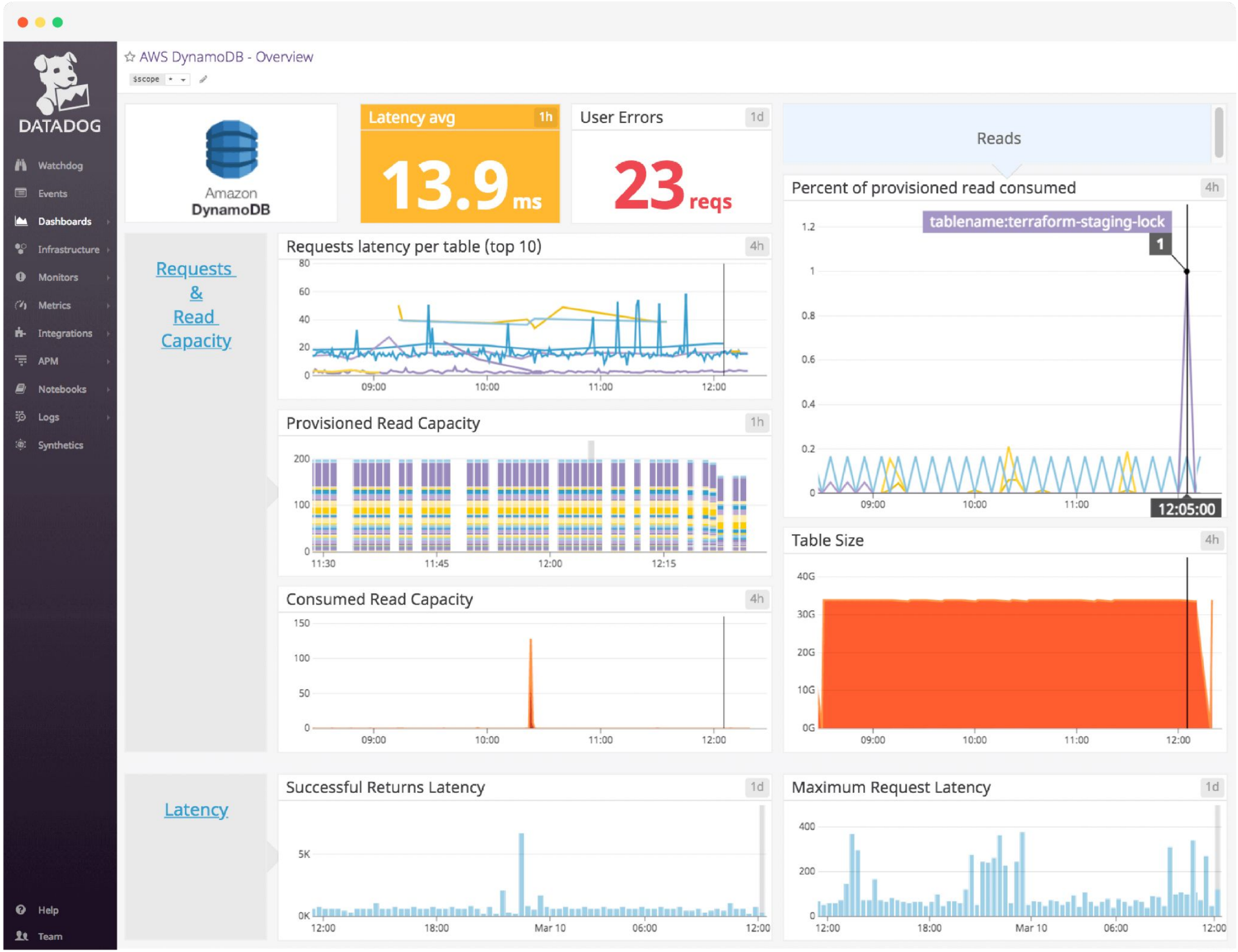
Inference value distribution over 30m



K-S Test over 30m

Classification *(3 panels)*

Ground truth *(1 panel)*


DATADOG
Watchdog
Events
Dashboards
Infrastructure
Monitors
Metrics
Integrations
APM
Notebooks
Logs
Synthetics
Help
Team
☆ AWS DynamoDB - Overview
$scope
Amazon DynamoDB
Latency avg 1h
13.9 ms
User Errors 1d
23 reqs
Reads
Requests & Read Capacity
Requests latency per table (top 10) 4h
80
60
40
20
0
09:00
10:00
11:00
12:00
Percent of provisioned read consumed 4h
tablename:terraform-staging-lock
1
1.2
1
0.8
0.6
0.4
0.2
0
09:00
10:00
11:00
12:05:00
Provisioned Read Capacity 1h
200
100
0
11:30
11:45
12:00
12:15
Table Size 4h
40G
30G
20G
10G
0G
09:00
10:00
11:00
12:00
Consumed Read Capacity 4h
150
100
50
0
09:00
10:00
11:00
12:00
Latency
Successful Returns Latency 1d
5K
0K
12:00
18:00
Mar 10
06:00
12:00
Maximum Request Latency 1d
400
200
0
12:00
18:00
Mar 10
06:00
12:00

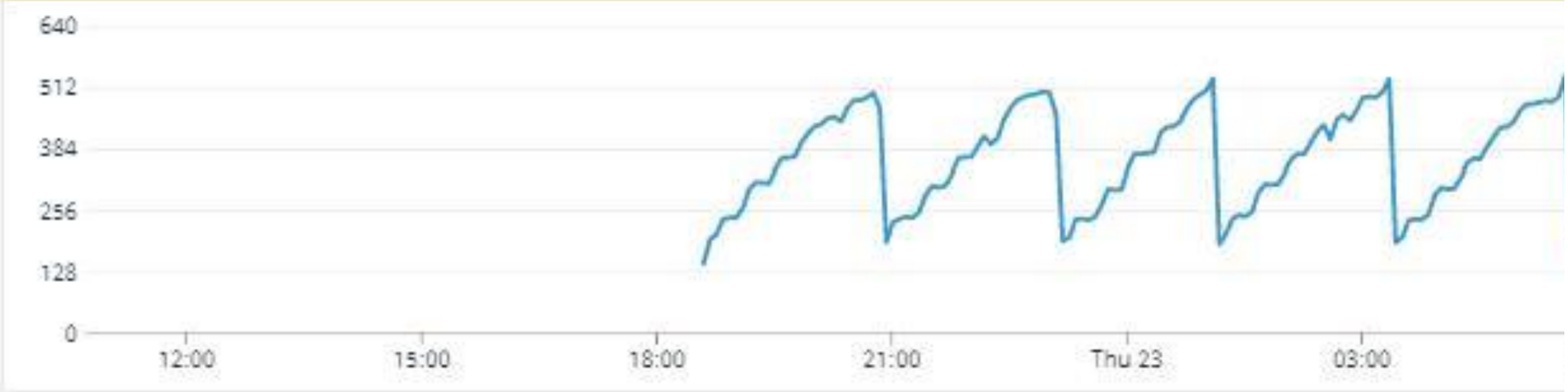
Ferramentas

TRADE-OFF

Orçamento

Requerimento de projeto

Antes



Depois

Resumo

- → Métricas de ML
- → Métricas de Aplicação
- → Ferramentas de Mercado
- → Caso Real

data expert.®

Data Science & Machine Learning

06. Planejando a Manutenção

Consultor: Murilo Mendonça

O que veremos nesta aula:

01

Manutenção

02

Repositório

03

CI/CD

Manutenção

Programada: Preciso retreinar o modelo de série temporal toda semana

Manutenção

Não-programada: Preciso retreinar meu modelo porque o negócio percebeu

Manutenção

Não-programada: Preciso retreinar meu modelo porque recebi um alerta

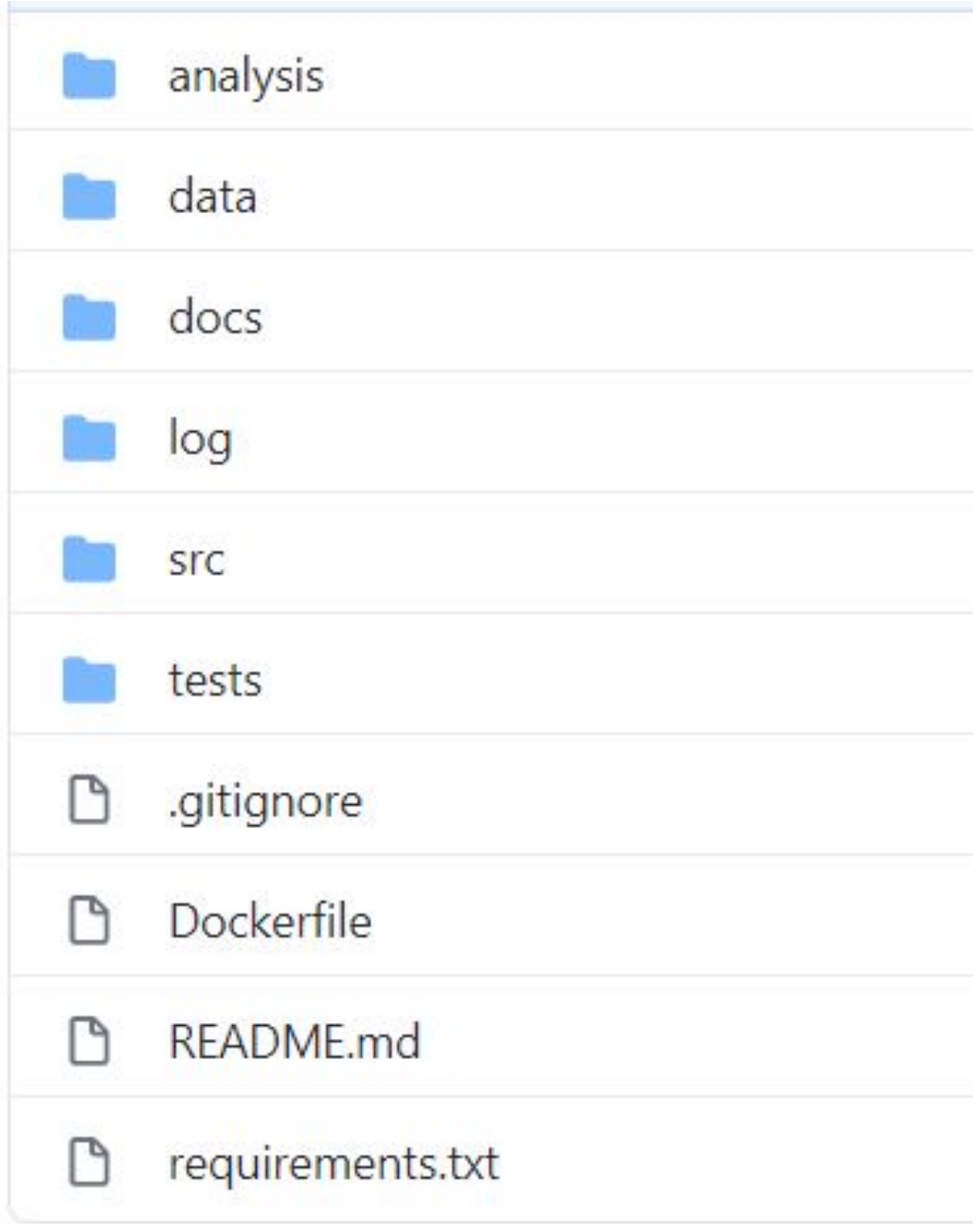
Manutenção

Programada: Vou subir uma nova versão da API com novas features

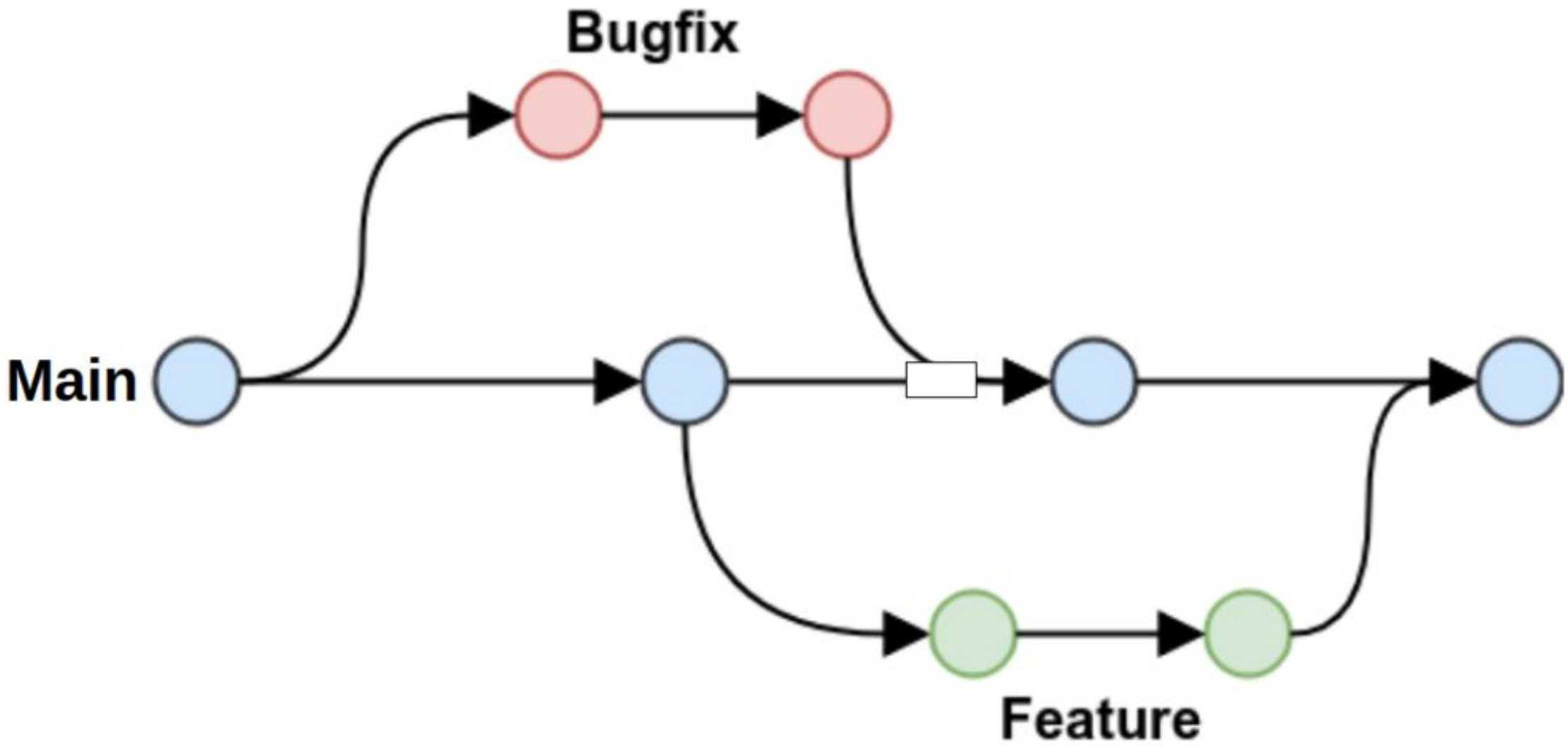
Manutenção

Não-Programada: Minha API está retornando erro 500 em 10% das vezes

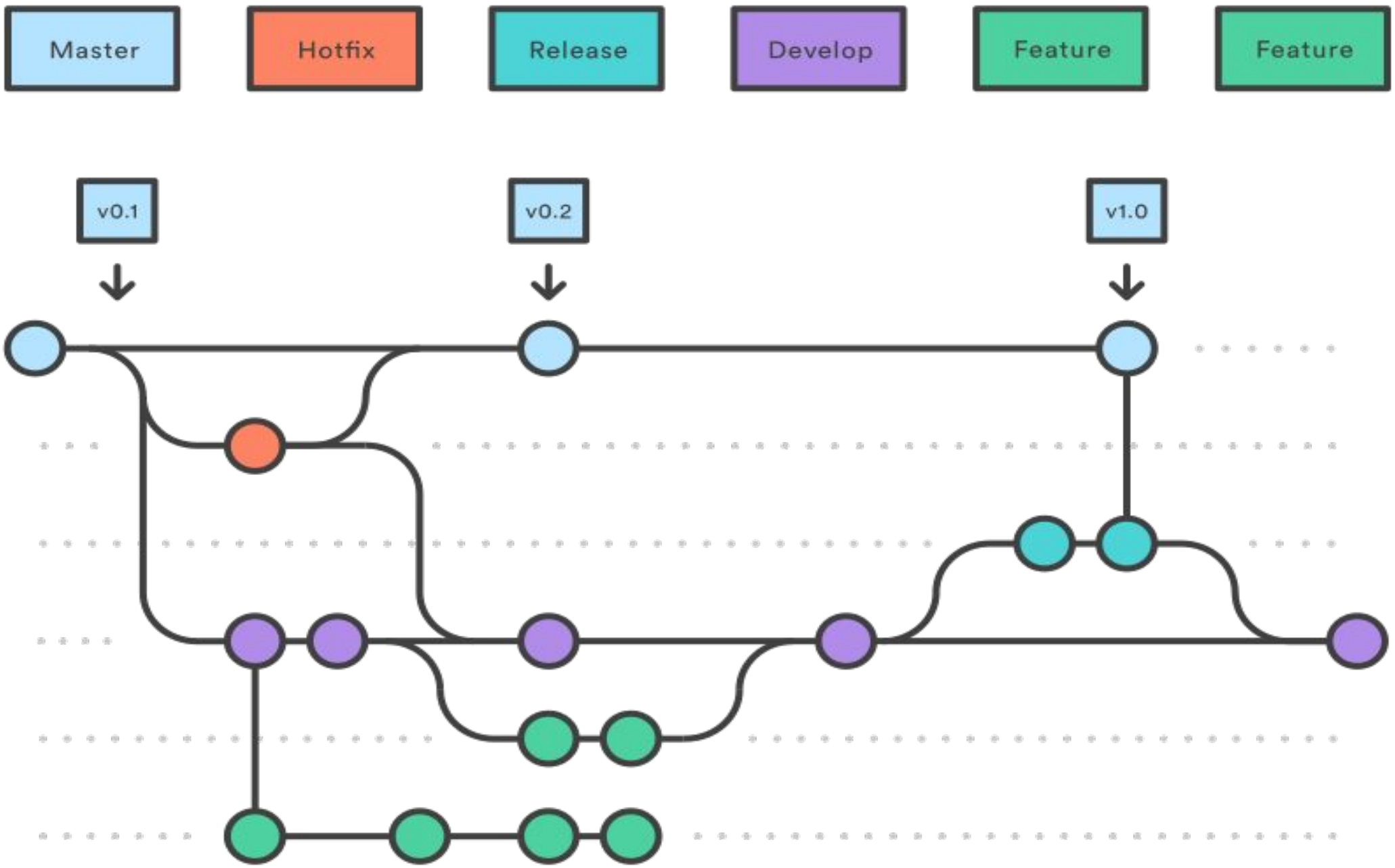
Repositório

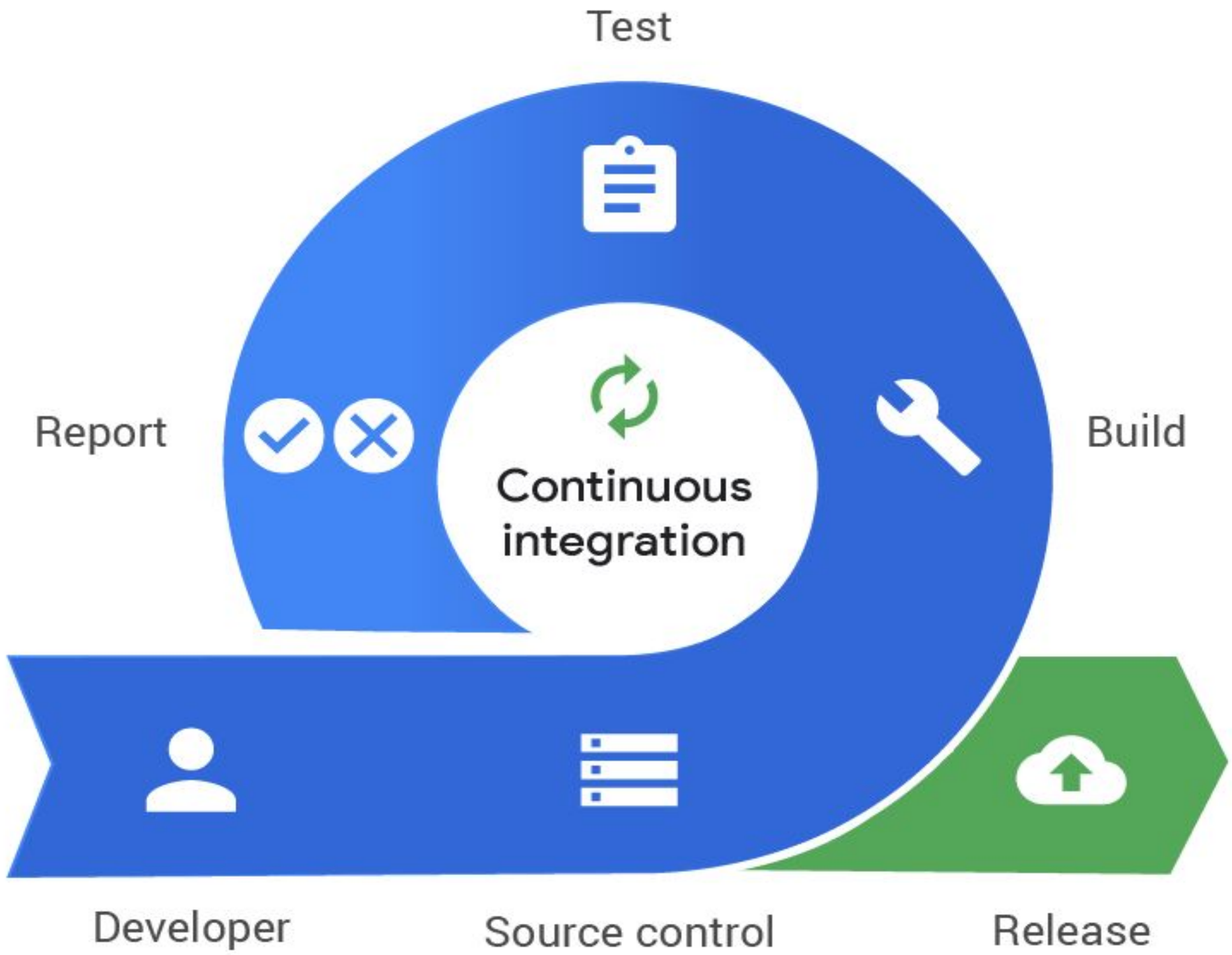


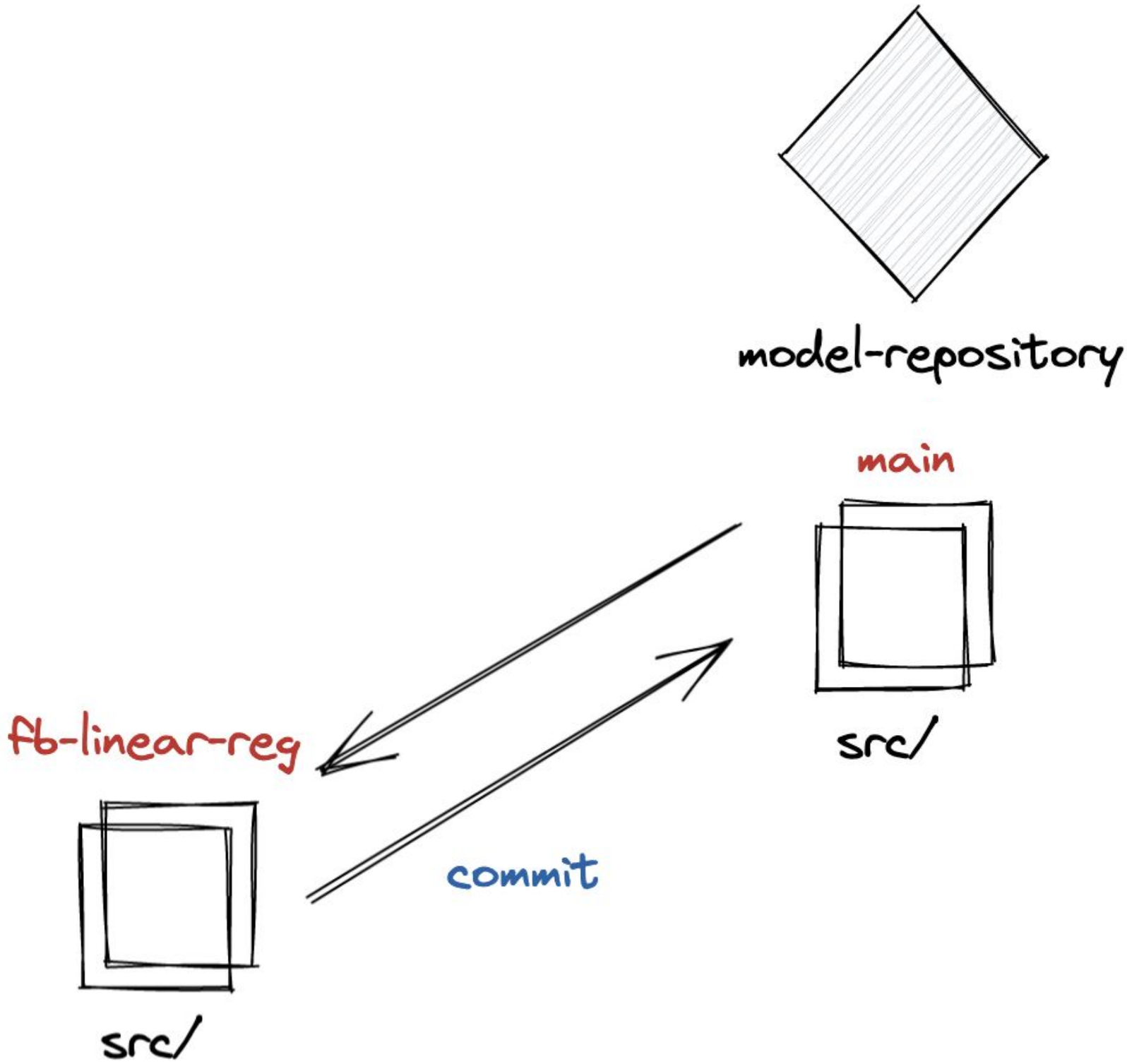
Repositório

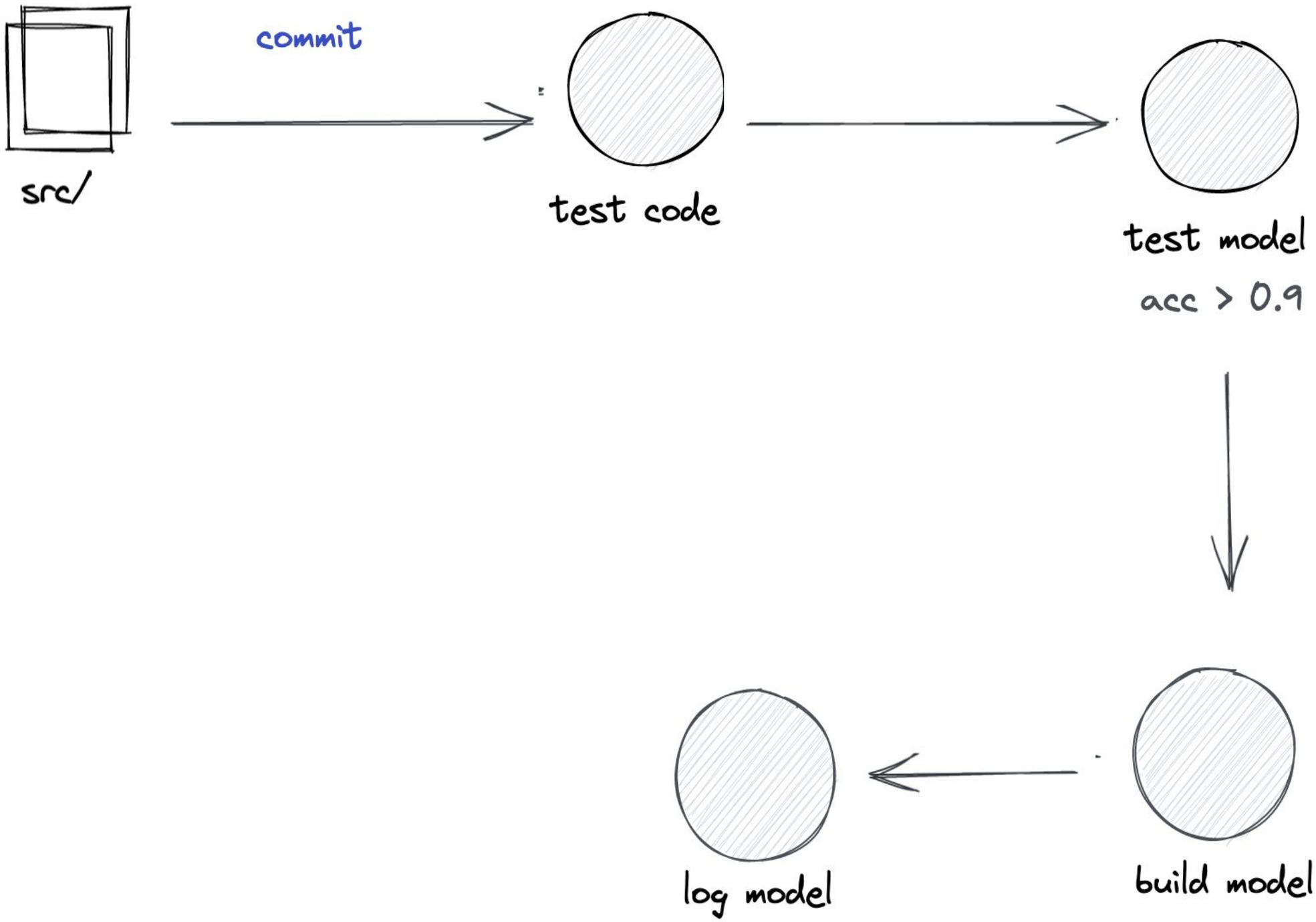


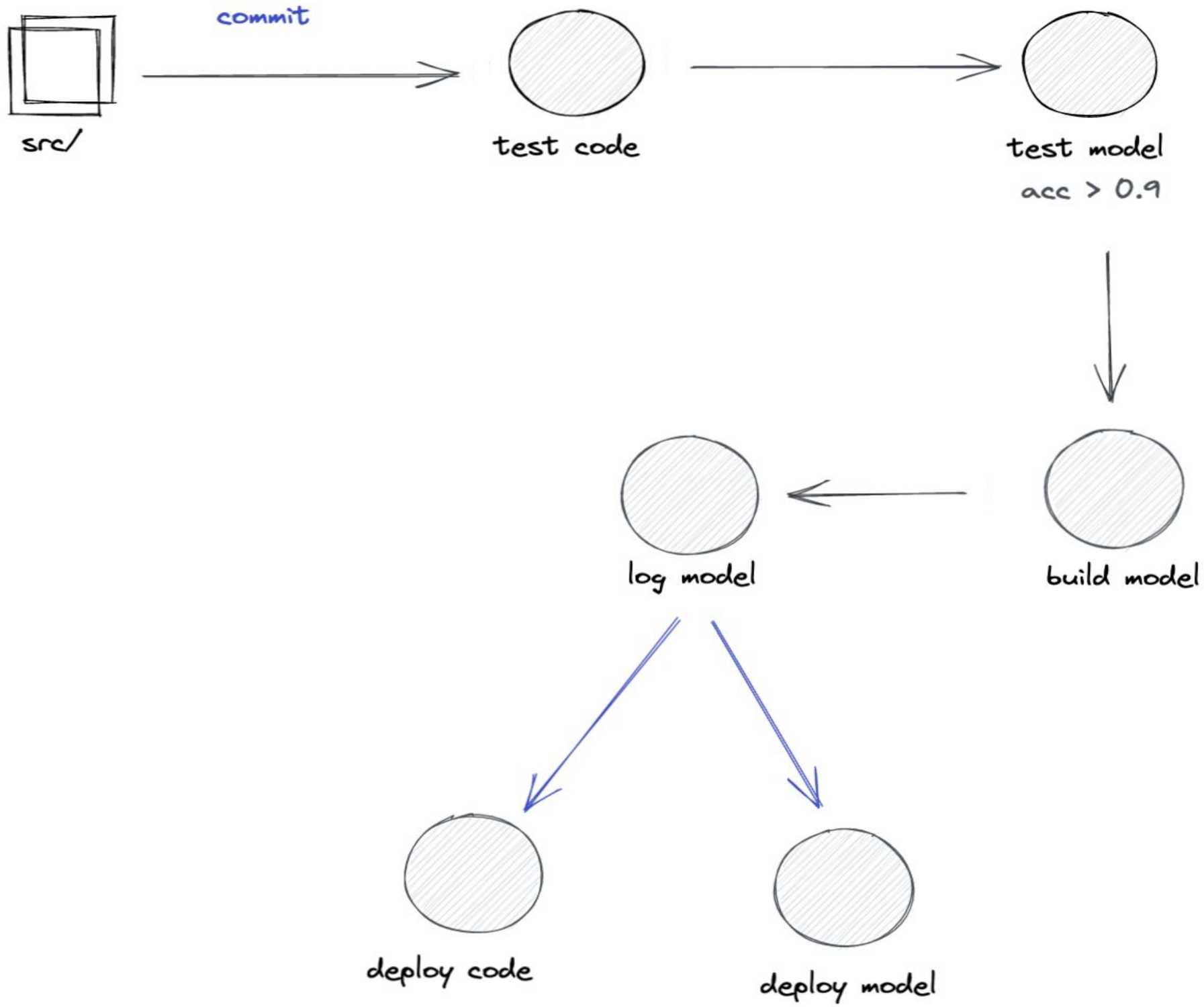
Repositório

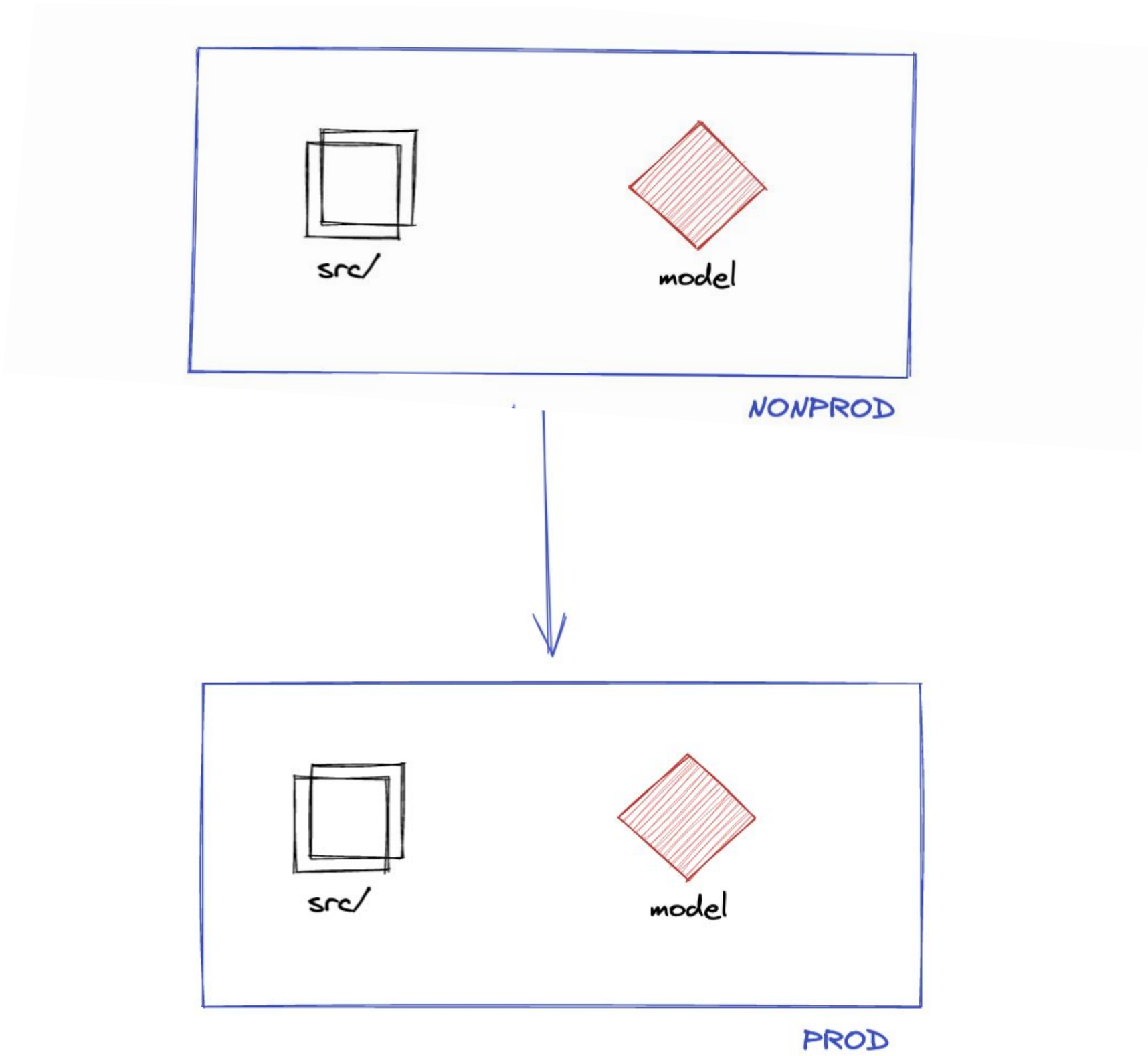



Test
Report
Build
Continuous integration
Developer
Source control
Release


model-repository
main
src/
fb-linear-reg
commit
src/


commit
src/
test code
test model
acc > 0.9
build model
log model


commit
src/
test code
test model
acc > 0.9
build model
log model
deploy code
deploy model


src/
model
NONPROD
src/
model
PROD

O que vimos nesta aula:

01

Manutenção

02

Repositório

03

CI/CD

Resumo

Manutenção

Repositório

Ci/CD

data expert.®

Data Science & Machine Learning

07. Report Final

Consultor: Murilo Mendonça

O que veremos nesta aula:

01

Definição

02

Checklist

03

Formatos e ferramentas

Report final

Documentar tudo.
Business understanding ao Deploy.

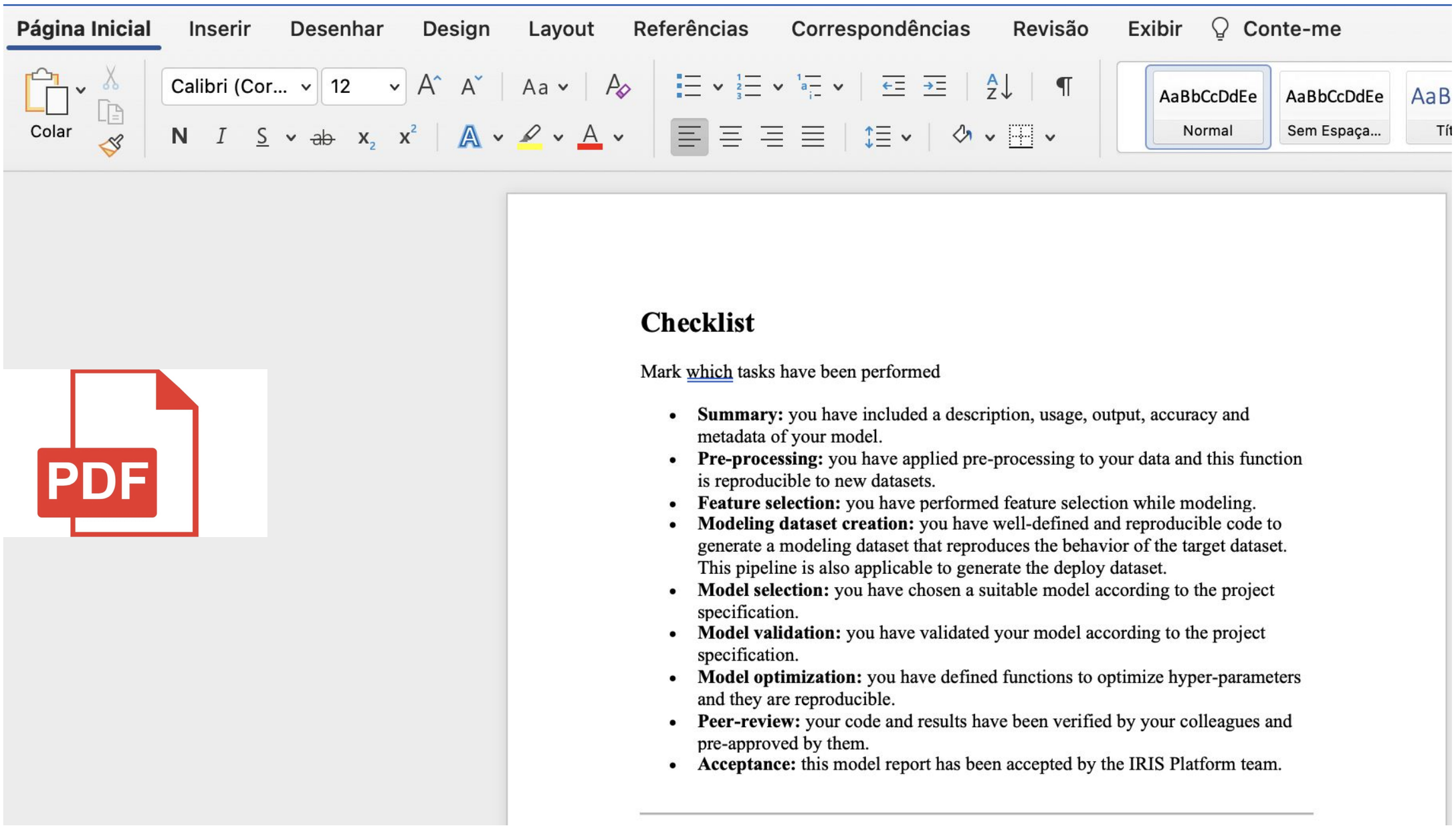
Quem contatar para dúvidas, melhorias ou problemas?

Tornar explícito como e qual o problema o modelo resolve.

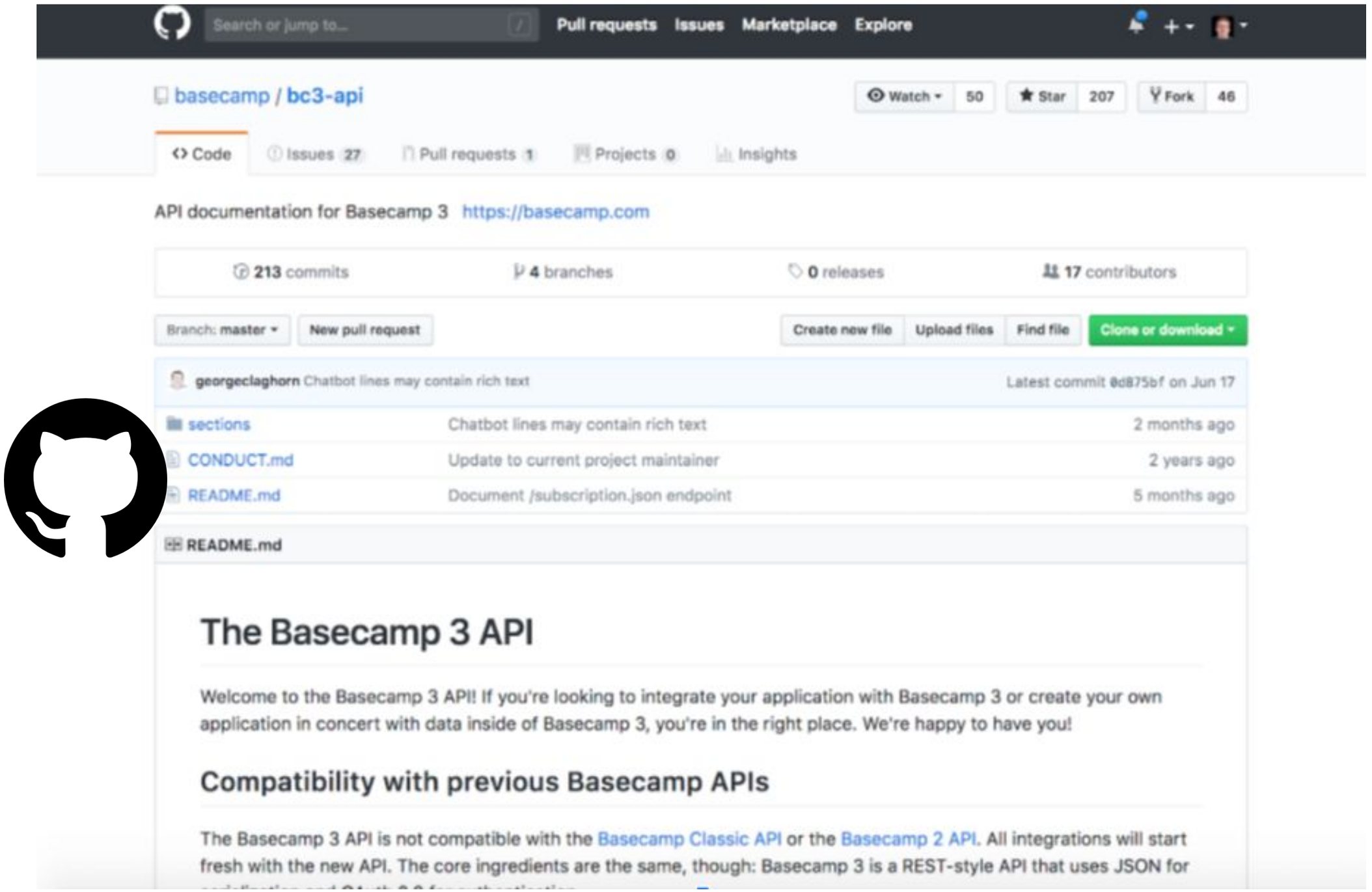
Checklist

- Pre-processing
- Feature selection
- Contracts
- Model selection
- Model validation
- Model optimization
- Model stage
- Drifting and retraining
- Deployment flavor
- Foreseen improvs.

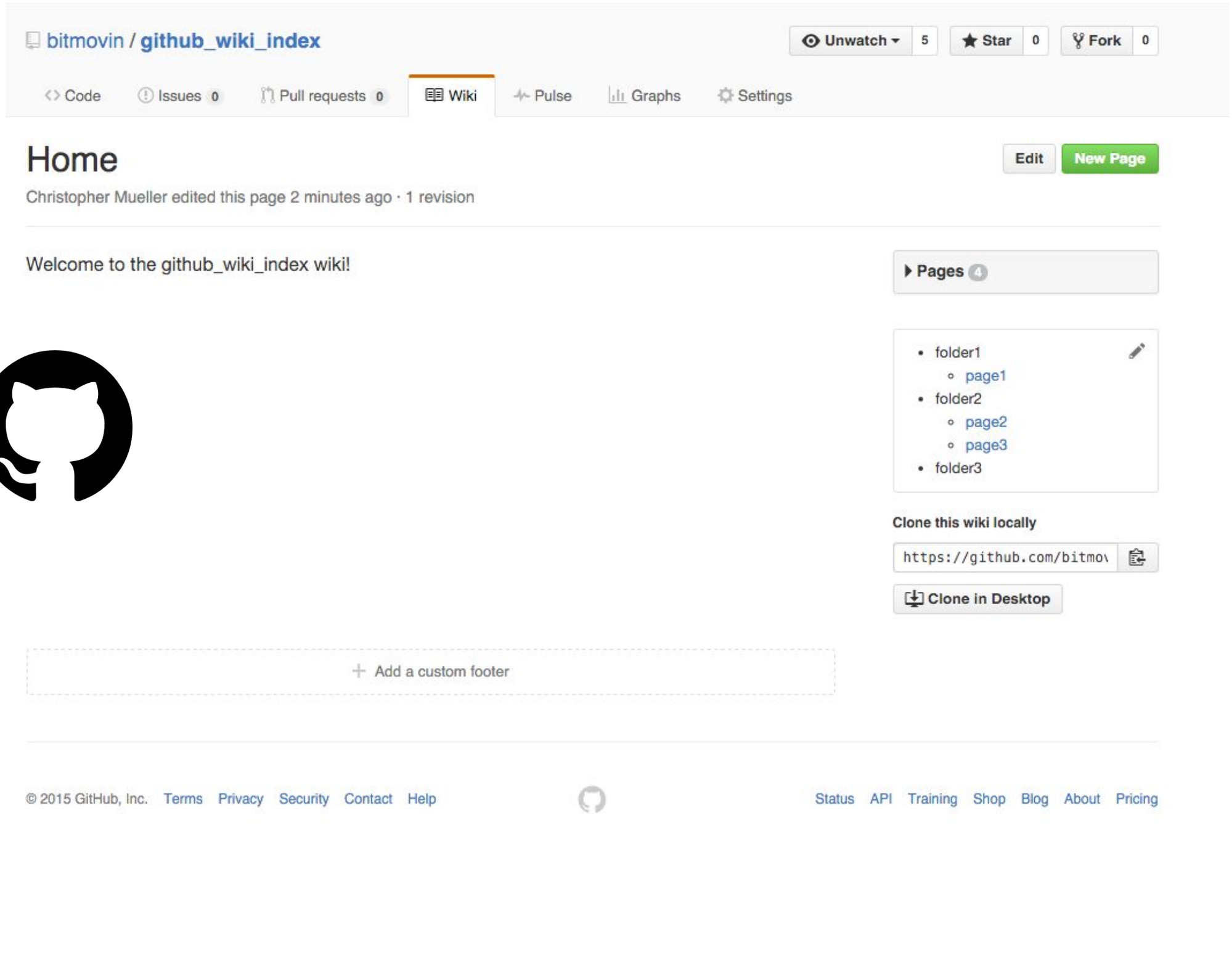
Formatos e Ferramentas



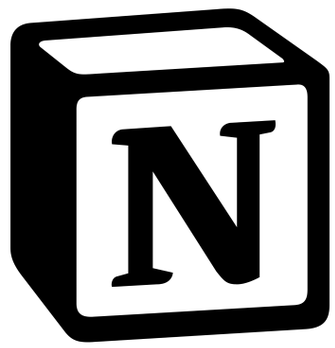
Formatos e Ferramentas



Formatos e Ferramentas



Formatos e Ferramentas



Resumo

- → Definição
- → Checklist
- → Formatos e Ferramentas

data expert.®

Data Science & Machine Learning

08. Apresentação Final

Consultor: Murilo Mendonça

O que veremos nesta aula:

01

O que não fazer

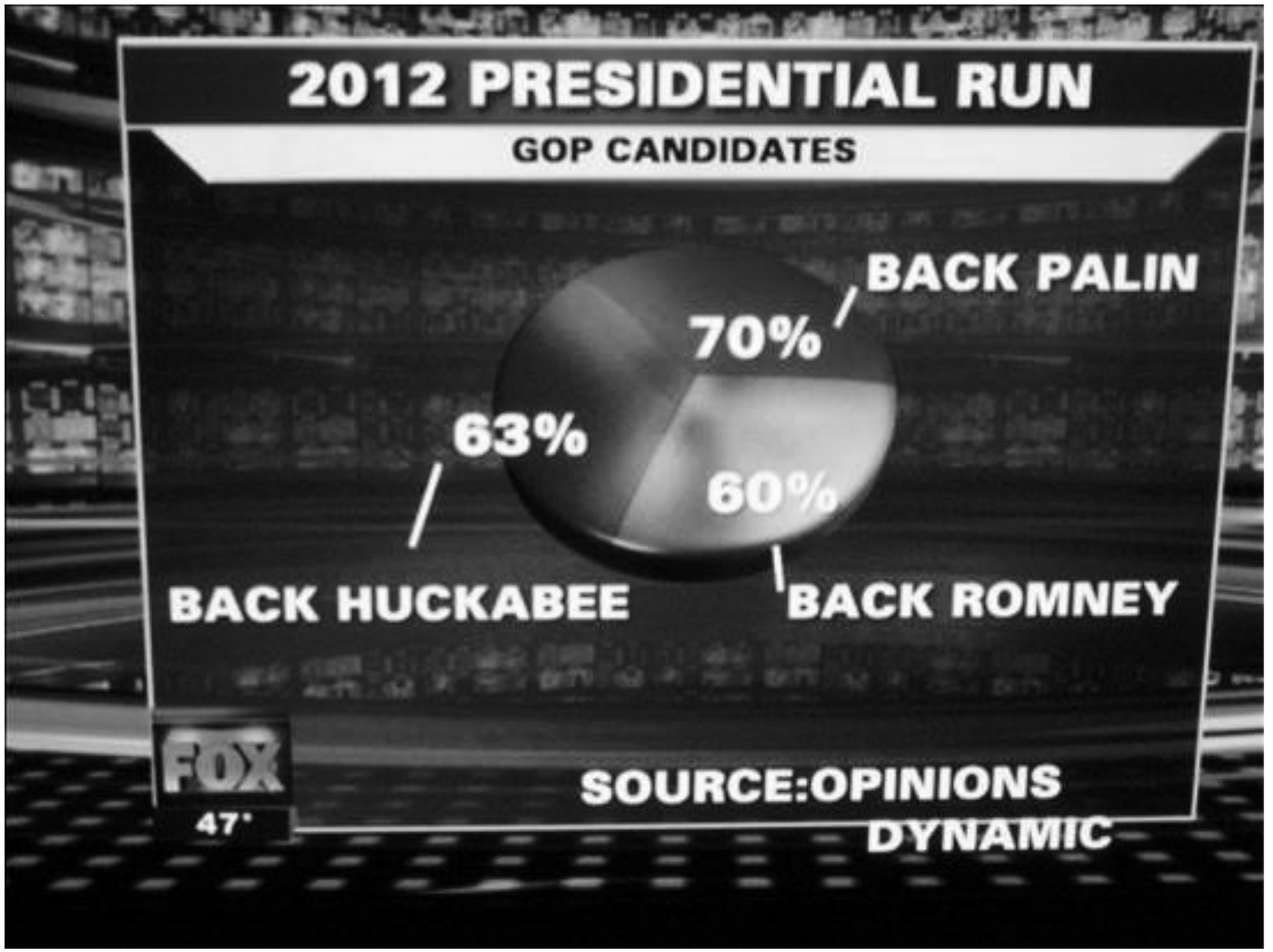
02

Design

03

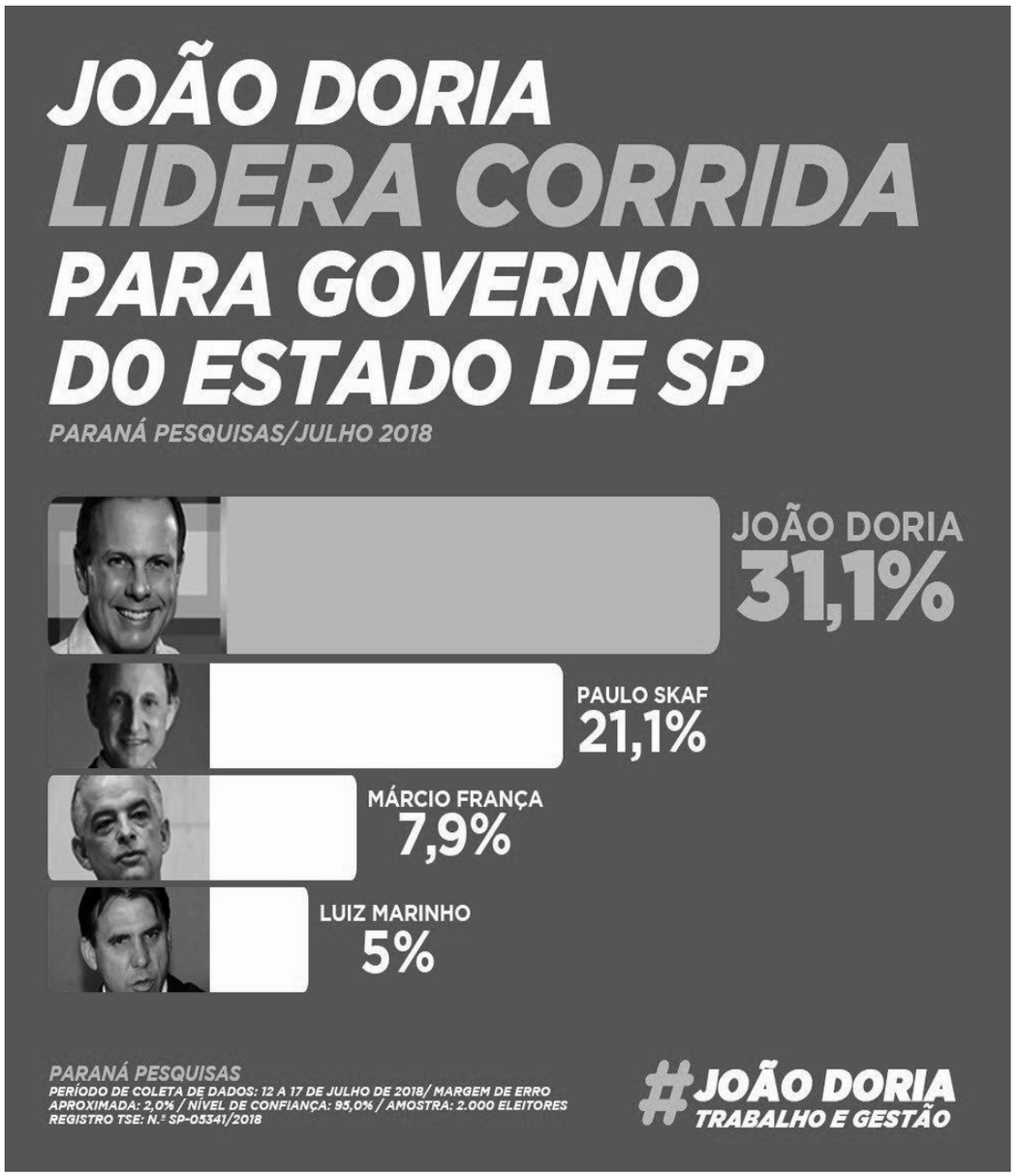
Storytelling

O que não fazer



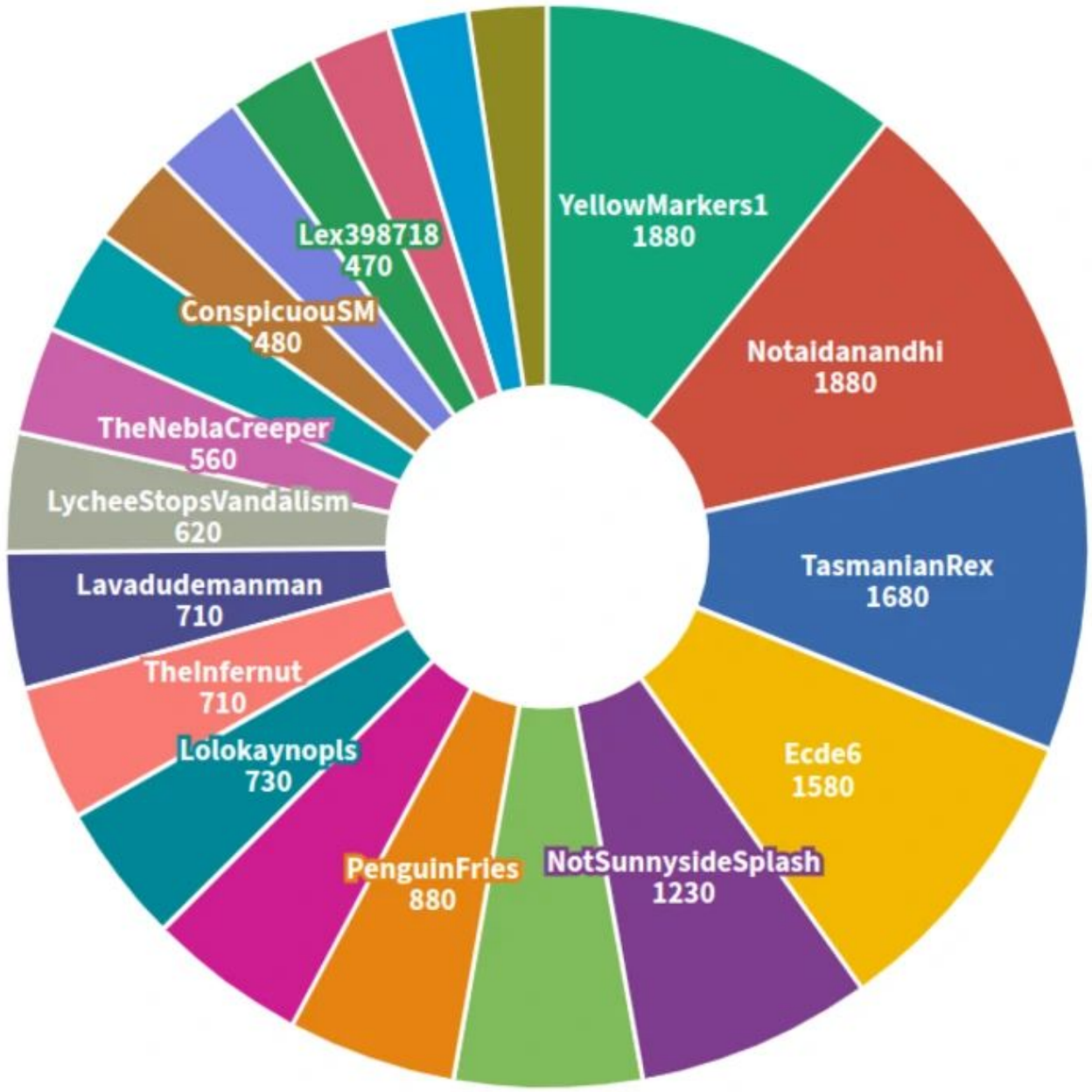
Google: "fox news data visualization"

O que não fazer



Google: "João Dória números"

O que não fazer



Google: "Pie Chart Hell"

Design

HEX #FFC20A
R 255 G 194 B 10

HEX #0C7BDC
R 12 G 123 B 220

HEX #994F00
R 153 G 79 B 0

HEX #006CD1
R 0 G 108 B 209

HEX #E1BE6A
R 225 G 190 B 106

HEX #40B0A6
R 64 G 176 B 166

HEX #E66100
R 230 G 97 B 0

HEX #5D3A9B
R 93 G 58 B 155

HEX #1AFF1A
R 26 G 255 B 26

HEX #4B0092
R 75 G 0 B 146

HEX #FEFE62
R 254 G 254 B 98

HEX #D35FB7
R 211 G 95 B 183

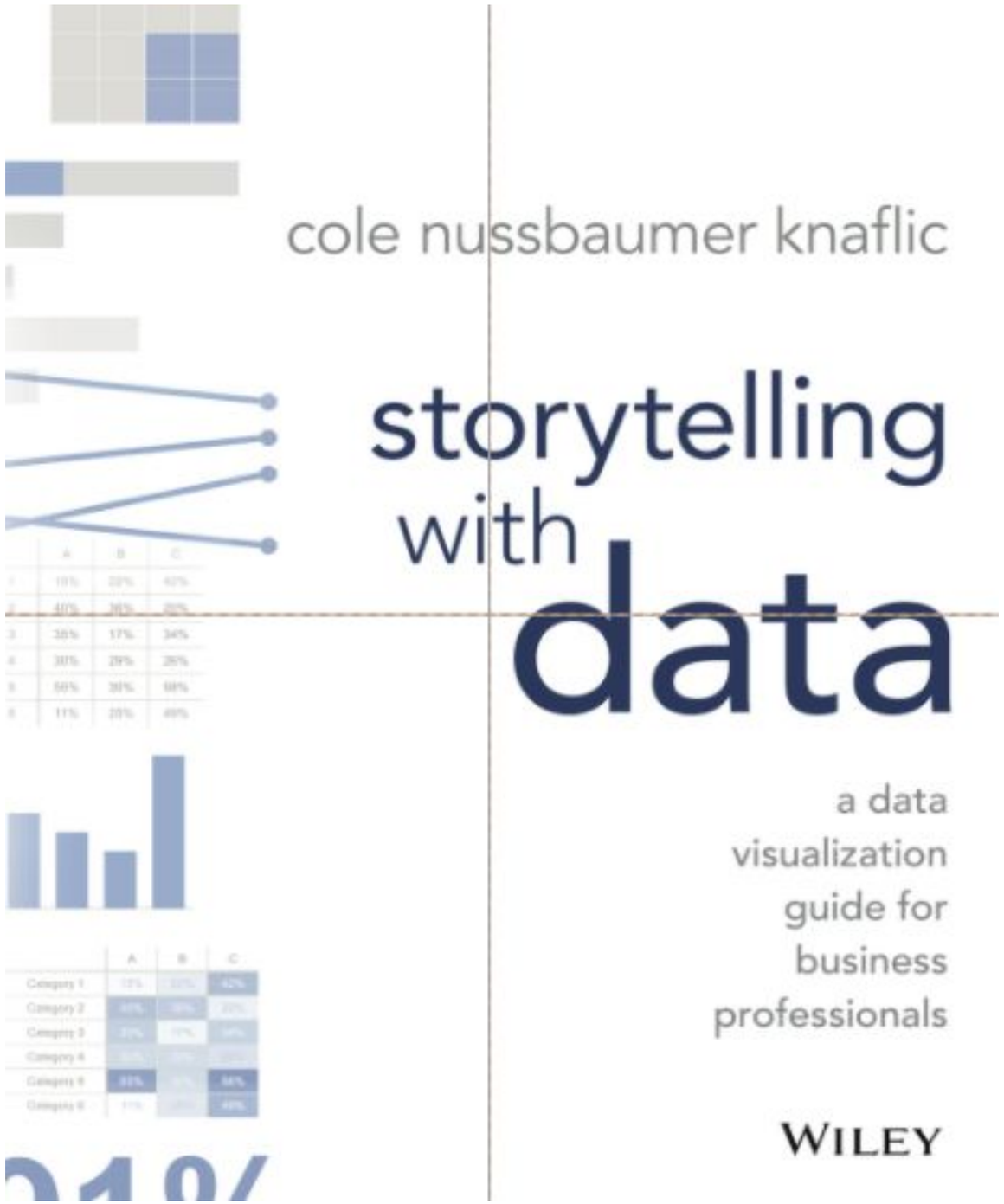
HEX #005AB5
R 0 G 90 B 181

HEX #DC3220
R 220 G 50 B 32

HEX #1A85FF
R 26 G 133 B 255

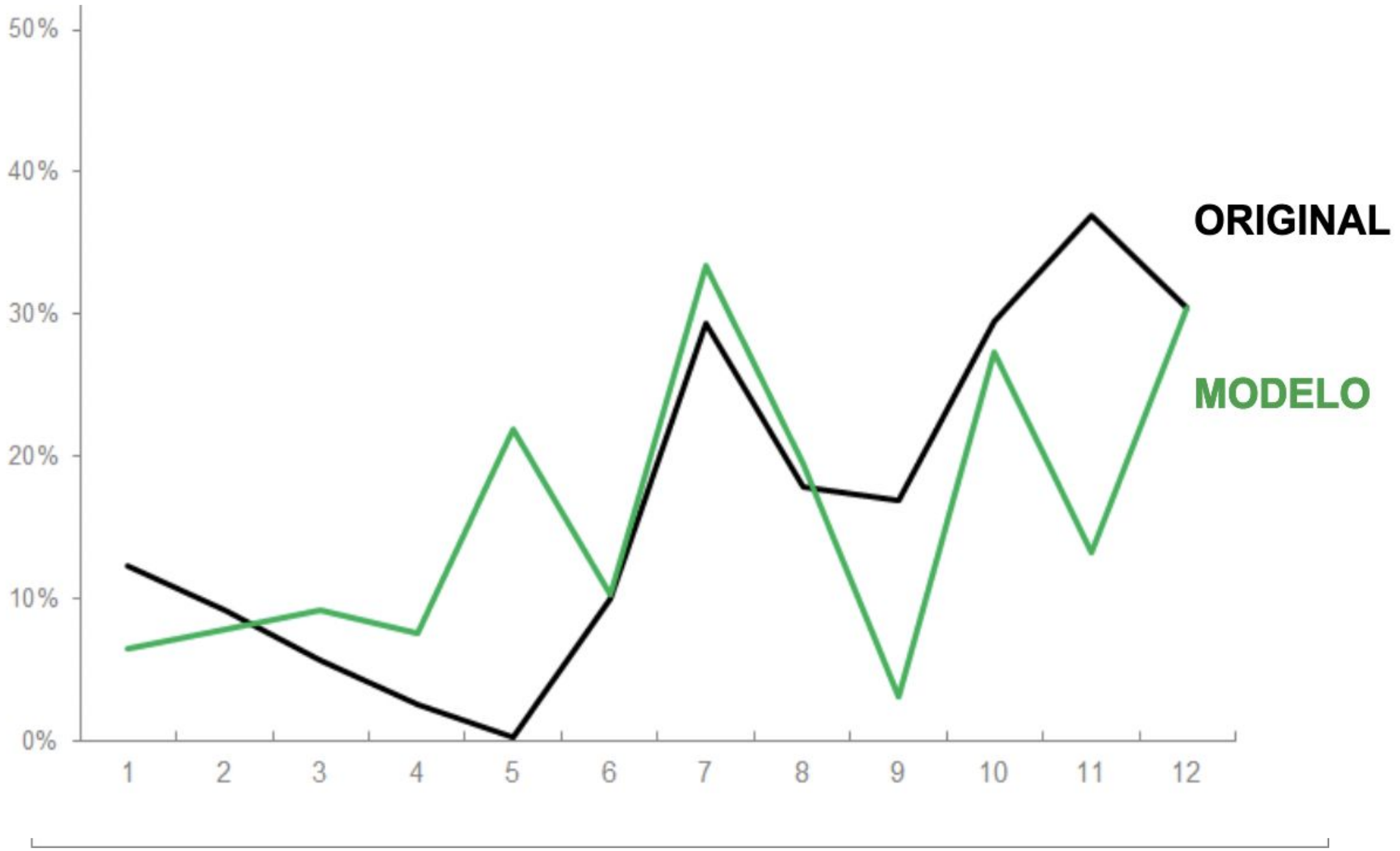
HEX #D41159
R 212 G 17 B 89

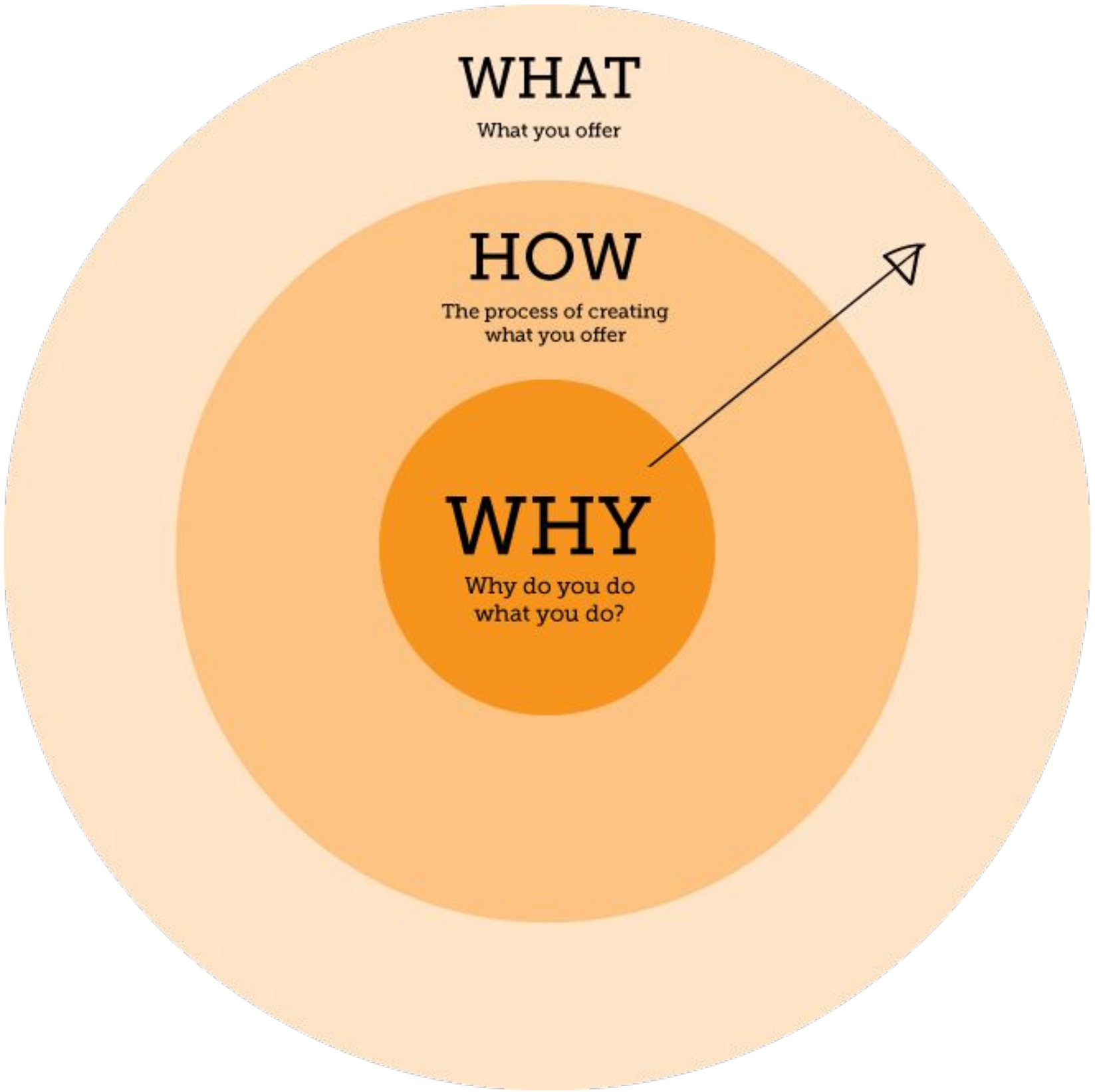
Design


cole nussbaumer knaflic
storytelling
with
data
a data visualization guide for business professionals
WILEY

Design

Design




WHAT
What you offer
HOW
The process of creating what you offer
WHY
Why do you do what you do?

Sem storytelling

Vamos criar um modelo de Regressão Logística com otimização de hiperparâmetros, usando a infraestrutura do time de plataforma para entregar uma previsão de fraude para o time de crédito.

Com storytelling e *design*

O time de crédito reportou um aumento de 50% de fraude no ano passado com a pandemia. Por isso, vamos criar um modelo com a infraestrutura de plataforma para rapidamente começar um MVP com uma regressão logística.

Com Storytelling e *Design*

Aumento de

50%

de Fraude

Plataforma **+**

Resumo

→ O que Não Fazer

→ Design

→ Storytelling

Resumo

Design

Storytelling

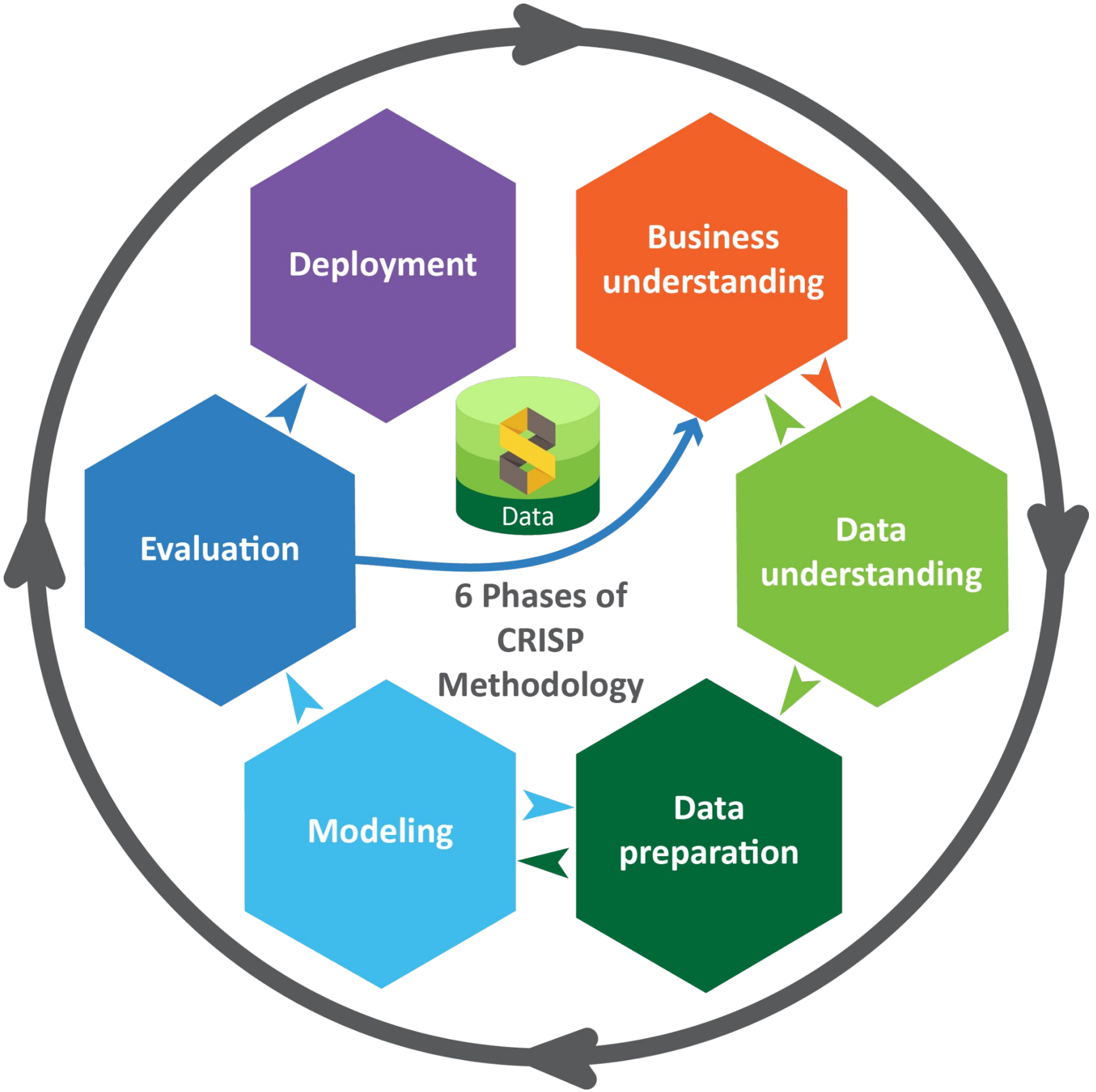
O que Não Fazer

data expert.®

Data Science & Machine Learning

11. Resumo do Módulo

Consultor: Murilo Mendonça


Deployment
Business understanding
Data
Evaluation
Data understanding
6 Phases of CRISP Methodology
Modeling
Data preparation

Resumo deste módulo:

01

Ambiente de produção

02

Planejando o deployment -Batch

03

Planejando o deployment - API

04

Planejando o monitoramento

05

Planejando a manutenção

06

Produzindo o report final

07

Produzindo uma apresentação final