

FIAP

NABA



COGNITIVE ENVIRONMENTS

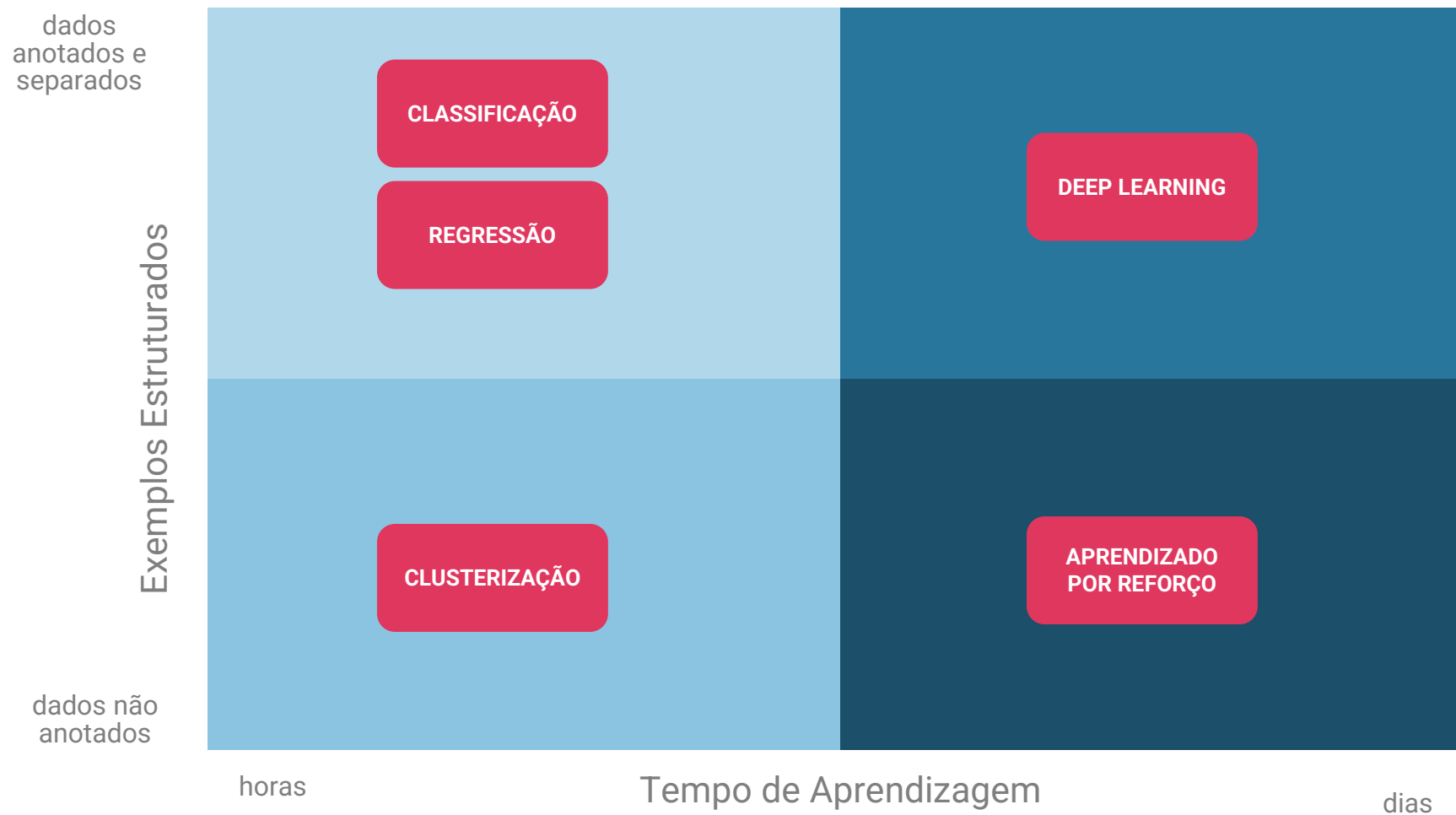
DATA SCIENCE & AI MBA



INTRODUÇÃO A PLATAFORMAS COGNITIVAS



POR QUE UTILIZAR **PLATAFORMAS?**



MUNDO MODERNO **DE PROCESSAMENTO DE DADOS**

DADOS ORGANIZADOS E CURADOS

Dados abundantes e padronizados.

Livros digitais.

Bibliotecas de fotos.

Devices & IOT.

CUSTOS DE PROCESSAMENTO MAIS BAIXOS

Custo de armazenamento e processamento cada vez mais baixos.

SoC, arquitetura ARM.

Tecnologia SSD mais acessível.

DIMINUIÇÃO DA DISTÂNCIA ENTRE TEORIA E PRÁTICA

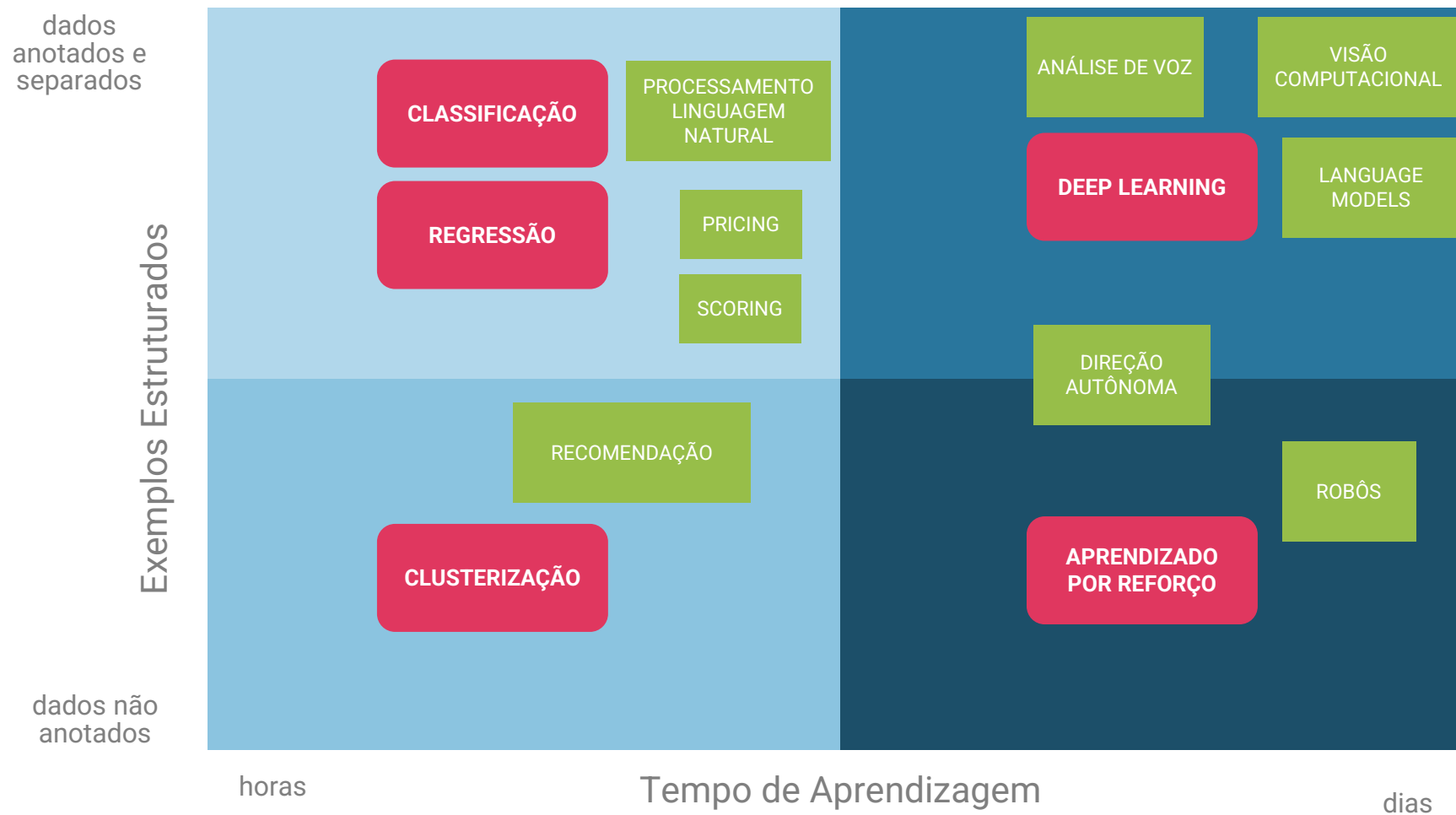
Big Techs investindo em times de pesquisadores e abrindo iniciativas Open Source.

TensorFlow (Google), Llama 3 (Meta).

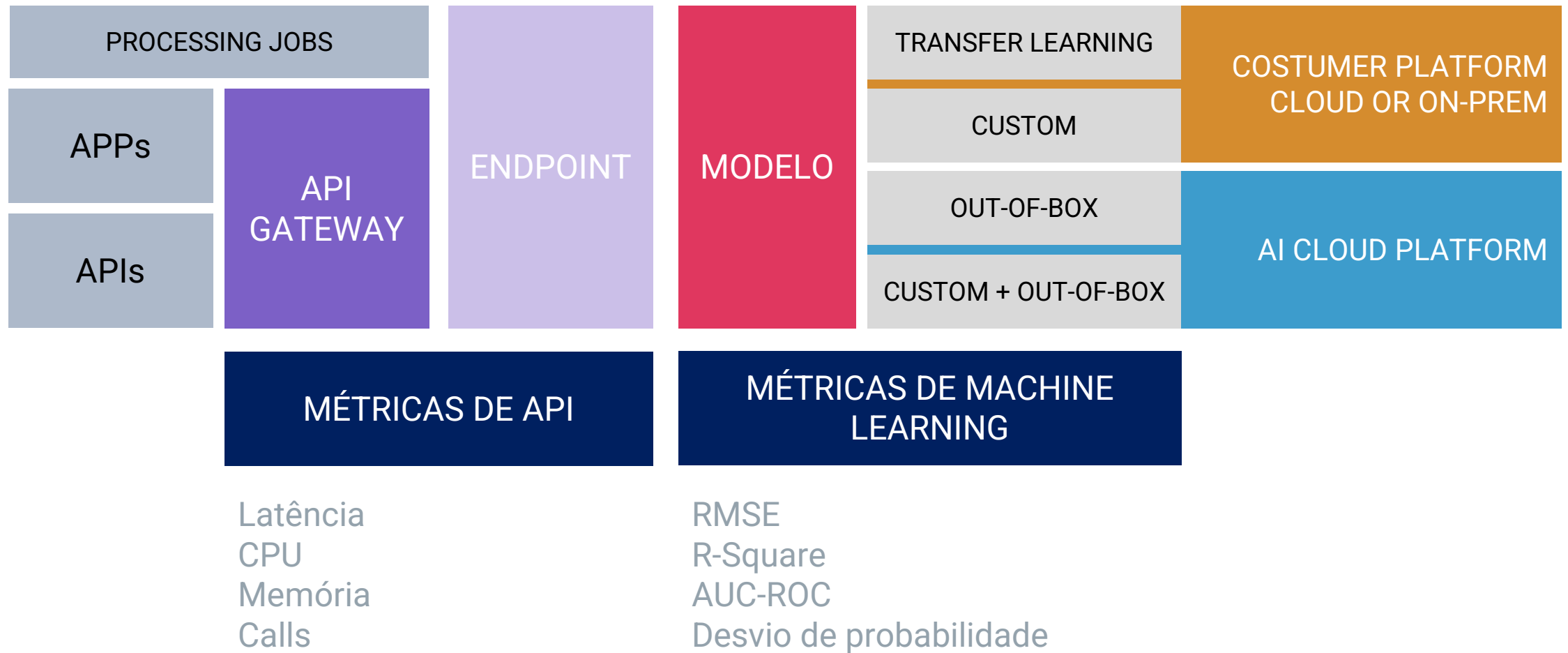
PLATAFORMAS COM UX MAIS ACESSÍVEIS

Provedores de cloud trazendo eficiências de escala e experiência de usuário.

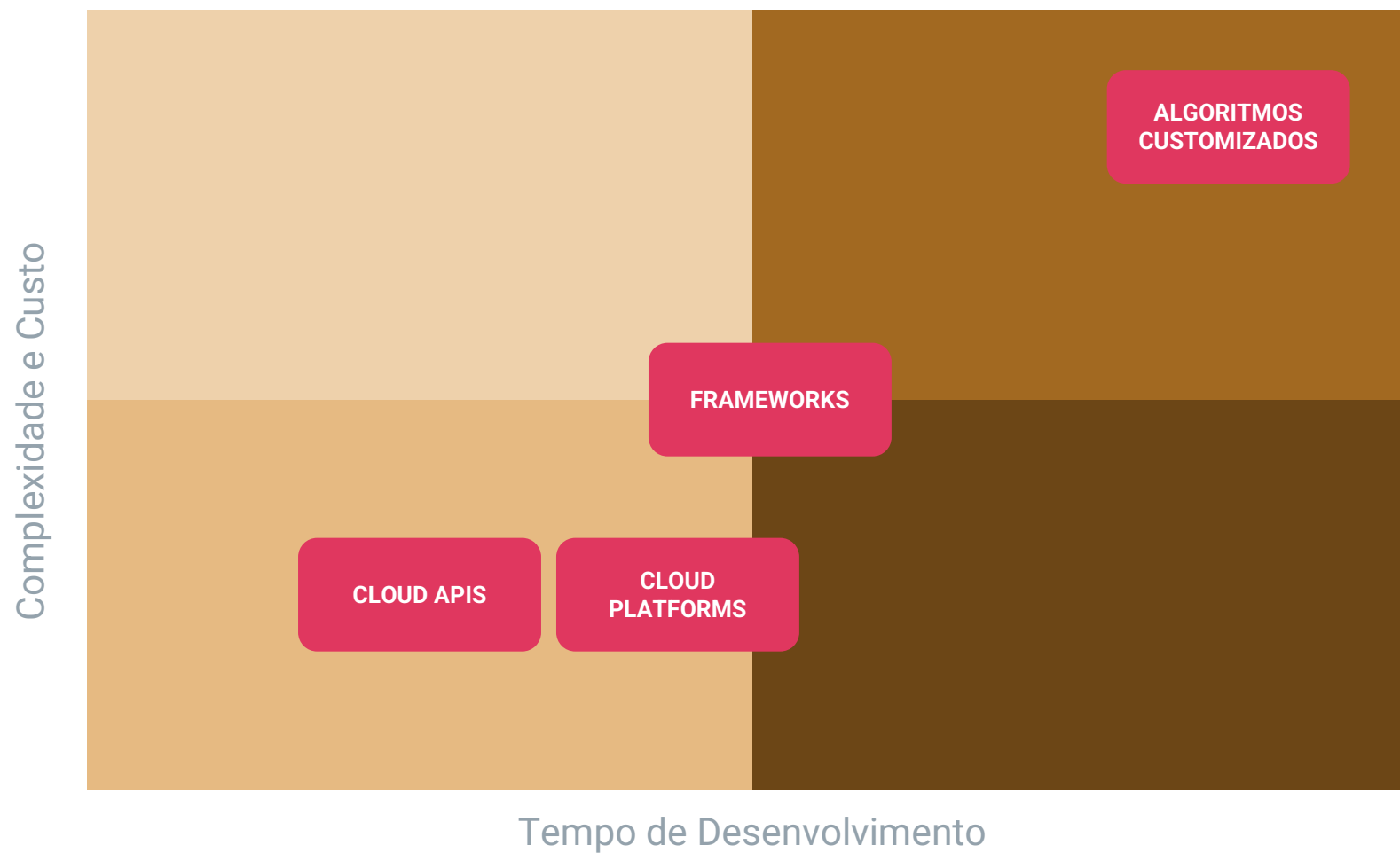
POR QUE UTILIZAR **PLATAFORMAS?**



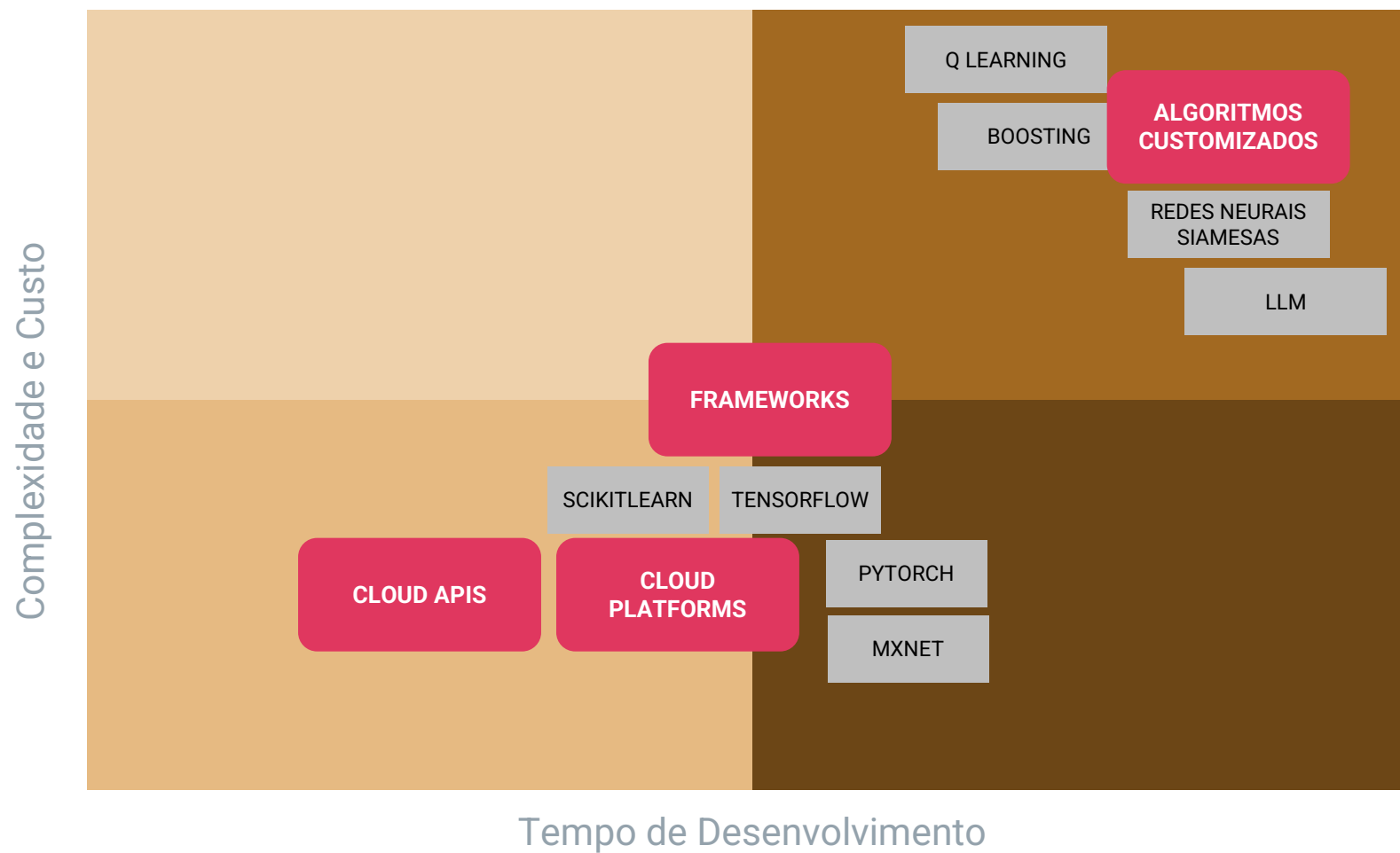
ANATOMIA DE UM **AI PRODUCT**



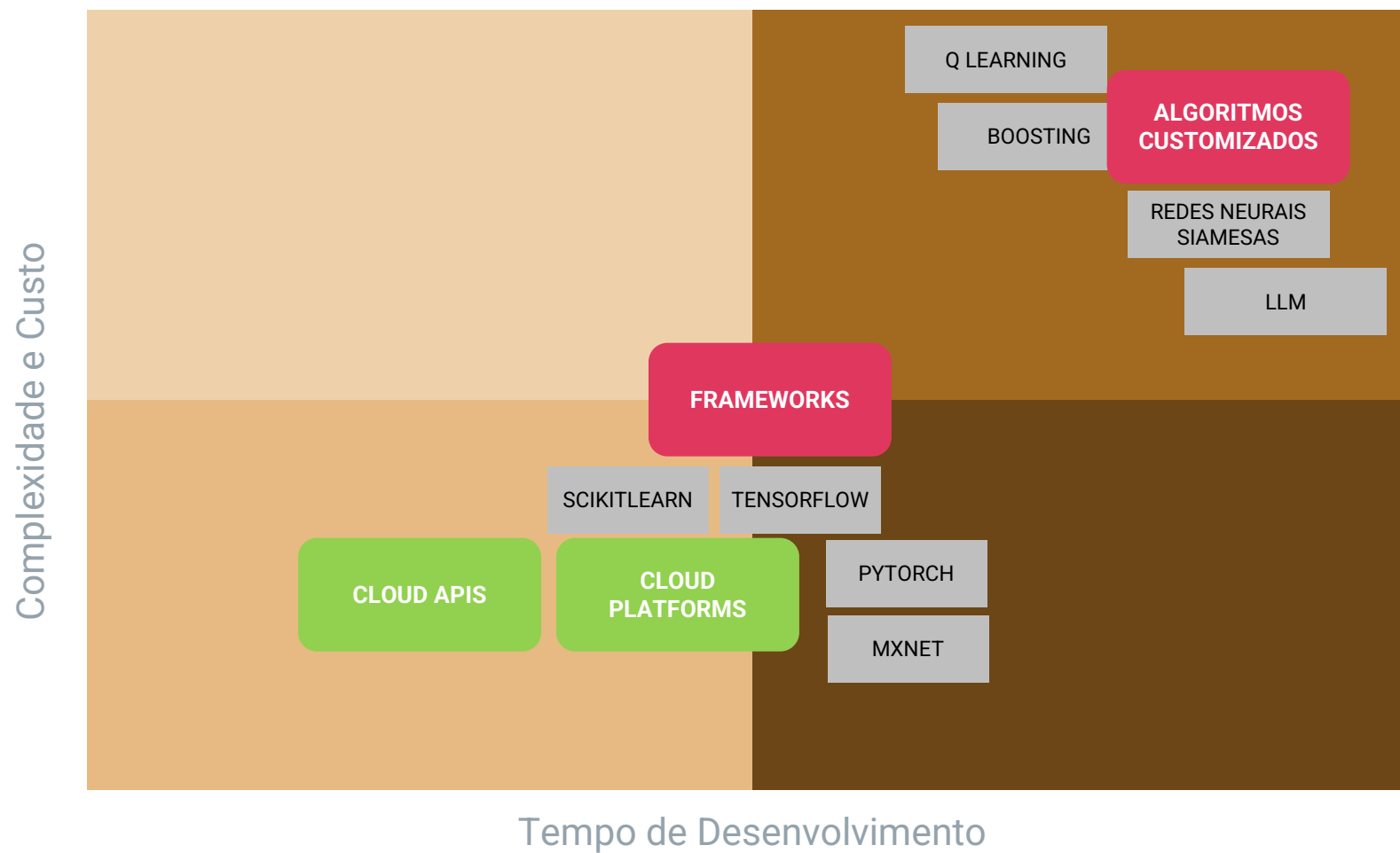
POR QUE UTILIZAR **PLATAFORMAS**?



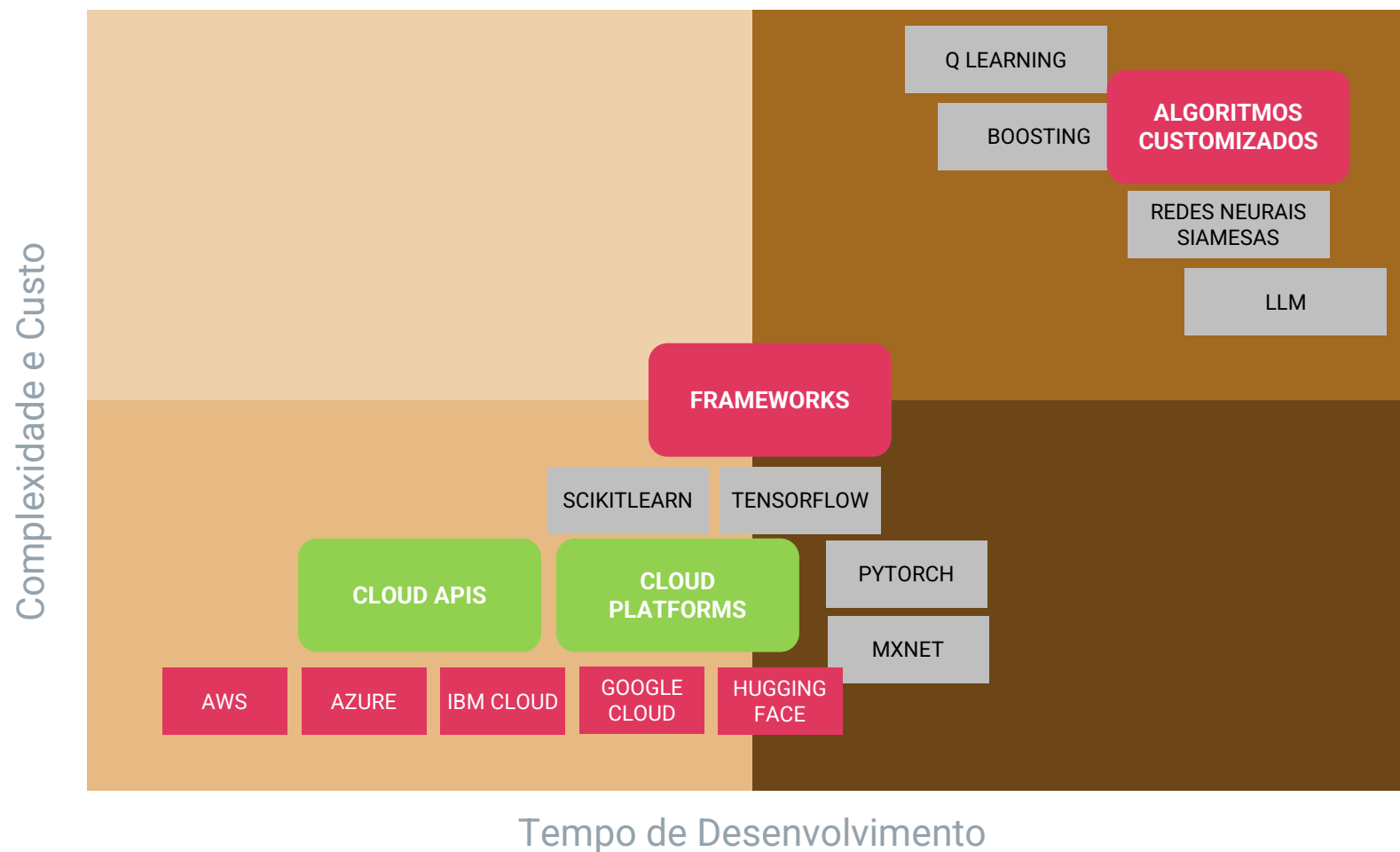
POR QUE UTILIZAR **PLATAFORMAS**?



POR QUE UTILIZAR **PLATAFORMAS**?



POR QUE UTILIZAR **PLATAFORMAS**?



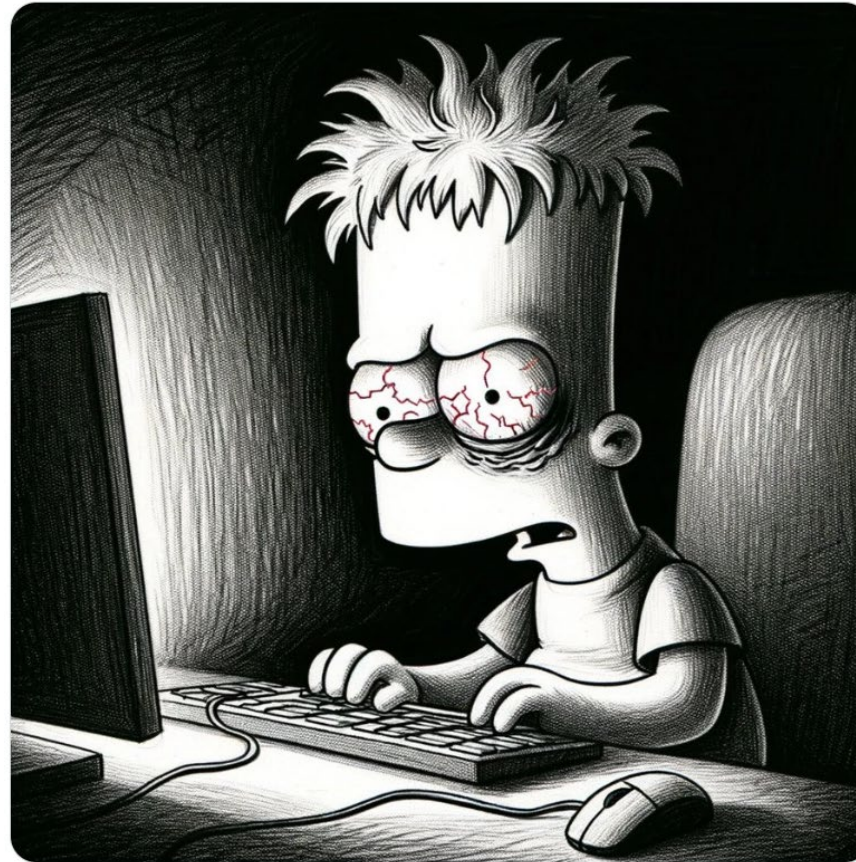
OU É PARA EVITAR...



anton  @abacaj · 8 de out

...

I love training small models, they are so much easier, faster, and cheaper to train. Just train a bunch of them, you don't need to use those big private models bro



 45

 64

 1.093

 168,2 mil



PLATAFORMAS DE CLOUD

O quadrante mágico para Cloud AI Developer Services destaca o impacto de cada provedor de cloud no mercado.

Dentre os provedores, podemos destacar os 4 maiores no segundo quadrante, que define como líderes e visionários e que basicamente são os grandes provedores de serviços cloud.

Os critérios utilizados para compor o quadrante mágico são:

- Estratégia go-to-market focada em Cloud AI Developer Services
- Capacidades centrais voltadas a serviços de Auto ML, serviços de idioma e serviços de visão.

Figure 1: Magic Quadrant for Cloud AI Developer Services







OFERTA AMPLA DE SERVIÇOS

CLOUD APIS

Amazon Comprehend
Amazon Poly
Amazon Rekognition
Amazon Textextract
Amazon Transcribe
Amazon Translate

PRONTOS PARA
USO

DEVELOPMENT BASED

Amazon Sagemaker
Studio
StudioLab
JumpStart

EXEMPLOS

TIPO DO MODELO

FINE TUNING

FULLY MANAGED

Amazon Forecast
Amazon Personalize
Amazon Fraud Detector

TUNING



Generalistas

Específicos

SAGEMAKER CANVAS

É a versão mais simples para construir modelos. Seu foco principal é atender um novo perfil de generalistas que não querem entrar a fundo na construção de modelos, no entanto utilizar a tecnologia para ganhos relevantes, no contexto de Citizen AI.

Possui diversos modelos pré-treinados para análises comuns como análise sentimental, leitor de OCR, etc.

Esta plataforma se utiliza da tecnologia AutoML da AWS.

My models / Appointment No Show / Version 2

Select

Build

Analyze

Predict

Model status

Accuracy ⓘ Optimization metric

80.602%

The model predicts the correct Appointment 80.602% of the time. ⓘ

Overview

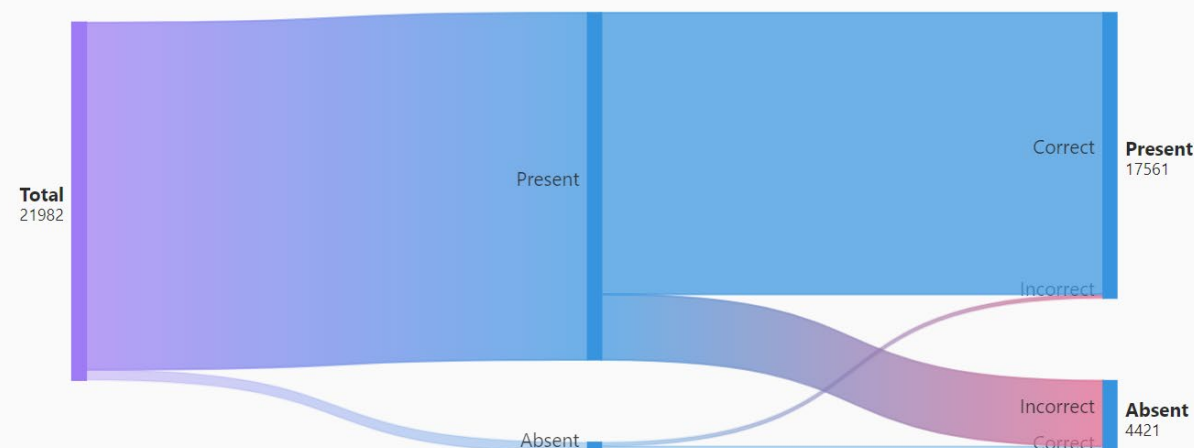
Scoring

Predicted vs. Actual

All predictions

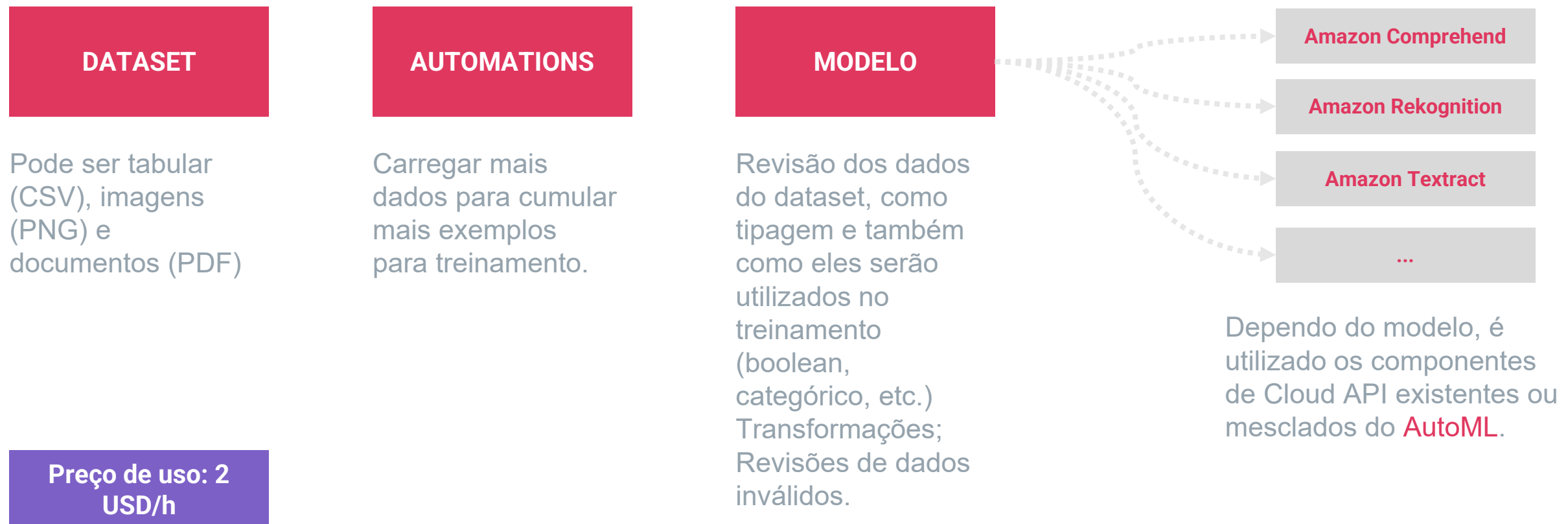
Predicted

Actual



SAGEMAKER CANVAS

Dois passos são necessários para começar a utilizar o treinamento de modelos. O primeiro é carregar um dataset. Após o carregamento não temos a opção de revisar os dados, isso é feito na construção do modelo.





Vamos construir um modelo para prever quando um paciente não comparece a uma consulta marcada.

Vamos utilizar este dataset do Kaggle: ariks90/cleaned-brazil-may2016-noshow-medical-appointments.

Depois de carregar o arquivo de treinamento, no nosso caso tabular, vamos criar um modelo novo e selecionar o dataset carregado.

Para maximizar os dados de treinamento, precisamos ampliar as características pelos dados existentes e remover uma coluna que não terá impacto no treinamento.

1. Remover coluna de ID dos pacientes.

No “Column View” desmarcar a coluna.

2. Adicionar dados adicionais a data de agendamento, como dia da semana,

No “Extract” extrair os dados adicionais da coluna data de agendamento: mês, dia, hora, dia da semana, quarter e semana do ano.

3. Inclua como “Target column” o atributo “appointment”.

4. Inicie o “Standardt Build”, que tem como objetivo buscar a Precisão. Em “Advanced settings” é possível alterar a métrica de objetivo bem como o tipo de dado de predição.

As métricas do modelo podem ser consultadas no painel de resultados.

Os modelos do Canvas também estão disponíveis no **SageMaker Studio**.

Advanced metrics

Positive Class

Absent

Present

Accuracy ⓘ Optimization metric

80.602%

F1 ⓘ

15.531%

Precision ⓘ

62.52%

Recall ⓘ

8.867%

AUC ⓘ

0.757

Model performance

Predicted values

Absent

Present

Actual values

Absent

Present

True positive (TP)

392

1.8% of predicted results

False positive (FP)

235

1.1% of predicted results

False negative (FN)

4029

18.3% of predicted results

True negative (TN)

17326

78.8% of predicted results

Close

Download

SAGEMAKER **CANVAS**

Após o treinamento do modelo podemos seguir com as seguintes ações:

1. Registrar o modelo

O modelo é registrado para ser utilizado no SageMaker Studio

2. O treinamento do Canvas nada mais é do que uma execução simplificada do AutoML presente no SageMaker Studio.

3. O deploy de um endpoint do modelo, a partir de experimentações AutoML, somente podem ser realizadas em instâncias de machine learning dedicadas (realtime ou batch).

EXEMPLO DE USO DOS MODELOS **IMPLANTADOS**

Após a implantação do modelo, podemos consumi-lo diretamente pelo Boto3.

```
ENDPOINT_NAME = "meu-endpoint-sagemaker"

sagemaker_runtime = boto3.client("sagemaker-runtime", region_name="us-east-1")

payload = {
    "instances": [
        {"feature1": 5.1, "feature2": 3.5, "feature3": 1.4, "feature4": 0.2}
    ]
}

payload_str = json.dumps(payload)

response = sagemaker_runtime.invoke_endpoint(
    EndpointName=ENDPOINT_NAME,
    ContentType="application/json",
    Body=payload_str,
)

result = json.loads(response["Body"].read().decode("utf-8"))
print(result)
```

CHAVE DE ACESSO: IAM

Ir no Amazon IAM, criar novo usuário (sem acesos a console) de uso programático. Incluir as seguintes políticas de permissões:

- AmazonS3FullAccess
- AmazonTextExtractFullAccess
- AmzonPollyFullAccess
- AmazonRekognitionFullAccess
- AmazonTranscribeFullAccess
- ComprehendFullAccess
- AssumeRoleComprehend

Abrir o usuário que foi criado e ir em “Credenciais de Segurança”.

Depois criar uma “Chave de Acesso”, selecionar “Aplicação executada fora da AWS”, entrar um nome “google-colab” (ou outro) e guardar em local seguro.

Evitar acesso root em caso da chave for perdida.

BOTO: **AWS SDK PYTHON**

A biblioteca Boto é o SDK oficial Python para todos os serviços da AWS, incluindo os de Machine Learning.

Quando utilizamos ele, temos a opção de usar as credenciais armazenadas no AWS CLI, que é indicada para desenvolvimentos locais. Ou podemos utilizar as chaves diretamente no código, quando utilizamos serviços externos como o Google Colab.

Sempre utilize a documentação oficial abaixo.

Caso utilize o ChatGPT ou ferramentas de geração de código enfatize que os exemplos precisam utilizar o SDK Python.



Boto 3

CLOUD APIS: SDK OU CONSOLE?

A console fornece acesso rápido aos serviços e é recomendável como primeira interação de experimentação.

O uso do SDK é recomendável quando queremos orquestrar serviços ou automatizar ações, utilizando APIs ou serviços da própria AWS como Lambda e Step Functions.

The screenshot shows the AWS Management Console interface for Amazon Comprehend. The top navigation bar includes the AWS logo, a search bar, and various service icons. The left sidebar is titled 'Amazon Comprehend' and lists options like 'Real-time analysis', 'Analysis jobs', and 'Customization'. The main content area shows a notification about a new feature, followed by a section for 'NewsCategoryClassifier_Job' creation in progress. Below this, the 'Analysis jobs' section is active, displaying a table of jobs. The table has columns for Name, Analysis type, Start, End, and Status. Two jobs are listed: 'NewsCategoryClassifier_Job' and 'BibleClassifierJob_2', both with a status of 'Completed'.

	Name	Analysis type	Start	End	Status
<input type="radio"/>	NewsCategoryClassifier_Job	Custom classification	05/08/2023, 18:02:41	05/08/2023, 18:08:48	Completed
<input checked="" type="radio"/>	BibleClassifierJob_2	Custom classification	02/08/2023, 23:53:35	02/08/2023, 23:59:41	Completed

CLOUD API: **AMAZON Textract**

Serviço responsável por converter, a partir de imagens, textos detectados ou também conhecido como OCR.

O documento pode ser lido de forma binária (local) ou pelo armazenamento do S3.



LOJAS RENNER S.A.
AV. ASSIS BRASIL, 1010 - PORTO ALEGRE - RS
CNPJ:92.754.738/0013-04 IE:096/0372202
03/05/2013 10:05:40 CCF:002260 COO:010169

CUPOM FISCAL

ITEM	CÓDIGO	DESCRIÇÃO	QTD.	UN.	VL.UNIT(R\$)	ST	IAT	VL.ITEM(R\$)
001	530236310	Blusa m3 4	1,000	Un x	89,90	T04	A	89,90%
002	520196220	PERFUME PA	1,000	Un x	199,00	F1	A	199,00%
TOTAL R\$								288,90
Dinheiro								288,90
T04=17,00%								

MD5:1b94ecab054d01b1fd8ac75ed0e3d9f1
Total Impostos Pagos R\$116,56(40,35%)Fonte:IBPT

www.lojasrenner.com.br
LJ Otavio Rocha
OP:014789-SANDRA NV:001644

YYT2bXshR@T-R/h@+2pX-pb=S*XYR&!&R-Yx9u*>T=T==>
ZPH ZPH/2EFC LOGGER ECF-IF
VERSÃO:03.04.00 ECF:014 LJ:0015 OPR:B
SSSSSSSSIPGHOMSE 03/05/2013 10:05:52
FAB:ZP030501143

Localização

Bounding boxings (x, y) em volta do texto identificado.

Tipo de Texto

Impresso ou escrito.

Confiança

Grau de precisão em porcentagem do texto identificado.

Relação

Formulários com características de “chave” e “valor” podem obter este relacionamento e tornar mais fácil a operação de detecção.

CLOUD API: **AMAZON Textract**

Serviço responsável por converter, a partir de imagens, textos detectados ou também conhecido como OCR.

O documento pode ser lido de forma binária (local) ou pelo armazenamento do S3.

```
file_name = "cupom-fiscal-1.jpg"

with open(file_name, 'rb') as file:
    img_test = file.read()
    bytes_test = bytearray(img_test)

session = boto3.Session(aws_access_key_id=ACCESS_ID, aws_secret_access_key= ACCESS_KEY)
client = session.client('textract', region_name=region)
response = client.analyze_document(Document={'Bytes': bytes_test}, FeatureTypes=['FORMS'])
```

O resultado da análise da API serão blocos com informações referentes ao posicionamento, tipo de texto encontrado, a confiança e o próprio texto detectado.

CLOUD API: **AMAZON TEXTTRACT**

Para nosso exemplo, vamos filtrar apenas os textos detectados com mais de 50% de confiança.

```
blocks = response["Blocks"]

for block in blocks:
    if block["BlockType"] == "WORD" and int(block["Confidence"])>50:
        print(block["Text"])
```

MONTEIRO
BRAGA
CONSULTORIA
EMPRESARIAL
LTDA
DEALERNET
...

Cada texto detectado está dentro de um bloco com diversas informações associadas.

```
{'BlockType': 'WORD', 'Confidence': 99.1909408569336,
'Text': 'MONTEIRO', 'TextType': 'PRINTED', 'Geometry':
{'BoundingBox': {'Width': 0.16584299504756927, 'Height':
0.017666451632976532, 'Left': 0.0032015624456107616,
'Top': 0.0019437094451859593}, 'Polygon': [{'X':
0.0032028346322476864, 'Y': 0.0019437094451859593}, {'X':
0.16904455423355103, 'Y': 0.0023132427595555782}, {'X':
0.16904117166996002, 'Y': 0.01961016096174717}, {'X':
0.0032015624456107616, 'Y': 0.019239557906985283}]], 'Id':
'd9d41600-7b48-4d0d-9e96-c19d7090d743'}
```

Como muitos textos estão em locais diferentes no documento, eles não representam a mesma linha na imagem original.

CLOUD API: **AMAZON REKOGNITION**

Este serviço é especializado em reconhecimento de imagens, com maior foco em análise facial, reconhecimento de pessoas, comparação de rostos como também detecção de objetos e descritor de cenas.

A forma de utilização segue o mesmo das outras APIs, via imagem binária ou armazenada no S3.

DESCRIPTOR DE CENA E DETECÇÃO DE OBJETOS

Descrição de imagens e detecção de objetos com informações de classes identificadas e posicionamento na imagem.

ANÁLISE FACIAL

Detecção de imagens com o posicionamento de todos os elementos faciais, incluindo direção de olhos, abertura de boca, pupila, etc.
Predição de gênero, idade e emoções.

FACE MATCH

Comparação de 2 imagens para confirmação se é a mesma pessoa.

RECONHECIMENTO FACIAL

Reconhecimento de indivíduos carregados e treinados.

Outras aplicações que segue o mesmo princípio, especializadas em:

- Reconhecimento de celebridades
- Texto em imagem
- Detecção de EPI

AMAZON REKOGNITION: DESCRITOR

Exemplo de descritor de cena e detecção de objetos, utilizando modelo treinado e pronto para uso em qualquer imagem.



AMAZON REKOGNITION: DESCRITOR

A chamada do SDK pode ser via imagem binária ou indicação do bucket S3.
Configuração como confiança mínima e máximo de objetos (labels) podem ser configurados.

```
# arquivo de imagem para análise
file_name = "aula-1-plataforma-aws/imagens/michel-friends.jpg"

# conversão da imagem em binário
with open(file_name, "rb") as file:
    img_file = file.read()
    bytes_file = bytearray(img_file)

# abrindo uma sessão
session = boto3.Session(aws_access_key_id=ACCESS_ID,
aws_secret_access_key= ACCESS_KEY)

# criando um cliente
client = session.client("rekognition", region_name=region)

# requisição para o descritor de cena
response = client.detect_labels(
    Image={'Bytes': bytes_file},
    MaxLabels=10, # Número máximo de rótulos retornados
    MinConfidence=70 # Confiança mínima necessária para considerar um
    rótulo
)
```

```
{'Labels': [{'Name': 'Urban',
'Confidence': 99.98980712890625,
'Instances': [],
'Parents': [],
'Aliases': [],
'Categories': [{'Name': 'Colors and Visual Composition'}]},
{'Name': 'Adult',
'Confidence': 98.67523956298828,
'Instances': [{'BoundingBox': {'Width':
0.14372475445270538,
'Height': 0.4697018563747406,
'Left': 0.5793469548225403,
'Top': 0.0},
'Confidence': 98.67523956298828},
{'BoundingBox': {'Width': 0.2395206093788147,
'Height': 0.8204092383384705,
'Left': 0.1193152666091919,
'Top': 0.17959074676036835},
'Confidence': 98.20369720458984},
...

```

Alguns labels retornam caixas delimitadoras na imagem.

AMAZON REKOGNITION: DESCRITOR

Plotando as caixas delimitadoras e suas respectivas classes (ou labels) indicam onde as detecções foram realizadas na imagem.



AMAZON REKOGNITION: **ANÁLISE FACIAL**

Podemos prever informações com base em estimativas do rosto, incluindo categorizações e posicionamento geram na imagem:

- **Gênero**
- **Idade**
- **Emoção**
- **Pelos na face (barba, bigode, etc.)**
- **Olhos (abertos, fechados, direção)**
- **Pupilas**
- **Nariz**
- **Abertura da boca**



AMAZON REKOGNITION: ANÁLISE FACIAL

A forma de requisitar a API para análise facial segue o mesmo padrão da chamada anterior. As informações da face como o enquadramento, sentimento, gênero, idade estimada e marcos faciais são retornadas para cada face detectada.

```
# nome do arquivo
file_name = "aula-1-plataforma-aws/imagens/faces-1.jpg"

# conversão para binário
with open(file_name, "rb") as file:
    img_file = file.read()
    bytes_file = bytearray(img_file)

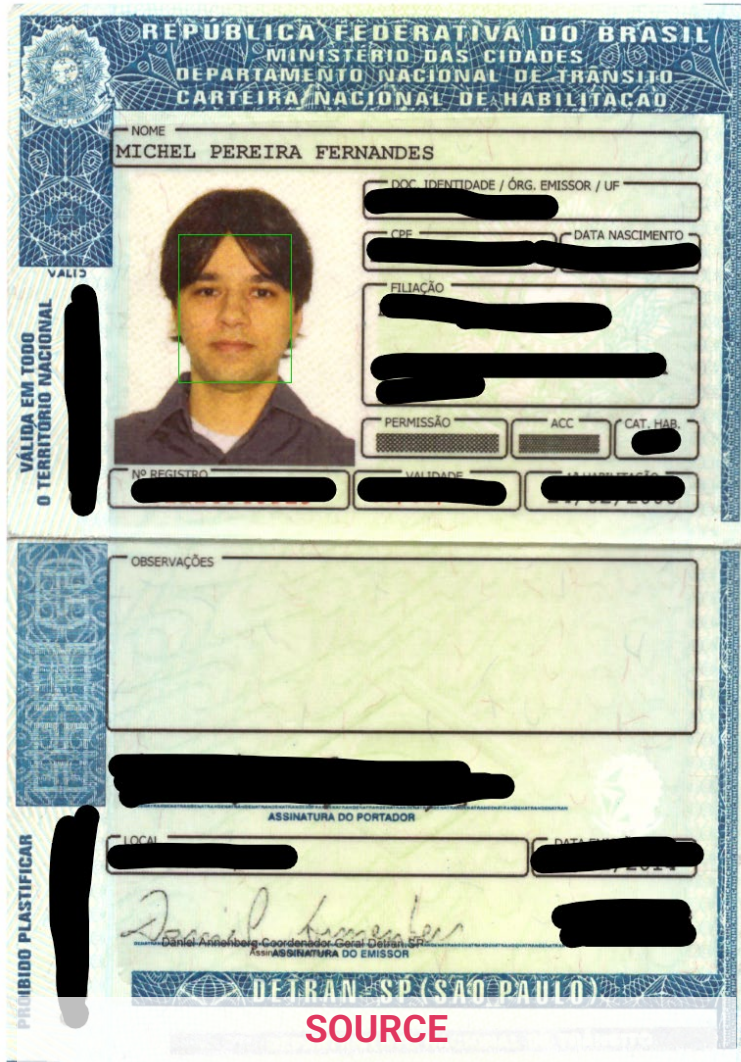
# abrindo a sessão
session = boto3.Session(aws_access_key_id=ACCESS_ID,
aws_secret_access_key= ACCESS_KEY)

# criando o cliente
client = session.client("rekognition", region_name=region)

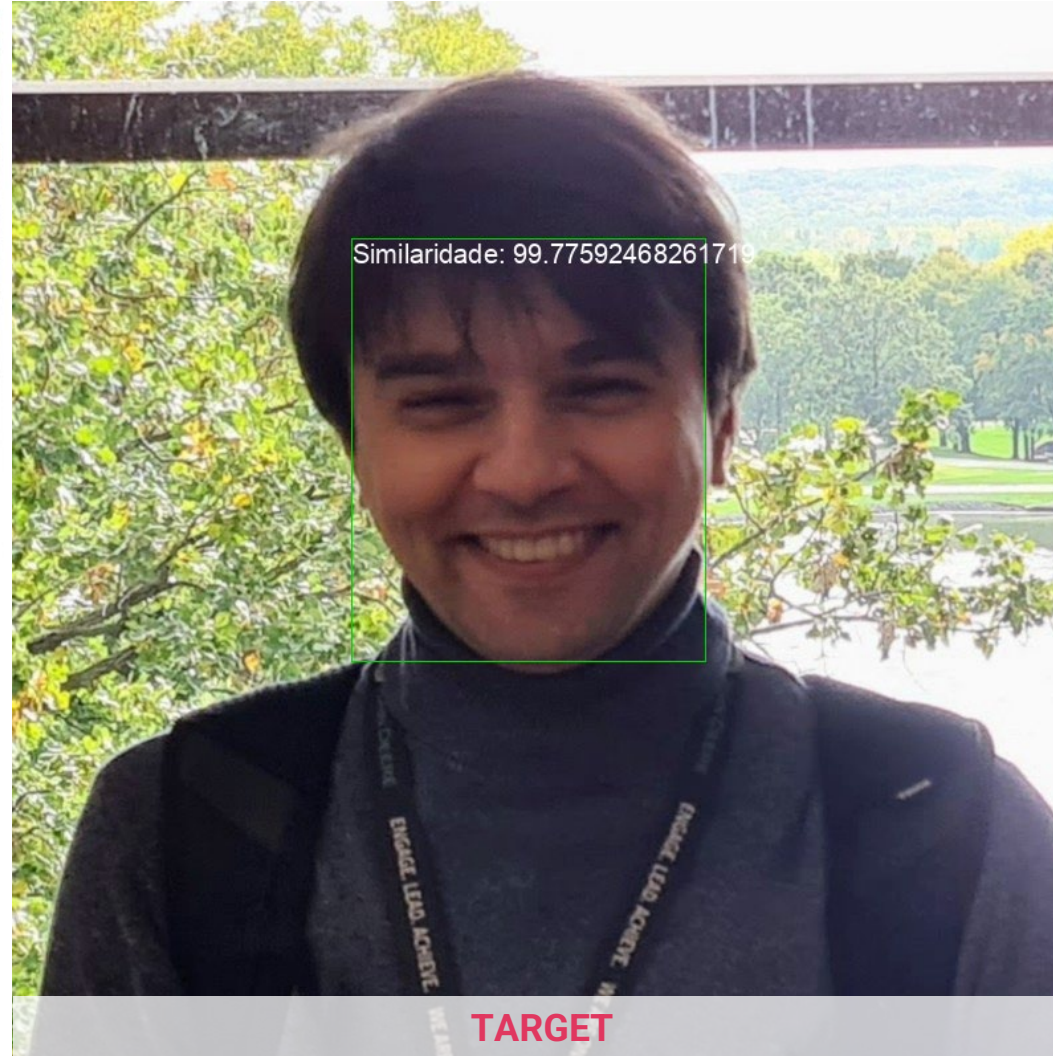
# criando a requisição
response = client.detect_faces(
    Image={'Bytes': bytes_file},
    Attributes=["ALL"]
)
```

```
{'FaceDetails': [{'BoundingBox': {'Width': 0.20637963712215424,
'Height': 0.3756640553474426, 'Left': 0.15681588649749756, 'Top':
0.5703257322311401}, 'AgeRange': {'Low': 27, 'High': 37}, 'Smile':
{'Value': True, 'Confidence': 78.45015716552734}, 'Eyeglasses':
{'Value': True, 'Confidence': 99.99114990234375}, 'Sunglasses':
{'Value': False, 'Confidence': 99.96205139160156}, 'Gender': {'Value':
'Male', 'Confidence': 99.93952178955078}, 'Beard': {'Value': True,
'Confidence': 99.21994018554688}, 'Mustache': {'Value': False,
'Confidence': 86.19662475585938}, 'EyesOpen': {'Value': True,
'Confidence': 97.26605987548828}, 'MouthOpen': {'Value': True,
'Confidence': 92.33683013916016}, 'Emotions': [{'Type': 'HAPPY',
'Confidence': 80.98323059082031}, {'Type': 'SURPRISED',
'Confidence': 14.667749404907227}, {'Type': 'FEAR', 'Confidence':
6.782601833343506}, {'Type': 'ANGRY', 'Confidence':
2.4909274578094482}, {'Type': 'SAD', 'Confidence':
2.258896589279175}, {'Type': 'DISGUSTED', 'Confidence':
1.3579745292663574}, {'Type': 'CALM', 'Confidence':
0.8241883516311646}, {'Type': 'CONFUSED', 'Confidence':
0.7052881121635437}], 'Landmarks': [{'Type': 'eyeLeft', 'X':
0.23348233103752136, 'Y': 0.6874889135360718}, {'Type':
'eyeRight', 'X': 0.318643718957901, 'Y': 0.6816147565841675},
{'Type': 'mouthLeft', 'X': 0.25294509530067444, 'Y':
0.8228636384010315}, {'Type': 'mouthRight', 'X':
0.3237247169017792, 'Y': 0.8176872730255127}, {'Type': 'nose', 'X':
...
...
...}]
}]
```


AMAZON REKOGNITION: **FACE MATCH**



SOURCE



TARGET

Comparação de 2 imagens para verificar similaridade entre elas.

Geralmente é aplicada para comparar uma foto de um documento oficial com uma selfie, para validar identidade do portador.

Apesar de não garantir uma prova de vivacidade

AMAZON REKOGNITION: FACE MATCH

A operação da comparação de faces leva em conta duas imagens. A análise vai buscar faces em cada uma das imagens e realizar a comparação entre as, com o retorno dos marcos faciais e enquadramento. A depender a similaridade, é possível tomar a decisão que as imagens são da mesma pessoa. Este valor deve ser ajustado dependendo da aplicação.

```
file_name_source = "aula-1-plataforma-aws/imagens/cnh-michel.png"
file_name_target = "aula-1-plataforma-aws/imagens/michel-1.jpg"
```

```
with open(file_name_source, "rb") as file:
    img_file = file.read()
    bytes_file_source = bytearray(img_file)
```

```
with open(file_name_target, "rb") as file:
    img_file = file.read()
    bytes_file_target = bytearray(img_file)
```

```
session = boto3.Session(aws_access_key_id=ACCESS_ID,
aws_secret_access_key= ACCESS_KEY)
```

```
client = session.client("rekognition", region_name=region)
```

```
response = client.compare_faces(
    SourceImage={'Bytes': bytes_file_source},
    TargetImage={'Bytes': bytes_file_target},
)
```

```
{'SourceImageFace': {'BoundingBox': {'Width': 0.14736957848072052,
'Height': 0.13547378778457642, 'Left': 0.2341586947441101, 'Top':
0.21867385506629944}, 'Confidence': 99.99771118164062},
'FaceMatches': [{'Similarity': 99.77592468261719, 'Face': {'BoundingBox':
{'Width': 0.33358535170555115, 'Height': 0.3994441330432892, 'Left':
0.320880651473999, 'Top': 0.2238314300775528}, 'Confidence':
99.99954986572266, 'Landmarks': [{'Type': 'eyeLeft', 'X':
0.40618500113487244, 'Y': 0.3734648525714874}, {'Type': 'eyeRight', 'X':
0.5506705045700073, 'Y': 0.36284691095352173}, {'Type': 'mouthLeft',
'X': 0.43064579367637634, 'Y': 0.51111101269721985}, {'Type':
'mouthRight', 'X': 0.5520378351211548, 'Y': 0.5024483799934387},
{'Type': 'nose', 'X': 0.48180511593818665, 'Y': 0.4522764980792999}],
'Pose': {'Roll': -4.149410247802734, 'Yaw': -2.893834352493286, 'Pitch': -
0.2870776951313019}, 'Quality': {'Brightness': 45.62997055053711,
'Sharpness': 86.86019134521484}}}], 'UnmatchedFaces': [],
'ResponseMetadata': {'RequestId': '92126656-4c5f-49b1-aaf3-
7ad4053af8e7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-
requestid': '92126656-4c5f-49b1-aaf3-7ad4053af8e7', 'content-type':
'application/x-amz-json-1.1', 'content-length': '915', 'date': 'Sun, 06 Aug
2023 17:43:45 GMT'}, 'RetryAttempts': 0}}
```

DESAFIO 1

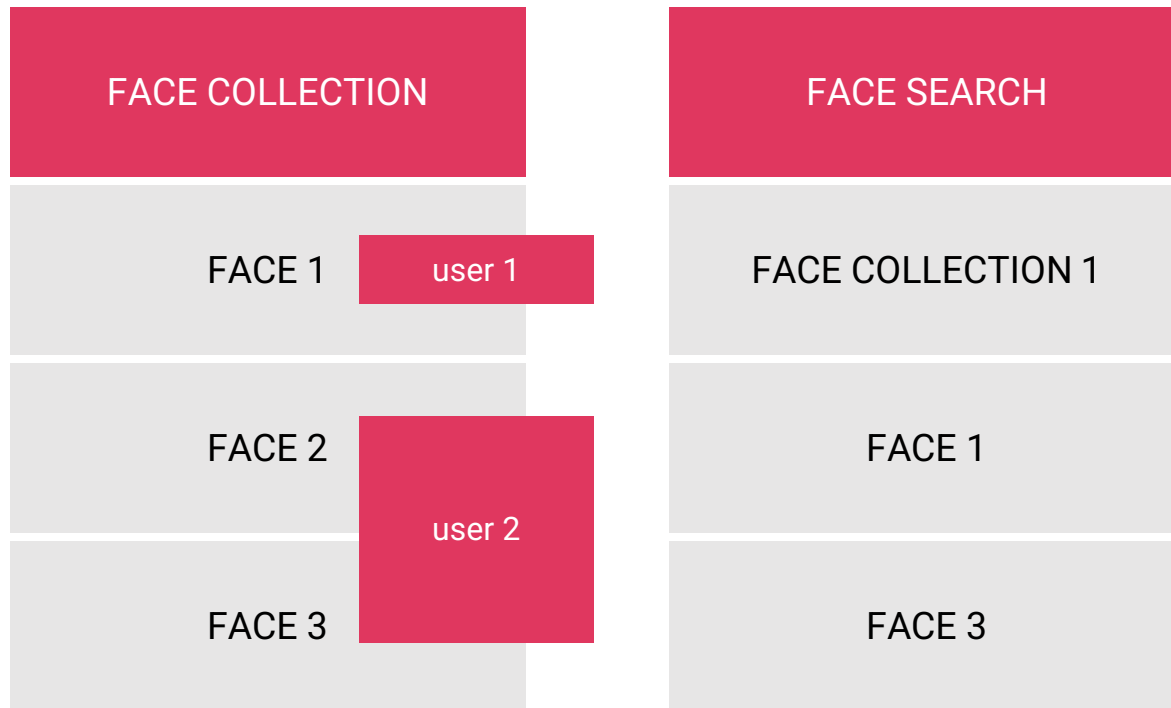
Construa um algoritmo para comparar 2 faces afim de determinar se elas são iguais. Para ambas as faces garanta que nenhuma das condições abaixo seja permitido em nenhuma das faces de comparação:

1. Face com oclusão (face parcialmente oculta)
2. Óculos de sol
3. Olhos fechados

Além disso, o algoritmo deve considerar o valor de similaridade como *score* e somente valores acima de 90% deverão ser aceitos.

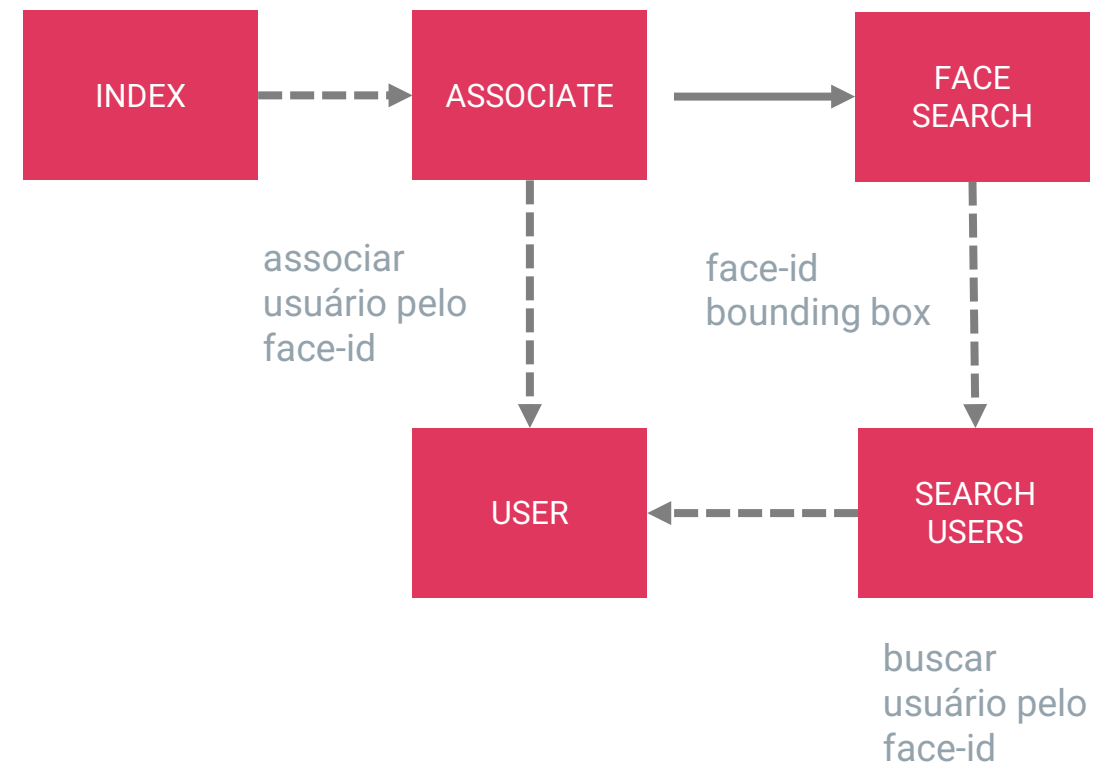
AMAZON REKOGNITION: **FACE RECOGNITION**

Para reconhecer faces customizadas, isto é, aquelas diferente das pré-carregadas (detecção de celebridades), precisamos utilizar um conjunto de operações para treinar e posteriormente identificar em uma nova imagem. Limite de 20 milhões para cada coleção e 4096 para correspondências.



As faces não precisam ser recortadas, basta que tenha uma única face na imagem enviada.

Se for encontrado alguma correspondência será retornado qual id de face e sua similaridade.



A busca por um usuário é feita pelo face-id com maior similaridade.

FACE RECOGNITION: **DETALHES IMPORTANTES**

Use o Amazon Rekognition como o primeiro passo para encontrar possíveis correspondências. As respostas das operações de reconhecimento facial permitem que você obtenha rapidamente um conjunto de correspondências potenciais para uma consideração posterior.

Não utilize as respostas do Amazon Rekognition para tomar decisões autônomas em cenários que requerem análise por um humano. Se você é uma agência de aplicação da lei usando o Amazon Rekognition para auxiliar na identificação de uma pessoa em conexão com uma investigação criminal, e ações serão tomadas com base na identificação que possam impactar os direitos civis ou equivalentes direitos humanos dessa pessoa, a decisão de agir deve ser tomada por uma pessoa devidamente treinada com base em sua análise independente da evidência de identificação.

Utilize um **limiar de similaridade de 99%** para cenários em que correspondências de similaridade facial altamente precisas são necessárias. Um exemplo disso é autenticar o acesso a um prédio.

Quando os direitos civis são uma preocupação, como em casos de uso envolvendo aplicação da lei, utilize limiares de confiança de 99% ou mais e empregue a revisão humana das previsões de comparação facial para garantir que os direitos civis de uma pessoa não sejam violados.

Utilize um limiar de similaridade inferior a 99% para cenários que se beneficiam de um conjunto maior de correspondências potenciais. Um exemplo disso é encontrar pessoas desaparecidas. Se necessário, você pode usar o atributo de resposta "Similarity" para determinar o quão similares as correspondências potenciais são em relação à pessoa que você deseja reconhecer.

Tenha um plano para correspondências faciais falsas positivas que são retornadas pelo Amazon Rekognition. Por exemplo, melhore a correspondência usando várias imagens da mesma pessoa ao construir o índice com a operação IndexFaces.

AMAZON REKOGNITION: **FACE RECOGNITION**



Cada imagem de referência é automaticamente selecionada uma região de interesse do rosto, de forma semelhante a operação de detectar face.

Posteriormente é realizada uma operação de comparação das faces de uma determinada coleção, que pode haver mais de um indivíduo com uma imagem de alvo.

Por padrão, haverá uma correspondência com imagens maiores ou iguais à similaridade de 80%, da qual pode ser configurado para outros valores.

AMAZON REKOGNITION: **FACE RECOGNITION**

O reconhecimento de faces requer pelo menos uma coleção e, opcionalmente, um usuário para realizar o processo.

A detecção em outras imagens sempre terá como referência a identificação da face. A partir dela conseguimos depois associar com o usuário respectivo.

```
# criando sessão
session = boto3.Session(aws_access_key_id=ACCESS_ID,
aws_secret_access_key= ACCESS_KEY)

# abrindo cliente
client = session.client("rekognition", region_name=region)

# criando uma coleção
response = client.create_collection(
    CollectionId = "fiap-faces"
)

# criando nova pessoa para usar como identidade
response = client.create_user(
    CollectionId='fiap-faces',
    UserId='michel-fernandes'
)
```

Preparando a associação das faces:

1. Criação de uma coleção para receber as imagens das faces
2. Criação de um usuário para associar com as faces que serão enviadas.

AMAZON REKOGNITION: **FACE RECOGNITION**

Cada face criada deverá ser associada uma coleção pela operação de indexação.

```
# adicionando face 5
```

```
# adicionando face 4
```

```
# adicionando face 3
```

```
# adicionando face 2
```

```
# adicionando face 1
```

```
# identificação do bucket
```

```
bucket_name = "fiap-cognitive-platforms"
```

```
# identificação do arquivo
```

```
file_name = "faces/michel/michel-face-1.jpg"
```

```
response = client.index_faces(
```

```
    CollectionId='fiap-faces',
```

```
    Image={
```

```
        'S3Object': {
```

```
            'Bucket': bucket_name,
```

```
            'Name': file_name
```

```
        }
```

```
    }
```

```
)
```

```
...
```



AMAZON REKOGNITION: **FACE RECOGNITION**

Após a indexação, a qual retorna uma identificação para cada face, criamos um usuário associado essas identificações. Assim, detecções desta faces poderão ser identificadas pelo usuário respectivo.

```
response = client.associate_faces(  
    CollectionId="fiap-faces",  
    UserId="michel-fernandes",  
    FaceIds=["e83e5d07-620c-42fe-b887-c87edf276e5e", "2cc76cfb-7e16-4812-8099-47ebd9482985", "78041e1c-ceb3-4ac1-8e39-10c3de1c7b3b", "78041e1c-ceb3-4ac1-8e39-10c3de1c7b3b", "7de5b11f-8756-4ed8-ae29-9b5c6c434ad5"]  
)
```

```
file_name_source = "aula-1-plataforma-aws/imagens/michel-2.jpg"
```

```
with open(file_name_source, "rb") as file:  
    img_file = file.read()  
    bytes_file_source = bytearray(img_file)  
  
# enviando a requisição com a imagem local  
response = client.search_faces_by_image(  
    Image={'Bytes': bytes_file_source},  
    CollectionId='fiap-faces'  
)
```

AMAZON REKOGNITION: **FACE RECOGNITION**

O resultado da análise de reconhecimento de faces retorna quais identificações de face possuem maior similaridade, a partir da face detectada.

No caso abaixo, o *FacelId* com maior similaridade, é a primeira a ser exibida (99,997%), e83e5d07-620c-42fe-b887-c87edf276e5e'.

```
{'SearchedFaceBoundingBox': {'Width': 0.08660564571619034, 'Height': 0.1640600711107254, 'Left': 0.4237428903579712, 'Top': 0.21024146676063538}, 'SearchedFaceConfidence': 99.9974594116211, 'FaceMatches': [{ 'Similarity': 99.99769592285156, 'Face': { 'FacelId': 'e83e5d07-620c-42fe-b887-c87edf276e5e', 'BoundingBox': { 'Width': 0.06068820133805275, 'Height': 0.1598840057849884, 'Left': 0.3239020109176636, 'Top': 0.29020199179649353}, 'ImageId': 'be2b4636-bf0e-3ca7-b8eb-3d4763abc069', 'Confidence': 99.98999786376953, 'IndexFacesModelVersion': '6.0'}}, { 'Similarity': 99.99230194091797, 'Face': { 'FacelId': '78041e1c-ceb3-4ac1-8e39-10c3de1c7b3b', 'BoundingBox': { 'Width': 0.17435400187969208, 'Height': 0.12009300291538239, 'Left': 0.3828980028629303, 'Top': 0.17252500355243683}, 'ImageId': '291d95be-d0a0-3b2b-a1ee-73e58b8f84d8', 'Confidence': 99.99039459228516, 'IndexFacesModelVersion': '6.0'}}, { 'Similarity': 99.96699523925781, 'Face': { 'FacelId': '2cc76cfb-7e16-4812-8099-47ebd9482985', 'BoundingBox': { 'Width': 0.5389999747276306, 'Height': 0.41245800256729126, 'Left': 0.21515800058841705, 'Top': 0.29811200499534607}, 'ImageId': '99683268-9be7-37e5-9f74-f7150b67d592', 'Confidence': 99.99970245361328, 'IndexFacesModelVersion': '6.0'}}, { 'Similarity': 99.96072387695312, 'Face': { 'FacelId': '7de5b11f-8756-4ed8-ae29-9b5c6c434ad5', 'BoundingBox': { 'Width': 0.6120060086250305, 'Height': 0.4575960040092468, 'Left': 0.27334800362586975, 'Top': 0.22377200424671173}, 'ImageId': 'c4564d26-c117-3e5c-85c9-f795db4375af', 'Confidence': 99.99970245361328, 'IndexFacesModelVersion': '6.0'}}], 'FaceModelVersion': '6.0', 'ResponseMetadata': { 'RequestId': '29ef4d09-87b2-49de-9751-1000457d3e10', 'HTTPStatusCode': 200, 'HTTPHeaders': { 'x-amzn-requestid': '29ef4d09-87b2-49de-9751-1000457d3e10', 'content-type': 'application/x-amz-json-1.1', 'content-length': '1522', 'date': 'Sun, 06 Aug 2023 23:42:33 GMT'}, 'RetryAttempts': 0}}
```

AMAZON REKOGNITION: **FACE RECOGNITION**

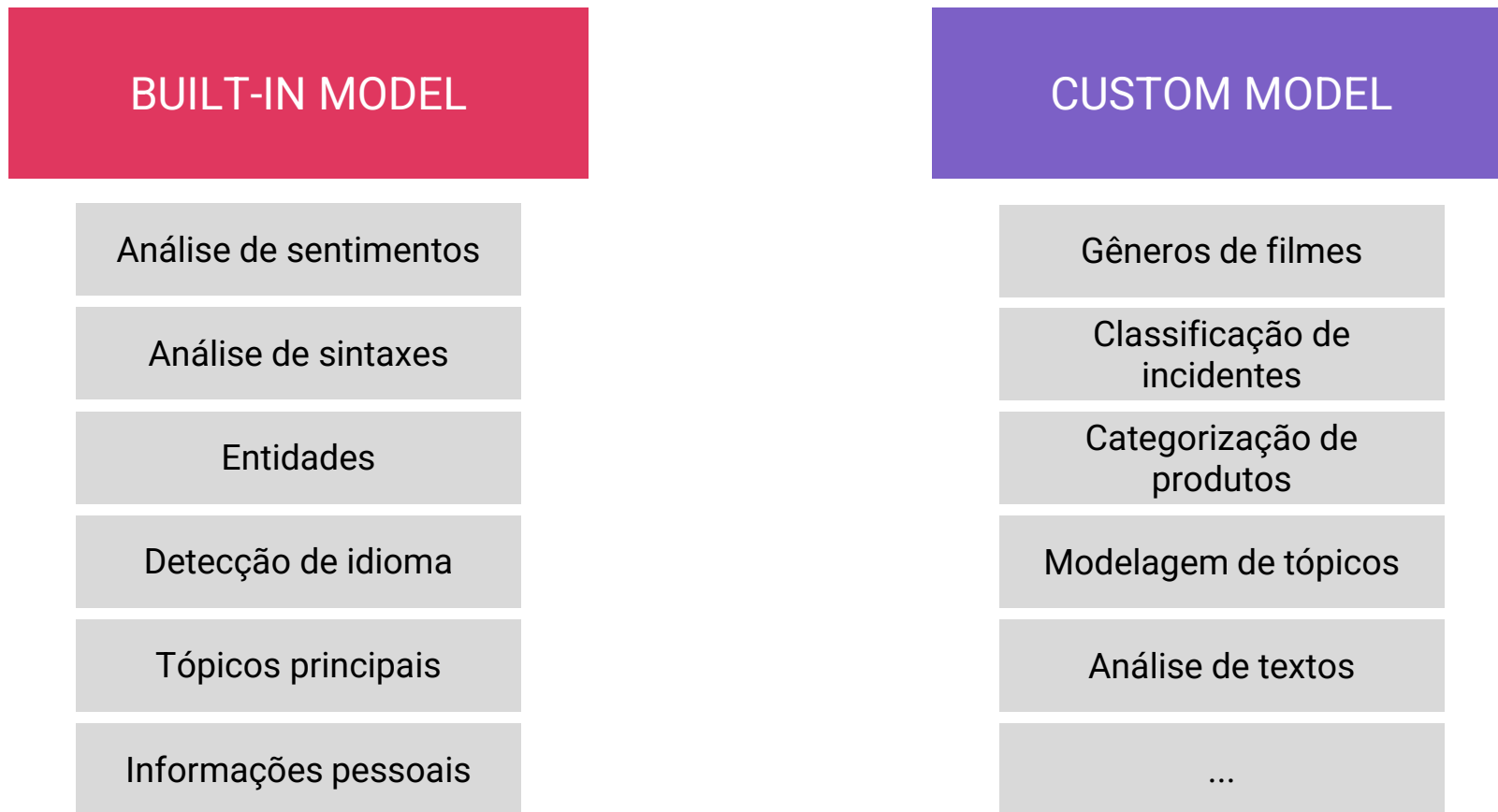
A etapa final é conferir qual usuário está associado ao *FaceId* detectado com maior similaridade.

```
response = client.search_users(  
    FaceId=face_id_max_similarity,  
    CollectionId='fiap-faces'  
)
```

```
{'UserMatches': [{'Similarity': 99.99992370605469, 'User': {'UserId': 'michel-fernandes', 'UserStatus': 'ACTIVE'}}], 'FaceModelVersion': '6',  
'SearchedFace': {'FaceId': 'e83e5d07-620c-42fe-b887-c87edf276e5e'}, 'ResponseMetadata': {'RequestId': '5e5e48ca-2c43-4f0a-95b6-  
0c092bc86d19', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '5e5e48ca-2c43-4f0a-95b6-0c092bc86d19', 'content-type':  
'application/x-amz-json-1.1', 'content-length': '197', 'date': 'Sun, 06 Aug 2023 23:43:48 GMT'}, 'RetryAttempts': 0}}
```

AMAZON COMPREHEND

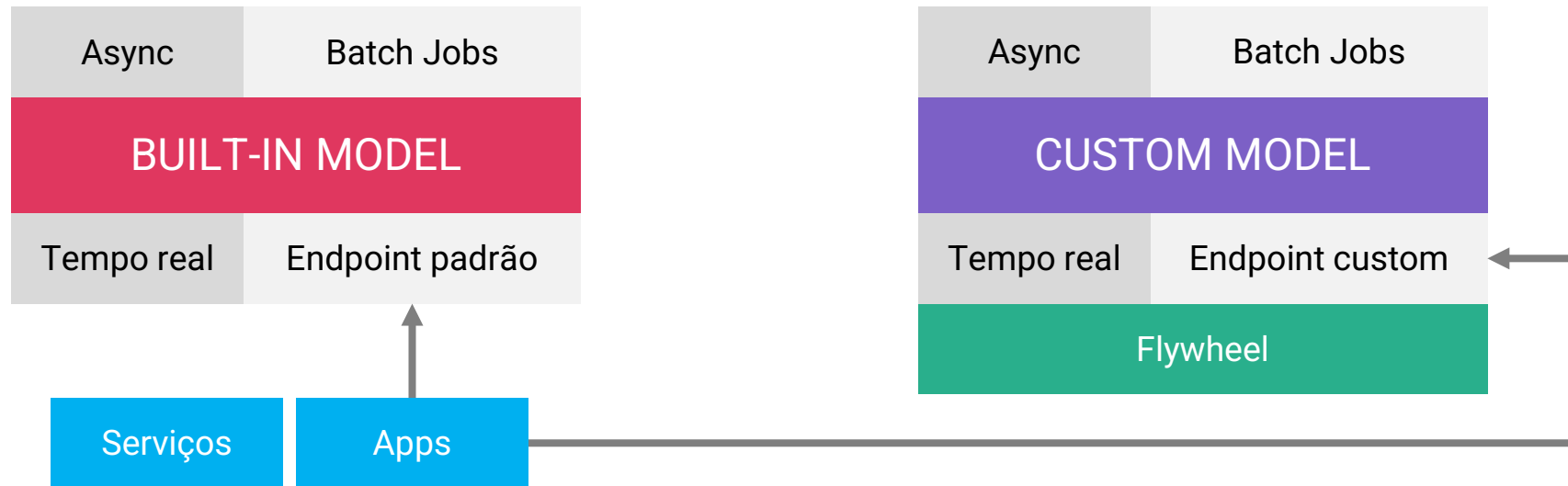
O serviço de análise de linguagem natural, utiliza dois tipos de modelos. O primeiro é o padrão (built-in), capaz de realizar análise de sentimentos e de sintaxes, como também detectar entidades. O segundo é utilizado modelos customizados, com bases de treinamentos próprias para tarefas bem específicas.



TIPOS DE EXECUÇÃO

As execuções podem ser feitas em ambos os tipos de modelos de duas formas:

- Tempo real (mais caro, mas instantâneo)
- Execução em batch (mais barato, mas não é instantâneo, dependendo pode durar mais que um minuto)



Os modelos custom ainda contam com uma funcionalidade de re-treinamento automático conhecida como “Flywheel”.

Isso garante que o modelo esteja sempre atualizado com dados novos (desde que curados e prontos para treino).

DETECÇÃO DE ENTIDADES

A extração de entidades é realizada a partir de um texto e, opcionalmente, do idioma associado, pois assim o modelo não precisa detectar automaticamente e com isso aumenta a precisão das detecções.

"Se depender do bilionário Elon Musk, a luta contra Mark Zuckerberg (CEO da Meta, que controla o Instagram, WhatsApp e Facebook) está em pé e ainda haverá transmissão ao vivo do evento. Em publicação neste domingo (6), Musk disse que o embate será transmitido na rede social X (anteriormente conhecida como Twitter) e que toda a receita envolvida será doada para veteranos dos EUA — militares que serviram nas Forças Armadas do país. Zuckerberg monta octógono no quintal de casa e 'compra briga' com a esposa Tchau, passarinho: Twitter deixa marca icônica para usar letra X Rinha de bilionários: Elon Musk e Zuckerberg se desafiam para luta em jaula. Ainda não há data para o evento nem confirmação por parte de Mark Zuckerberg. No fim de julho, Zuckerberg disse à agência de notícias Reuters que não tinha certeza se o embate contra Musk iria ocorrer de fato. Rinha de bilionários Depois de Musk comprar o Twitter, a Meta anunciou planos de fazer uma rede social concorrente — o Threads foi lançado no início de junho. Isso fez com que o bilionário dono da SpaceX e da Tesla passasse a fazer provocações a Zuckerberg. No fim de junho, Musk tuitou que estava pronto para uma luta na jaula, se ele [Zuckerberg] estiver pronto para isso. Na época, o cofundador do Facebook respondeu em uma postagem no Instagram que era só dizer o local. Zuckerberg é praticante de jiu-jitsu e treina MMA (artes marciais mistas) — recentemente, ele publicou uma foto praticando a modalidade com o campeão dos médios do UFC, Israel Adesanya. Ele leva a tarefa tão a sério que colocou um octógono no quintal de casa. Musk também tem conhecidos no ramo do MMA. Ele foi visto treinando com o ex-campeão dos meio-médios do UFC Georges St. Pierre. Além disso, o empresário tem feito postagens dizendo que tem se exercitado com frequência."

```
# abrindo a sessão
session = boto3.Session(aws_access_key_id=ACCESS_ID, aws_secret_access_key=
ACCESS_KEY)
# criando o cliente
client = session.client("comprehend", region_name=region)

# executando a requisição
response = client.detect_entities(
    Text=texto,
    LanguageCode='pt',
)
```

```
{'Entities': [{'Score': 0.9984291791915894,
  'Type': 'PERSON',
  'Text': 'Elon Musk',
  'BeginOffset': 26,
  'EndOffset': 35},
{'Score': 0.9963639974594116,
  'Type': 'PERSON',
  'Text': 'Mark Zuckerberg',
  'BeginOffset': 51,
  'EndOffset': 66},
{'Score': 0.9359315037727356,
  'Type': 'PERSON',
  'Text': 'CEO',
  'BeginOffset': 68,
  'EndOffset': 71},
...]
```

Quanto maior o Score maior a confiança na detecção.

DETECÇÃO DE TÓPICOS CHAVE

Os tópicos chave são complementares as entidades, auxiliam no *tagging* de textos e artigos. Sua requisição segue a mesma do detector de entidades.

```
session = boto3.Session(aws_access_key_id=ACCESS_ID,
aws_secret_access_key= ACCESS_KEY)
client = session.client("comprehend", region_name=region)

response = client.detect_key_phrases(
    Text=texto,
    LanguageCode='pt',
)
```

Quanto maior o Score maior a confiança na detecção.

```
{'KeyPhrases': [{ 'Score': 0.9986598491668701,
  'Text': 'bilionário',
  'BeginOffset': 15,
  'EndOffset': 25},
{ 'Score': 0.9945086240768433,
  'Text': 'Elon Musk',
  'BeginOffset': 26,
  'EndOffset': 35},
{ 'Score': 0.9980497360229492,
  'Text': 'a luta contra Mark Zuckerberg',
  'BeginOffset': 37,
  'EndOffset': 66},
{ 'Score': 0.9993777275085449,
  'Text': 'CEO da Meta',
  ...
```

ANÁLISE DE SENTIMENTOS

Análise em sentimentos de textos procura distinguir eventuais vieses de comentários e artigos. Pode se tornar uma ferramenta importante em revisão de escrita.

A determinação de sentimento é dividida em: **positivo**, **negativo**, **misto** e **neutro**.

```
session = boto3.Session(aws_access_key_id=ACCESS_ID,  
aws_secret_access_key= ACCESS_KEY)  
client = session.client("comprehend", region_name=region)  
  
response = client.detect_sentiment(  
    Text=texto,  
    LanguageCode='pt',  
)
```

Quanto maior o Score maior a confiança na detecção do sentimento no texto.

```
{'Sentiment': 'NEUTRAL',  
'SentimentScore': {'Positive': 0.0018087361240759492,  
'Negative': 0.002231369260698557,  
'Neutral': 0.9959061145782471,  
'Mixed': 5.382212475524284e-05},  
'ResponseMetadata': {'RequestId': '70812d27-2c32-  
4828-9243-b5d236700b92',  
'HTTPStatusCode': 200,  
'HTTPHeaders': {'x-amzn-requestid': '70812d27-2c32-  
4828-9243-b5d236700b92',  
'content-type': 'application/x-amz-json-1.1',  
'content-length': '165',  
'date': 'Mon, 07 Aug 2023 00:32:41 GMT'},  
'RetryAttempts': 0}}
```

INFORMAÇÕES PESSOAIS IDENTIFICÁVEIS (PII)

Informações privadas, tais como nome, endereço e documentos, podem envolver potenciais riscos com leis de proteção de dados.

O resultado é semelhante a detecção de entidades, mas com um foco exclusivo em informações pessoais.

```
session = boto3.Session(aws_access_key_id=ACCESS_ID,  
aws_secret_access_key= ACCESS_KEY)  
client = session.client("comprehend", region_name=region)
```

```
response = client.detect_pii_entities(  
    Text=texto,  
    LanguageCode='en',  
)
```

```
{'Entities': [{'Score': 0.9940863251686096,  
    'Type': 'NAME',  
    'BeginOffset': 26,  
    'EndOffset': 35},  
    {'Score': 0.9892066717147827,  
    'Type': 'NAME',  
    'BeginOffset': 51,  
    'EndOffset': 66},  
    {'Score': 0.9998918771743774,  
    'Type': 'NAME',  
    'BeginOffset': 218,  
    'EndOffset': 222},  
    {'Score': 0.9998757839202881,  
    'Type': 'NAME',  
    'BeginOffset': 433,  
    'EndOffset': 443},
```

Quanto maior o Score maior a confiança na detecção do sentimento no texto.

Além do tipo de PII informa o posicionamento no texto.

MODELO CUSTOMIZADO

Podemos criar um modelo customizado treinamento textos correspondentes a uma determinada classe, por exemplo artigos e sua determinada categoria.

Os dados precisam estar no formato CSV, com a **primeira coluna da classe e a segunda com o texto**. A melhor forma de separar os dados é utilizando o Pandas e não o Excel.

O Comprehend separa os dados de treinamento e teste automaticamente.

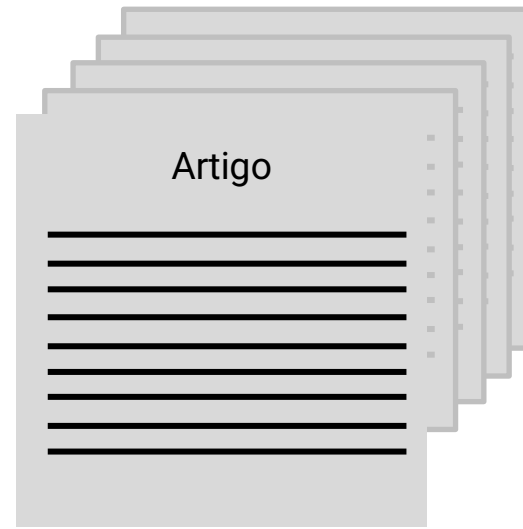
Modelos customizados suportam, além de texto, arquivos em PDF, Word e Imagens.



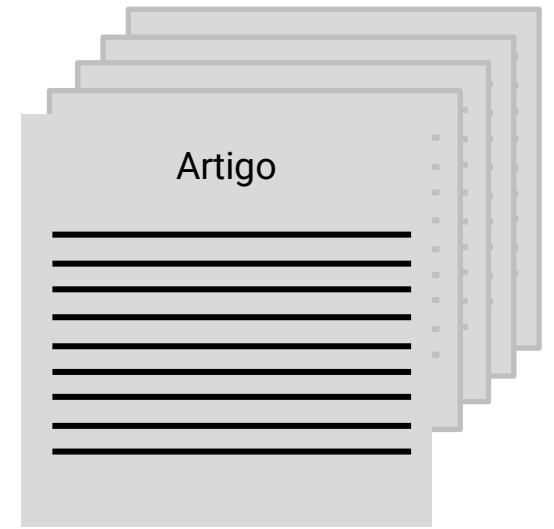
Esporte



Ciências



Tecnologia



Política

AMAZON COMPREHEND: MODELO CUSTOMIZADO

O dataset que iremos trabalhar são de notícias categorizadas da BBC News.

Os dados podem ser baixados da universidade College Dublin (<http://mlg.ucd.ie/datasets/bbc.html>) ou diretamente deste repositório da qual os dados já foram tratadas (<https://github.com/michelpf/dataset-bbc-news-category>). Neste último repositório tem os passos para transformar os dados para o formato necessário do Comprehend.

O dataset possui as seguintes categorias:

- Business
- Entertainment
- Politics
- Sport
- Tech

Em cada pasta de cada categoria possuem os textos referentes a ela, no formato TXT.



AMAZON COMPREHEND: MODELO CUSTOMIZADO

Primeiro, precisamos criar um modelo, e depois treiná-lo para torna-lo utilizável. Informações de identificação.

Model settings

Model name [Info](#)

The name can have up to 63 characters, and it must be unique. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Version name - *optional*

The name can have up to 63 characters, and it must be unique. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Language

Portuguese

▼

[Supported languages](#) [↗](#)

☐ Classifier encryption [Info](#)

AMAZON COMPREHEND: MODELO CUSTOMIZADO

Informações de dados, como o uso de texto plano ou PDF, Word ou CSV combinados.
 Tipo de formato de dados, neste caso utilizaremos o CSV sem necessidade de aumentar os dados existentes.

Data specifications
[Info](#)

Annotation and data format

Configure how you are providing your data.

Training model type

☒ **Plain text documents**
 Choose this option if you labeled plain-text documents for CSV or augmented manifest.

☐ **Native documents (PDF, Word, images)**
 Choose this option if you labeled PDF, Word documents or images for CSV.

Data format

To train your custom model, you must provide training data. This data must be formatted as either a CSV file or as one or more augmented manifest files.

☒ **CSV file** [Info](#)
 The CSV file that contains either the annotations or the entity lists for your training data. The required format depends on the type of CSV file that you provide.

☐ **Augmented manifest** [Info](#)
 A labeled training dataset that is produced by Amazon SageMaker Ground Truth. You can provide up to 5 augmented manifest files. To create an augmented manifest file, you can create a labeling job in Amazon SageMaker Ground Truth.

Classifier mode

☒ **Using Single-label mode**
 The training data file must have one class and one document on each line. It must have at least 10 documents for each class.

 Example

COMEDY	document text 1
COMEDY	document text 2
DRAMA	document text 3

☐ **Using Multi-label mode**
 The training data file must have one or more classes and one document on each line. It must have at least 10 documents for each class.

 Example

COMEDY	document text 1
DRAMA	document text 2
COMEDY DRAMA	document text 3

Training dataset
 A training dataset teaches your model to classify. Paste the URL of an input data file in S3, or select a bucket or folder location in S3.

Training data location on S3

Test dataset - new
 A test dataset evaluates the performance of your trained model.

Test data source
 Test data is used to provide data performance for your model.

☒ **Autosplit**
 Comprehend will select 10% of your provided training data to use as testing data.
 ☐ **Customer provided**
 Get advanced control over training, testing, and performance tuning by specifying your test data.

O classificador poderá ser multi ou single label, ou seja, podemos adicionar mais de um label para cada documento.

Por fim local dos arquivos de treinamento e como o treinamento irá dividir os dados de teste.

AMAZON COMPREHEND: MODELO CUSTOMIZADO

A última etapa é configurar uma *role* que tem permissão para escrever e ler nos *buckets* informados.

Output data - optional [Info](#)

S3 location

Paste the URL of a bucket or folder location in S3, or select a bucket or folder location in S3.

[View](#)[Browse S3](#)☐ Encryption [Info](#)

IAM role [Info](#)

IAM role

- ☒ Use an existing IAM role
- ☐ Create an IAM role

Role name

A role that grants access to the S3 input and output locations.

Output data é opcional, traz dados de classificação como a matriz de confusão. Estes dados também são apresentados na console depois do treinamento.

AMAZON COMPREHEND: MODELO CUSTOMIZADO

Depois do treinamento ter sido concluído, verificamos se as métricas estão em patamares mínimos, ou seja, dados de precisão acima de 80% (isso varia com a especificação de cada projeto).

Accuracy: percentual de classificações feitas corretamente total.
Os valores de Precision, Recall e F1 Score são calculados como média de cada classe, por isso também são chamados de Macro. Os valores Micro são aqueles scores sem a utilização das médias, e sim com o cálculo baseado em todo o dataset de teste.
Hamming loss é a fração de labels que foram incorretamente classificados, do total existentes. Quanto menor, mlehor.

Input & Output	Performance	Endpoints	Tags, VPC, and Policy	Application integration
Version performance				
Test data source	Precision	Recall	F1 score	
Autosplit	0.96	0.96	0.96	
Accuracy	Micro precision	Micro recall	Micro F1 score	
0.96	0.96	0.96	0.96	
Hamming loss				
0.04				

AMAZON COMPREHEND: MODELO CUSTOMIZADO

A operação de classificação pode ser feita na forma em tempo real (mais caro) ou em batch job (menos caro). Sempre que possível, opte por classificações em batch.

```
# abrindo sessão
session = boto3.Session(aws_access_key_id=ACCESS_ID, aws_secret_access_key= ACCESS_KEY)
# criando cliente
client = session.client("comprehend", region_name=region)

# realizando a requisição de classificação
# neste caso, executando no Colab, precisamos permitir a role abaixo para o usuário associado com a sessão
response = client.start_document_classification_job(
    JobName="NewsCategoryClassifier_Job",
    DocumentClassifierArn="arn:aws:comprehend:us-east-1:989944764342:document-classifier/NewsCategoryClassifier/version/1",
    InputDataConfig={
        "S3Uri": "s3://fiap-cognitive-platforms/docs/news-category/bbc_news_category_test.csv",
        "InputFormat": "ONE_DOC_PER_FILE",
    },OutputDataConfig={
        "S3Uri": "s3://fiap-cognitive-platforms/docs/news-category/",
    },
    DataAccessRoleArn="arn:aws:iam::989944764342:role/ComprehendJob",
)
```

O Inputformat ONE_DOC_PER_FILE indica que os documentos estarão no mesmo arquivo CSV.

AMAZON COMPREHEND: MODELO CUSTOMIZADO

Os resultados ficam armazenados em um arquivo “output.tar.gz”. Necessário descompactar para analisar seus resultados, que estão no formato JSON.

É associado para cada texto as classes detectadas e sua pontuação, quanto maior, maior a precisão neste tipo de categoria.

```
{
  "File": "bbc_news_category_test.csv",
  "Line": "0",
  "Classes": [
    { "Name": "politics", "Score": 1.0 },
    { "Name": "business", "Score": 0.0 },
    { "Name": "sport", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "1",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "entertainment", "Score": 0.0 },
    { "Name": "business", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "2",
  "Classes": [
    { "Name": "sport", "Score": 0.975 },
    { "Name": "politics", "Score": 0.0107 },
    { "Name": "entertainment", "Score": 0.0093 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "3",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "sport", "Score": 0.0 },
    { "Name": "politics", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "4",
  "Classes": [
    { "Name": "tech", "Score": 0.997 },
    { "Name": "business", "Score": 0.0014 },
    { "Name": "politics", "Score": 0.001 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "5",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "politics", "Score": 0.0 },
    { "Name": "sport", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "6",
  "Classes": [
    { "Name": "entertainment", "Score": 0.9797 },
    { "Name": "sport", "Score": 0.0104 },
    { "Name": "tech", "Score": 0.0044 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "7",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "sport", "Score": 0.0 },
    { "Name": "entertainment", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "8",
  "Classes": [
    { "Name": "sport", "Score": 0.9967 },
    { "Name": "business", "Score": 0.0012 },
    { "Name": "politics", "Score": 0.0011 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "9",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "business", "Score": 0.0 },
    { "Name": "politics", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "10",
  "Classes": [
    { "Name": "politics", "Score": 0.9374 },
    { "Name": "entertainment", "Score": 0.038 },
    { "Name": "tech", "Score": 0.0122 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "11",
  "Classes": [
    { "Name": "sport", "Score": 0.965 },
    { "Name": "politics", "Score": 0.0125 },
    { "Name": "business", "Score": 0.0086 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "12",
  "Classes": [
    { "Name": "sport", "Score": 0.9994 },
    { "Name": "entertainment", "Score": 0.0002 },
    { "Name": "politics", "Score": 0.0002 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "13",
  "Classes": [
    { "Name": "sport", "Score": 0.9991 },
    { "Name": "politics", "Score": 0.0004 },
    { "Name": "entertainment", "Score": 0.0003 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "14",
  "Classes": [
    { "Name": "sport", "Score": 0.9658 },
    { "Name": "entertainment", "Score": 0.0226 },
    { "Name": "politics", "Score": 0.0076 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "15",
  "Classes": [
    { "Name": "sport", "Score": 0.9941 },
    { "Name": "entertainment", "Score": 0.0025 },
    { "Name": "politics", "Score": 0.002 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "16",
  "Classes": [
    { "Name": "sport", "Score": 0.9829 },
    { "Name": "entertainment", "Score": 0.0084 },
    { "Name": "business", "Score": 0.0046 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "17",
  "Classes": [
    { "Name": "sport", "Score": 0.9993 },
    { "Name": "entertainment", "Score": 0.0004 },
    { "Name": "politics", "Score": 0.0002 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "18",
  "Classes": [
    { "Name": "sport", "Score": 0.9944 },
    { "Name": "entertainment", "Score": 0.0018 },
    { "Name": "business", "Score": 0.0015 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "19",
  "Classes": [
    { "Name": "business", "Score": 0.9996 },
    { "Name": "politics", "Score": 0.0002 },
    { "Name": "tech", "Score": 0.0001 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "20",
  "Classes": [
    { "Name": "sport", "Score": 0.996 },
    { "Name": "entertainment", "Score": 0.0028 },
    { "Name": "politics", "Score": 0.0006 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "21",
  "Classes": [
    { "Name": "politics", "Score": 1.0 },
    { "Name": "business", "Score": 0.0 },
    { "Name": "sport", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "22",
  "Classes": [
    { "Name": "tech", "Score": 0.9999 },
    { "Name": "business", "Score": 0.0001 },
    { "Name": "entertainment", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "23",
  "Classes": [
    { "Name": "entertainment", "Score": 0.9999 },
    { "Name": "tech", "Score": 0.0 },
    { "Name": "business", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "24",
  "Classes": [
    { "Name": "tech", "Score": 0.9999 },
    { "Name": "politics", "Score": 0.0 },
    { "Name": "sport", "Score": 0.0 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "25",
  "Classes": [
    { "Name": "entertainment", "Score": 0.7838 },
    { "Name": "business", "Score": 0.1513 },
    { "Name": "tech", "Score": 0.041 }
  ]
},
{
  "File": "bbc_news_category_test.csv",
  "Line": "26",
  "Classes": [
    { "Name": "tech", "Score": 1.0 },
    { "Name": "politics", "Score": 0.0 },
    { "Name": "business", "Score": 0.0 }
  ]
}
```

PARA SABER **MAIS**

<https://aws.amazon.com/pt/machine-learning/learn/>, Aprendizado de Machine Learning na AWS

<https://cursos.alura.com.br/course/machine-learning-utilizando-chatgpt-assistente>, Utilizando ChatGPT como assistente em Machine Learning

<https://cursos.alura.com.br/course/visao-computacional-analise-facial>, Análise facial

<https://cursos.alura.com.br/course/analise-classificacao-faces-visao-computacional-opencv>, Análise e classificação de faces

<https://cursos.alura.com.br/formacao-amazon-web-services>, Formação AWS

FIAP