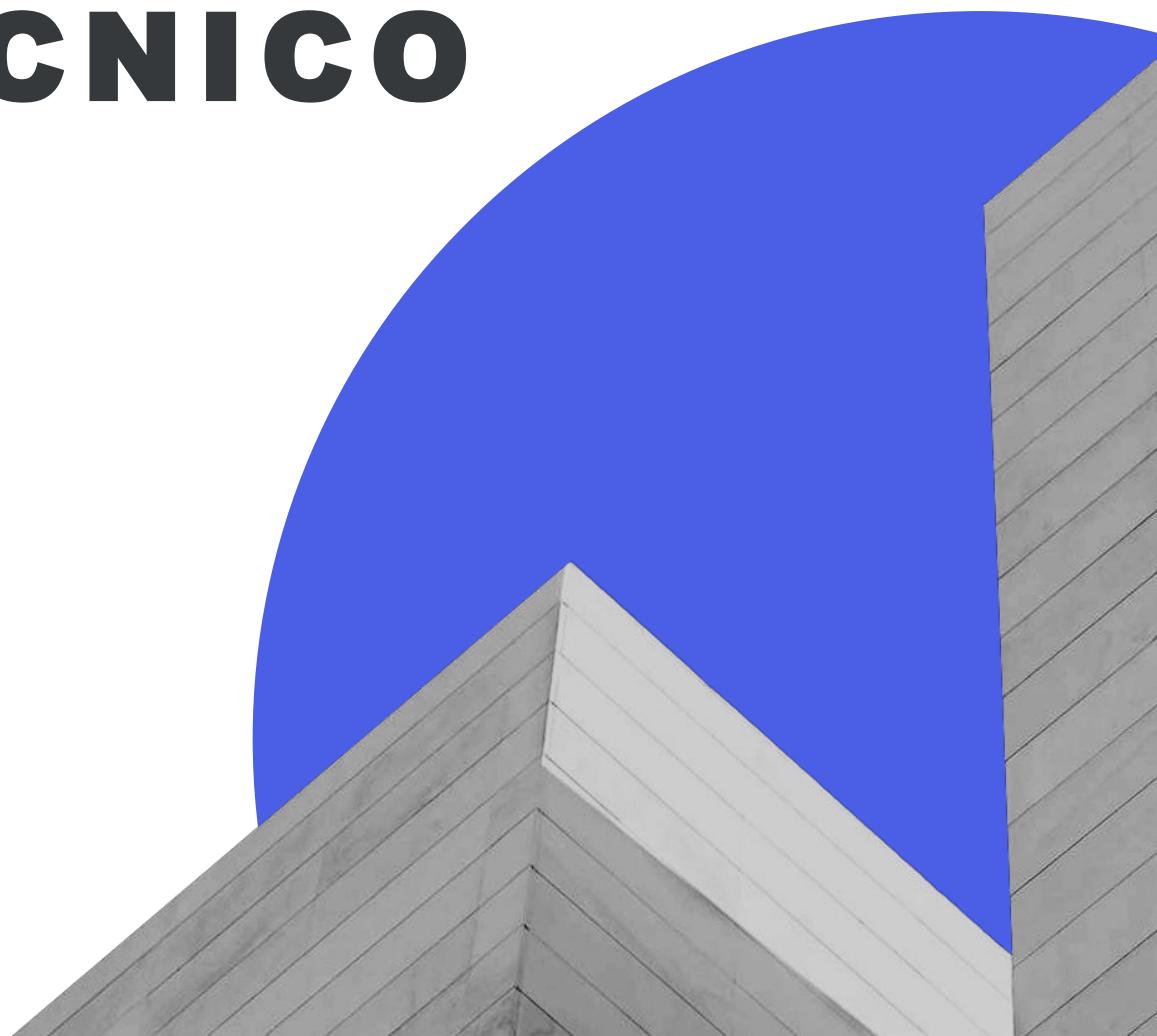
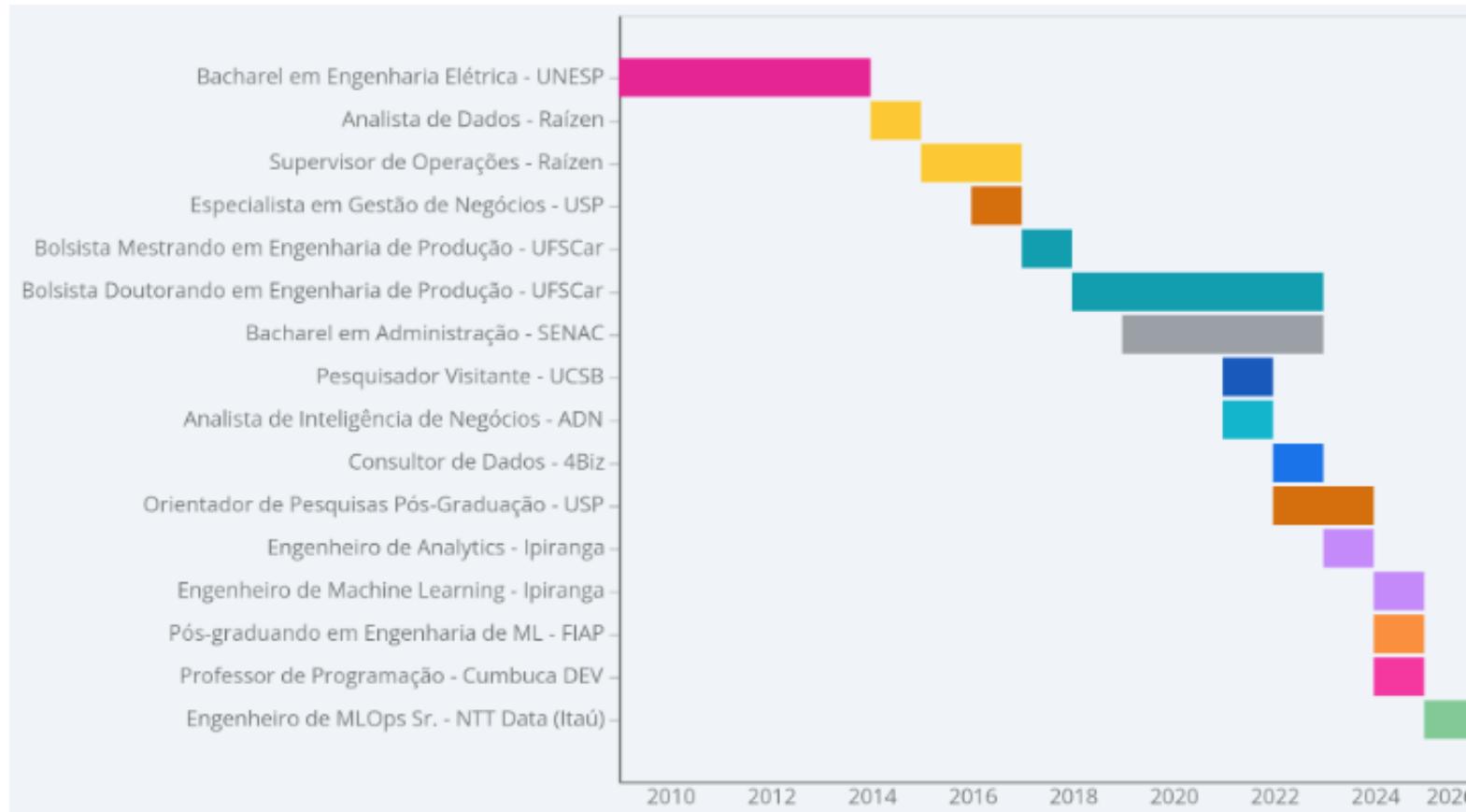


CASE IFOOD

# **DEEP DIVE TÉCNICO**



# BREVE APRESENTAÇÃO PESSOAL



Gus



# AGENDA

CONTEXTO E PROBLEMÁTICA

FLUXO DE TREINO - PT I

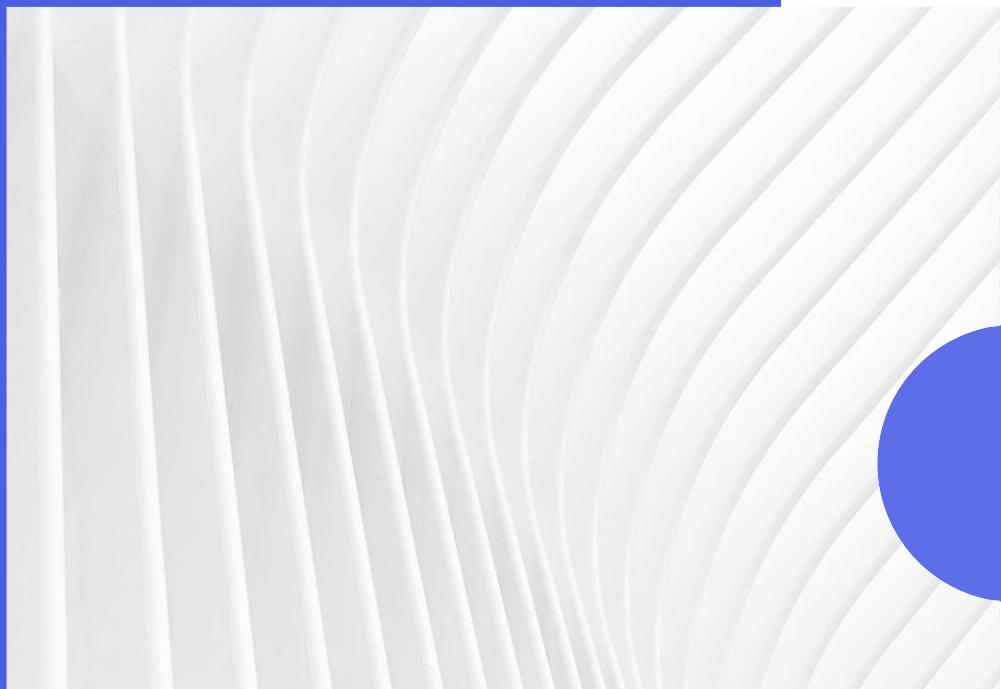
FLUXO DE TREINO - PT II

FLUXO DE PREDIÇÃO

DESAFIOS E APRENDIZADOS

CONSIDERAÇÕES FINAIS

# CONTEXTO



TCC da Pós em ML Engineer da FIAP,  
desenvolvido com dados reais  
disponibilizados pela **plataforma  
de notícias G1** em 2025.

Dois datasets com dados históricos  
anonimizados: um contendo **notícias** na  
íntegra do portal e outro com o **histórico  
de navegação** dos usuários.

O projeto foi executado de ponta a ponta  
ao longo de um período de **dois meses**.

# O PROBLEMA

CONSTRUIR UM SISTEMA DE RECOMENDAÇÃO  
PARA O G1 EFICIENTE E ESCALÁVEL

# DESAFIOS DE MODELAGEM



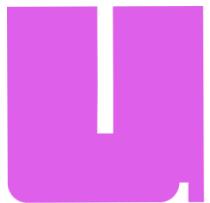
- **Gostos heterogêneos:** preferências de notícias variam drasticamente entre usuários (ex. política, esportes, etc.)
- **Cold start:** 8,5% dos usuários não-logados ou sem histórico de navegação
- **Recência:** notícias antigas perdem relevância muito rapidamente.
- **Target nebuloso:** assumimos engajamento como baseline, valorizando o histórico e a recência na construção de um score

# DESAFIOS DE CÓDIGO

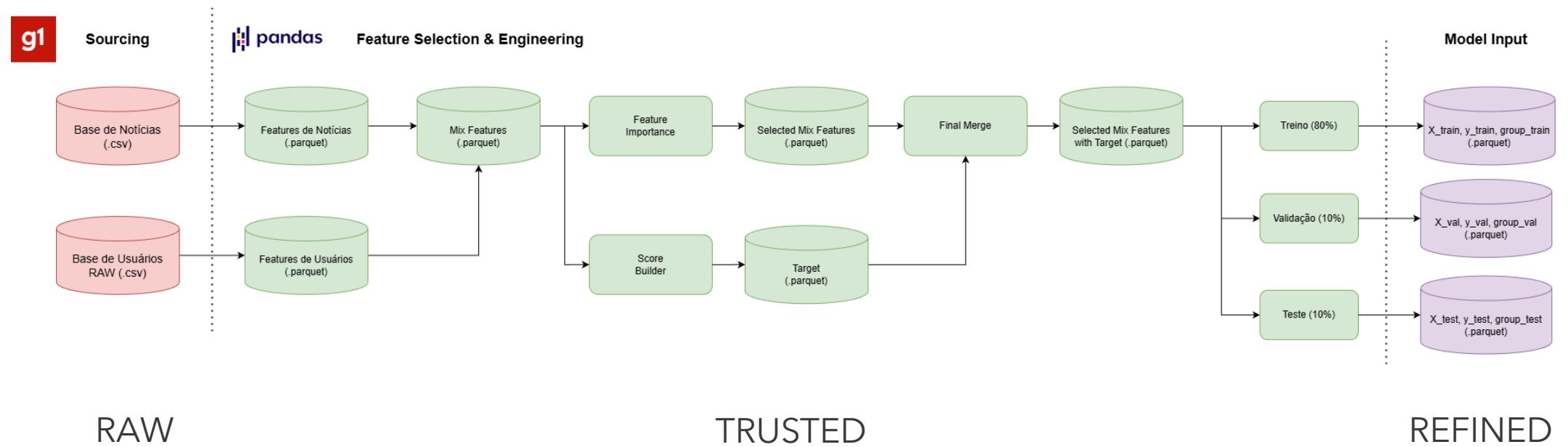


- **Fluxo end-to-end:** foi produtizado todo o fluxo desde a ingestão, feature engineering e modelagem; até a criação da API, integração com a cloud, dockerização e deploy.
- **Pipeline complexo:** orquestração de múltiplos serviços (MLFlow server, pipeline multistep, API)
- **Simulação de cenário real:** separação entre treino e inferência, promoção entre ambientes (dev = local, prd = cloud)

# A STACK



# FLUXO DE TREINO - PT I



# A INGESTÃO

## Usuários (600.000 registros)

**userType:** logado ou anônimo

**historySize:** quantas notícias consumiu

**history:** ID das notícias que consumiu [to be exploded]

**timeStampHistory:** momento que visitou a página

**timeOnPageHistory:** milissegundos que permaneceu

**numberOfClickHistory:** quantidade de cliques

**scrollPercentageHistory:** pct visualização da matéria

**pageVisitCountHistory:** vezes que visitou a mesma matéria

## Notícias (250.000 registros)

**Page:** ID da matéria

**URL:** link para a matéria

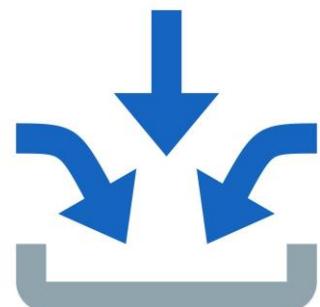
**Issued:** data que a matéria foi criada

**Modified:** data que a matéria foi modificada

**Title:** título da matéria

**Body:** conteúdo da matéria

**Caption:** subtítulo da matéria



# AS FEATURES



## Mix Feats (850.000 registros)

**userId:** ID do usuário

**newsId:** ID da notícia

**userType:** logado ou anônimo

**coldStart:** indica se o usuário tem histórico suficiente (split)

**localState:** estado associado à notícia

**localRegion:** região associada à notícia

**themeMain:** tema principal da notícia

**themeSub:** subtema da notícia

**issuedDatetime:** data e hora de publicação da notícia

**timestampHistoryDatetime:** momento em que o usuário  
consumiu a notícia

**isWeekend:** acesso ocorreu no fim de semana (sim ou não)

**dayPeriod:** período do dia (madrugada, manhã, tarde, noite)

**timeGapDays:** diferença em dias entre publicação e consumo

**relLocalState:** afinidade histórica do usuário com o estado

**relLocalRegion:** afinidade histórica do usuário com a região

**relThemeMain:** afinidade histórica com o tema principal

**relThemeSub:** afinidade histórica com o subtema

**interpretabilidade**

**features de fato**

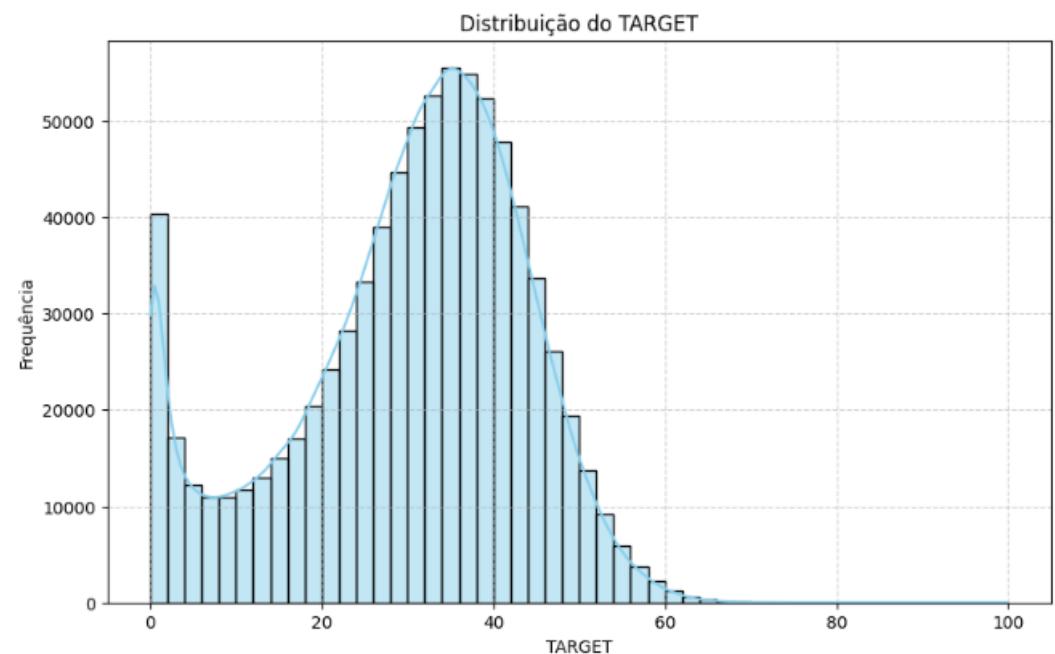
# O TARGET

$$\text{scoreBase} = \text{numberOfClicksHistory} + 1.5 \cdot \left( \frac{\text{timeOnPageHistory}}{1000} \right) + \text{scrollPercentageHistory} - \left( \frac{\text{minutesSinceLastVisit}}{60} \right)$$

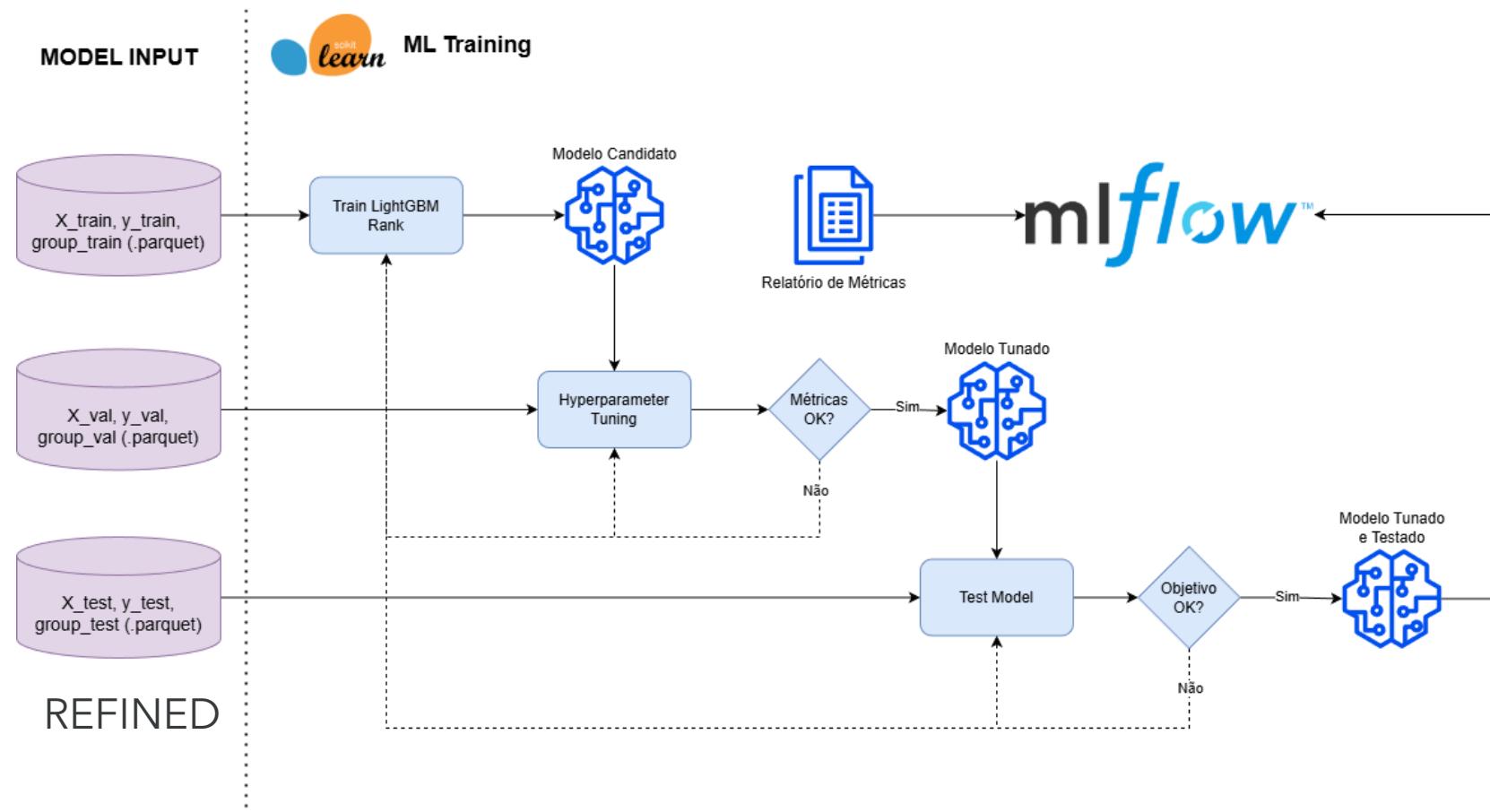
$$\text{rawScore} = \text{scoreBase} \cdot \left( \frac{\text{historySize}}{130} \right) \cdot \left( \frac{1}{1 + \left( \frac{\text{timeGapDays}}{50} \right)} \right)$$

$$\text{finalScore} = \text{MinMaxScale}(\log(1 + \text{rawScore}))$$

Valor de 0 a 100 (geralmente até 60) indicando o nível de engajamento do usuário com a notícia

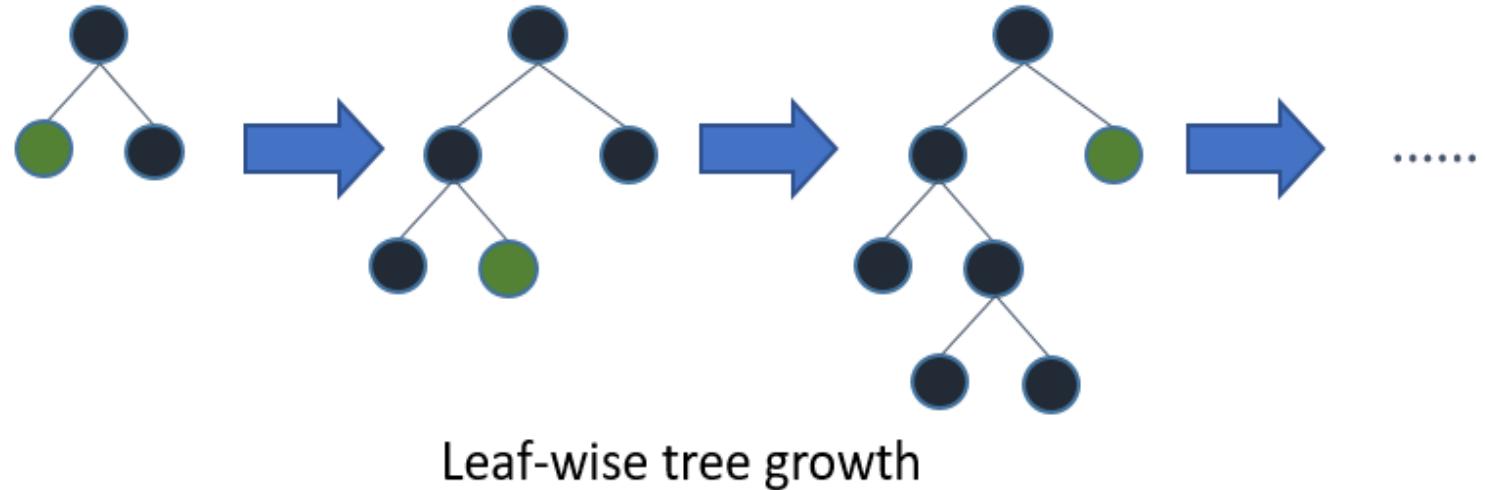


# FLUXO DE TREINO - PT II



# O MODELO

**LightGBMRanker:** modelo de boosting com árvores de decisão otimizadas para a função lambdarank (métrica de desempenho NDCG). Ao invés de aprender a prever um valor ou classe, aprende a ordenar.



# O MODELO

- **Cada usuário:** pertence a um grupo de ranking próprio
- **Cada registro:** do nosso dataframe representa um par usuário-notícia
- **O target:** representa o score provável de engajamento do usuário para aquela notícia



# O MODELO - EXEMPLO

ID

userId	Name	newsId	title
U123	Robert	N456	Nova API de IA lançada
U123	Robert	N457	Tutorial Docker para produção
U123	Robert	N458	Novo smartphone chega ao Brasil
U123	Robert	N459	Política econômica do estado
U123	Robert	N460	Festival cultural regional

features

timeGapDays	isWeekend	localState	localRegion	themeMain	themeSub
0	0	SP	Sudeste	Tecnologia	IA
1	0	SP	Sudeste	Tecnologia	DevOps
2	0	SP	Sudeste	Tecnologia	Dispositivos
4	0	SP	Sudeste	Política	Economia
12	1	BA	Nordeste	Cultura	Cinema

target

Engajamento
60
48
36
12
4

Retorna newsId nesta ordem!



**Robert**  
paulistano  
IT guy

# O REGISTRO DO MODELO

The screenshot shows the mlflow UI interface. At the top, there is a navigation bar with the mlflow logo (2.20.1), Experiments, Models (selected), GitHub, and Docs. Below the navigation bar, the page title is "Registered Models > news-recommender-dev". The model details are as follows:

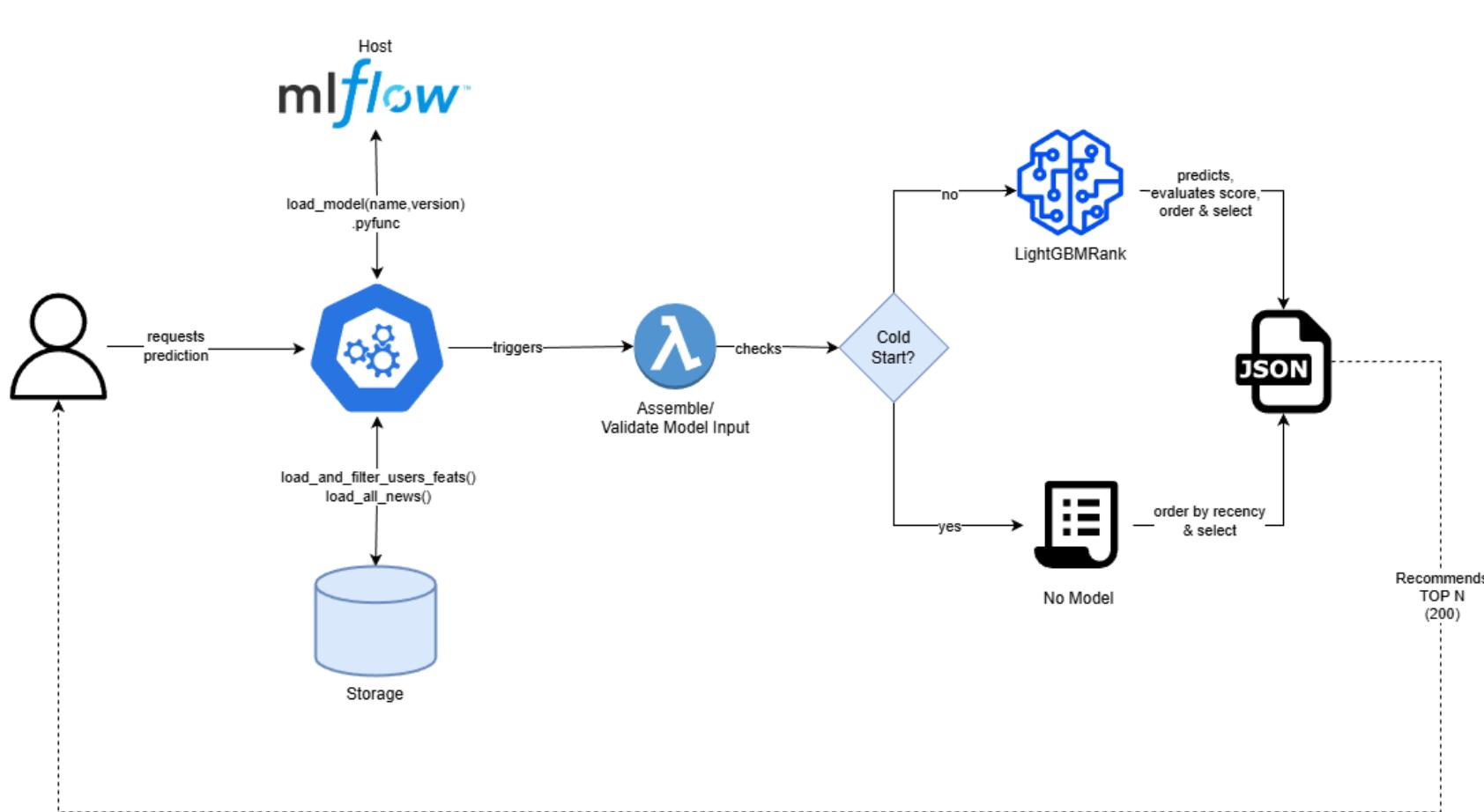
- Created Time: 2025-02-28 16:29:34
- Last Modified: 2026-02-12 20:16:55

The "Versions" section is expanded, showing the following table:

Version	Registered at	Created by	Tags	Aliases	Description
Version 16	2026-02-12 17:56:40		stage: PRD	@ champion	
Version 15	2026-02-12 17:55:01		stage: STG	@ candidate1	
Version 14	2026-02-12 17:48:39		stage: STG	@ candidate2	
Version 13	2025-03-02 18:30:39		stage: STG	@ candidate3	
Version 12	2025-03-02 18:25:32		stage: DEV	@ dev_experiment_01	
Version 11	2025-03-02 18:22:08		stage: DEV	@ dev_experiment_02	
Version 10	2025-03-02 18:18:50		stage: DEV	@ dev_experiment_03	
Version 9	2025-03-02 02:05:15		stage: DEV	@ dev_experiment_04	

A "Compare" button is located next to the Versions heading. A "New model registry UI" toggle switch is also present.

# FLUXO DE INFERÊNCIA



DEV =



PRD =



Amazon  
EC2



Amazon  
EC2



Amazon  
S3

# ENDPOINTS

## Monitoring

<b>GET</b>	/health	Health Check	>
<b>GET</b>	/info	Model Info	<

## Prediction

**POST** /predict Predict >

**Parameters** Cancel

No parameters

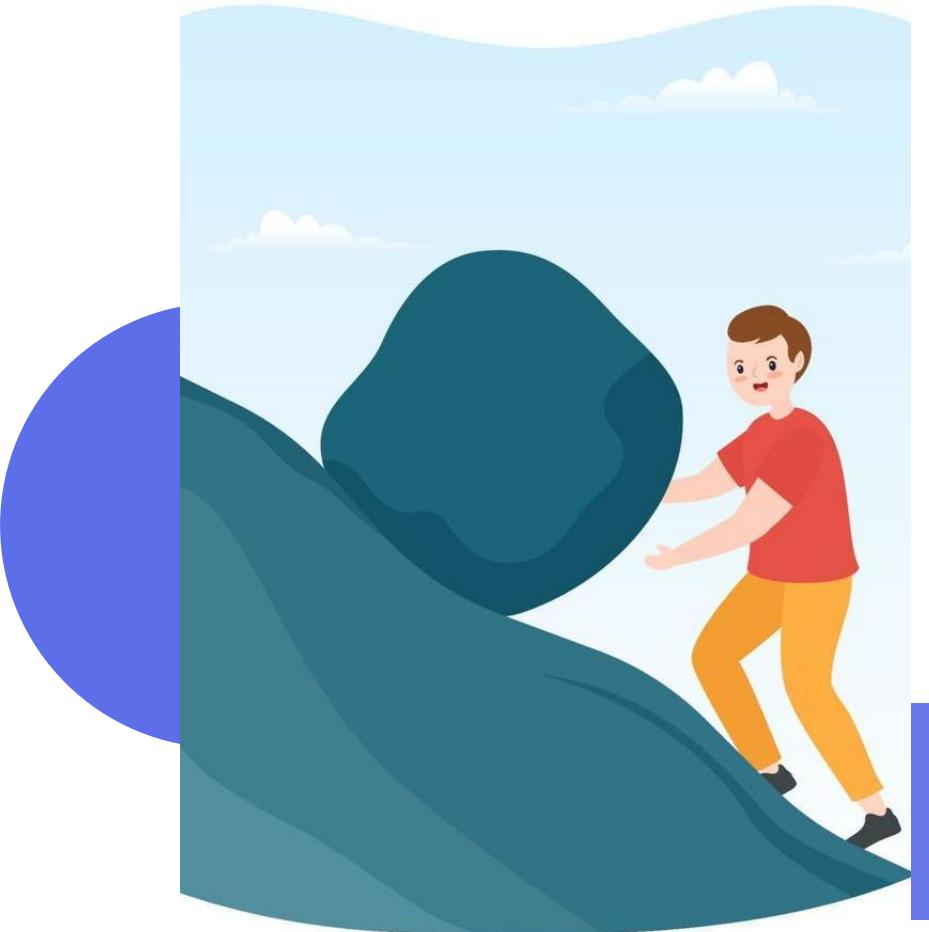
**Request body** required application/json

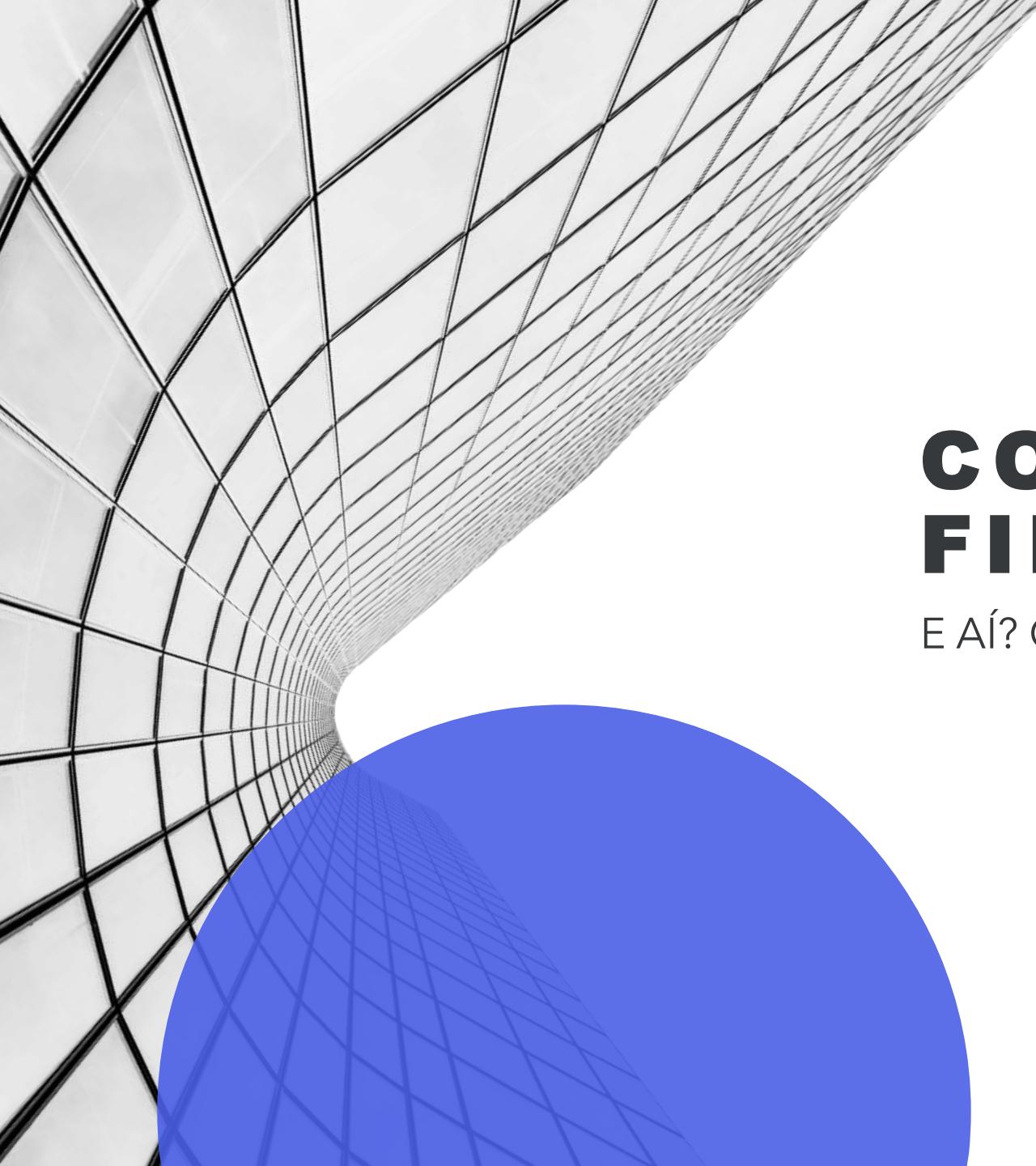
```
{  
  "userId": "4b3c2c5c0edaf59137e164ef6f7d88f94d66d0890d56020de1ca6af55b4f297",  
  "max_results": 5,  
  "minScore": 0.3  
}
```

# **DESAFIOS & APRENDIZADOS**

Percepção tardia de que precisávamos segregar as notícias visualizadas e não-visualizadas por aquele usuário.

Limitações de Arquitetura por estarmos usando AWS Lab ao invés de uma conta comum (ex. ECS Fargate e Sagemaker não disponíveis para uso).





# **CONSIDERAÇÕES FINAIS**

E AÍ? O QUE VIROU?

# RESULTADOS

## Banca Avaliadora:

- Felipe Ferreira (*Staff Data Scientist - Stone*)
- Leandro Zanon (*Lead Data Scientist - Itaú*)
- Ana Raquel Fernandes Cunha  
(*Coordenadora do Pós-Tech FIAP*)

Mais de 60 trabalhos avaliados, 5 finalistas,  
ganhamos o 1º lugar.



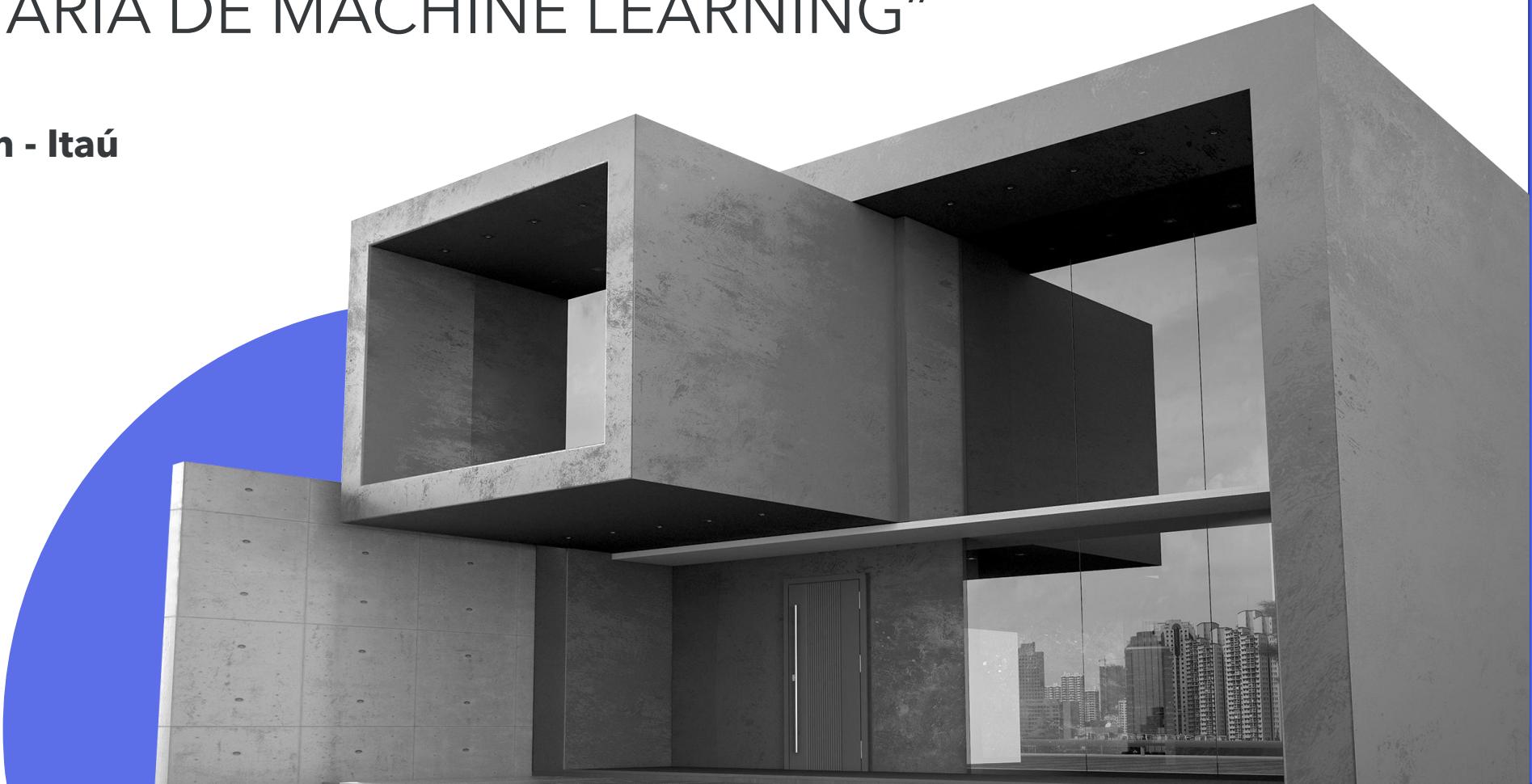
**"COM POUQUÍSSIMOS AJUSTES,  
CONSEGUIRÍAMOS COLOCAR O TRABALHO  
DE VOCÊS HOJE EM PRODUÇÃO".**

**Felipe Ferreira - Stone**

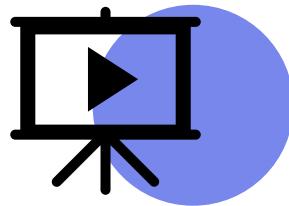


**"FOI O TRABALHO MAIS CUIDADOSO E QUE  
MELHOR EQUILIBROU CIÊNCIA DE DADOS E  
ENGENHARIA DE MACHINE LEARNING"**

**Leandro Zanon - Itaú**

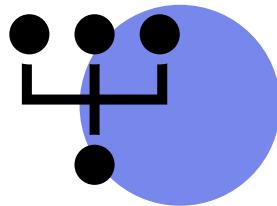


# RECURSOS ADICIONAIS



## VÍDEO NO YOUTUBE

Vídeo da apresentação final disponibilizado para a banca avaliadora ([link](#)).



## REPOSITÓRIO NO GITHUB

Repositório com todo o código-fonte do projeto ([link](#)).



## ARTIGO NO MEDIUM

Artigo explicando os pormenores das decisões internas do nosso projeto ([link](#)).

# OBRIGADO

Gustavo Mendonça Ferratti

/in/gmferratti

+55 16 99643 0580