DCA0305

ivanovitch.silva@ufrn.br

Machine Learning Based Systems Design

Beginning of the Journey

Linear Algebra & Math

Matrices & vector arithmetics, types, operations factorization, derivatives

Probability & Statistics

Cond. prob., distributions, bayesian prob., data viz., central limit theorem, hypothesis tests, correlation, resampling methods

Data Science

EDA, measurements of centrality (mean, mode, median, variand, std, z-score), data pipeline

Machine Learning

Supervised Learning (KNN, Linear Regression, Logistic Regression, DT, RF, Ensemble, XGBoost, MLP), Unsupervised Learning (K-means, PCA)

Deep Learning

Fundamentals of Deep Learning, Better Generalization vs Better Learning, Hyperparameter Tuning, Batch Normalization, CNN, Transfer Learning

TinyML

Optimization, quantization, deploy into a microcontroller

New Models

Transformers, Difusion, GAN, LLM, GNN, Generative Flow Networks

But ....

If a model is not deployed, it does not generate value

Cont.

2016

Pandas

2017

Scikit-Learn

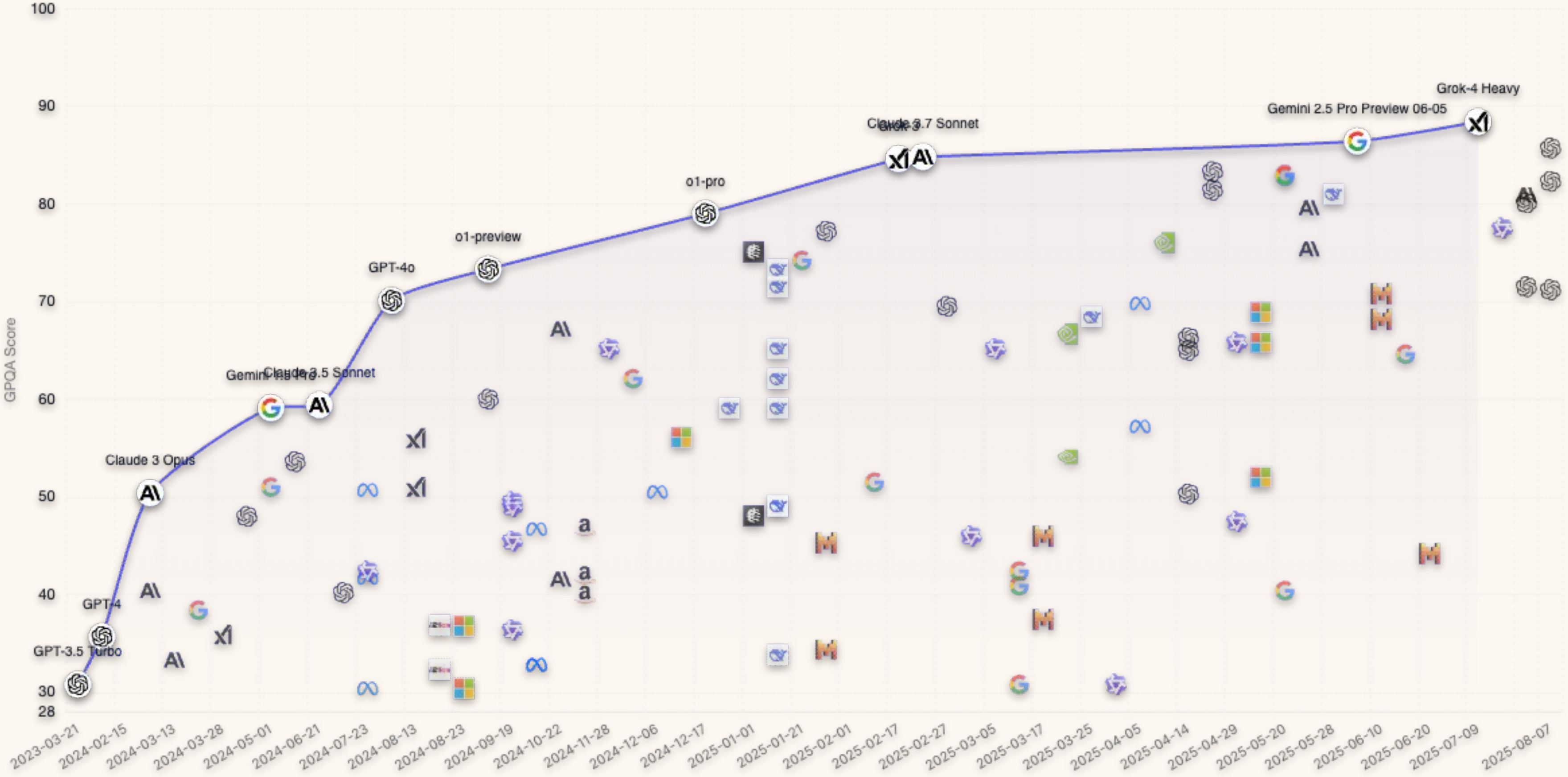
2018-2020

TensorFlow, Keras, Pytorch, Covid-19

2021-2023

MLOps, GAN, LLM, GPT-3, OpenAI

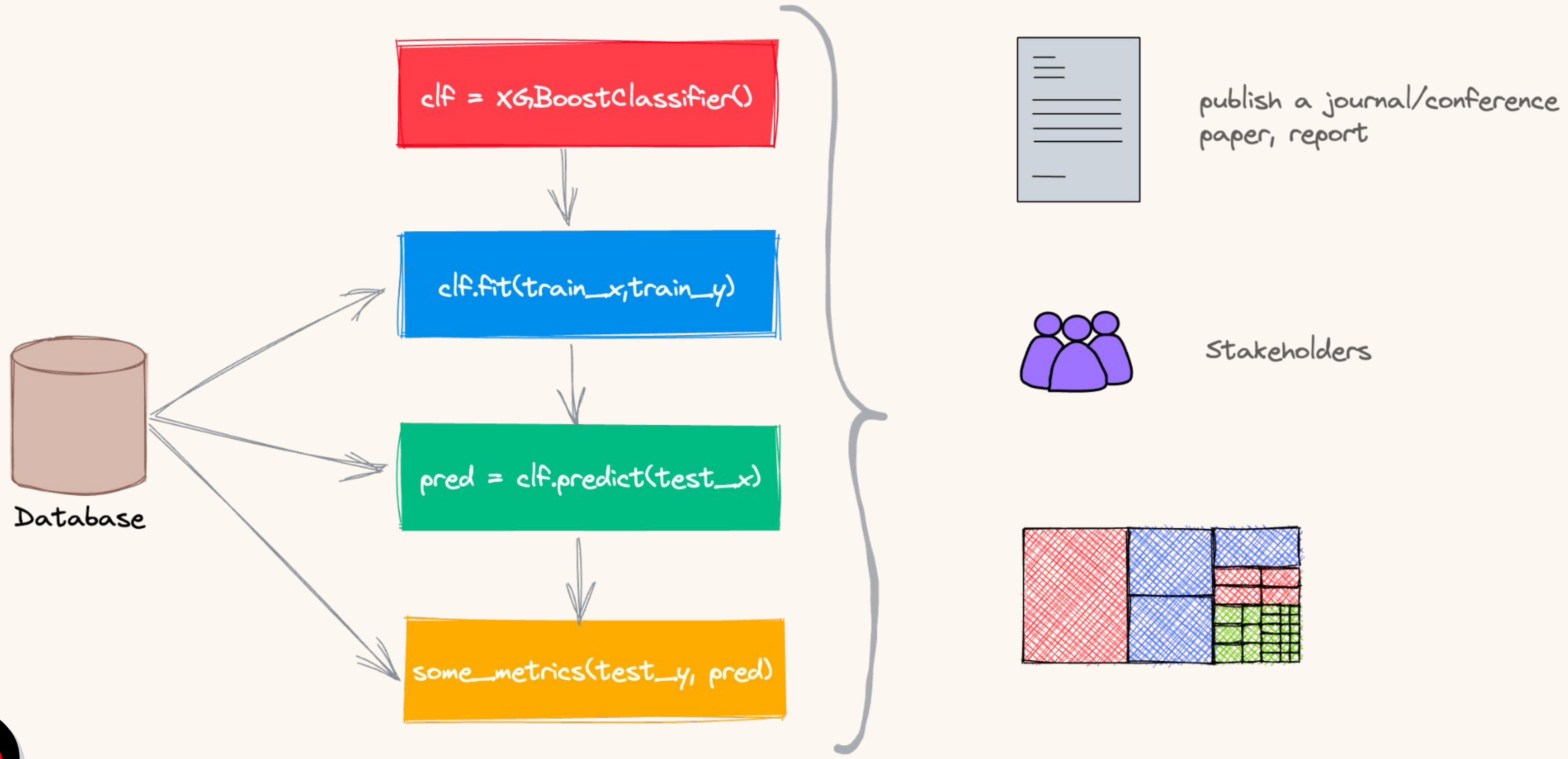
LLM Leaderboard & Vibe Coding



https://llm-stats.com

Last update: 16 august 2025

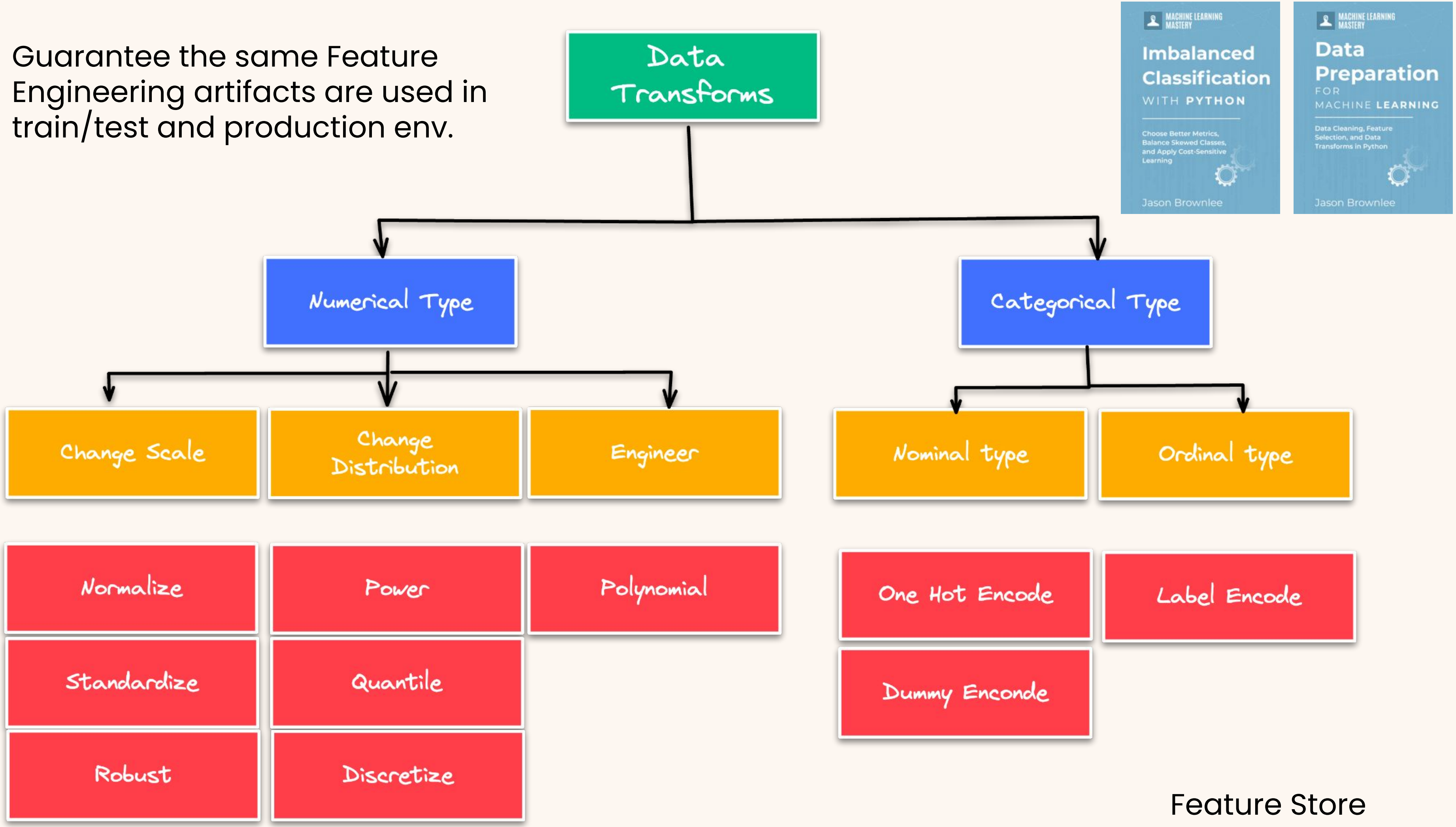
How to project a typical machine learning workflow?



What is the issue with this solution?

Automate data preprocessing
Store preprocessing statistics (e.g., normalization)
Track generated artifacts
Deploy model to production
Document process

Guarantee the same Feature Engineering artifacts are used in train/test and production env.



Feature Store

education sciences

MDPI

Article

Predictive Models for Imbalanced Data: A School Dropout Perspective

Thiago M. Barros [1,*,†], Plácido A. Souza Neto [1,†] and Ivanovitch Silva [2,†] and Luiz Affonso Guedes [2,†]

1 Federal Institute of Rio Grande do Norte (IFRN), 1559 Tirol Natal, Brazil; placido.neto@ifrn.edu.br
2 Federal University of Rio Grande do Norte (UFRN), 59078-970 Natal, Brazil; ivan@imd.ufrn.br (I.S.); affonso@dca.ufrn.br (L.A.G.)
* Correspondence: thiago.medeiros@ifrn.edu.br
† These authors contributed equally to this work.

check for updates

Received: 24 July 2019; Accepted: 10 November 2019; Published: 15 November 2019

Abstract: Predicting school dropout rates is an important issue for the smooth execution of an educational system. This problem is solved by classifying students into two classes using educational activities related statistical datasets. One of the classes must identify the students who have the tendency to persist. The other class must identify the students who have the tendency to dropout. This problem often encounters a phenomenon that masks out the obtain
into this phenomenon and provides a reliable educational data minir
predicts the dropout rates. In particular, the three data classifying techn
neural networks and Balanced Bagging, are used. The performances of t
and without the use of a downsample, SMOTE and ADASYN data bala
other parameters geometric mean and UAR provides reliable results w
rates using Balanced Bagging classifying techniques.

Keywords: dropout rates; accuracy paradox; imbalanced learning; d
mlp; decision tree; Balanced Bagging; UAR; SMOTE; ADASYN

Concept/
Data Drift

Evasão escolar de crianças e adolescente aumenta 171% na pandemia, diz estudo

Levantamento da organização Todos Pela Educação mostra que 244 mil crianças de 6 a 14 anos estavam fora da escola no segundo trimestre de 2021.

Por g1 — São Paulo
02/12/2021 13h28 · Atualizado há um ano

g1

Pandemia aumenta evasão escolar, diz relatório do Unicef

A quantidade de alunos, com idades entre 6 e 17 anos, que abandonaram as instituições de ensino foi de 1,38 milhão

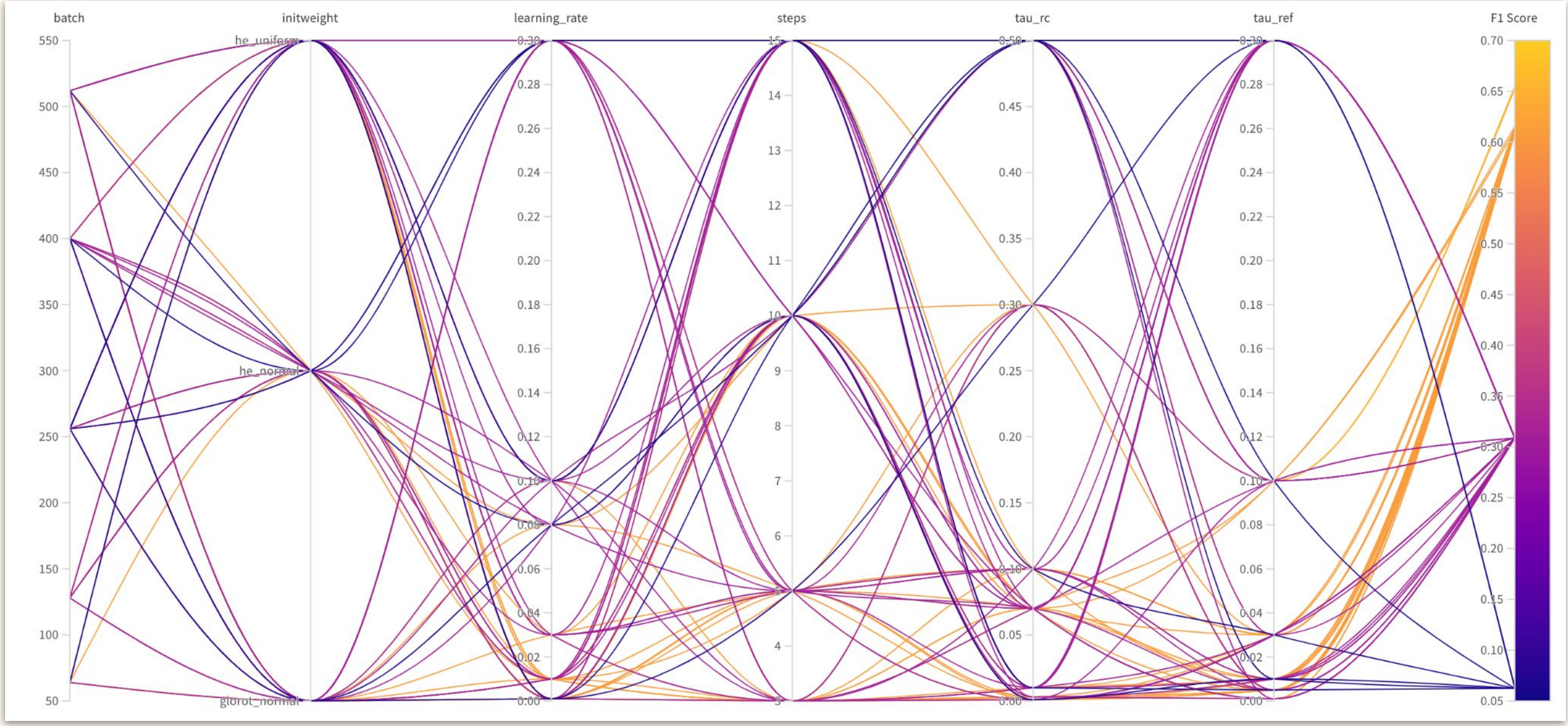
Educadores alertam para aumento de evasão escolar durante a pandemia

Para debatedores, desafio agora é atrair estudantes de volta à escola e recuperar o aprendizado

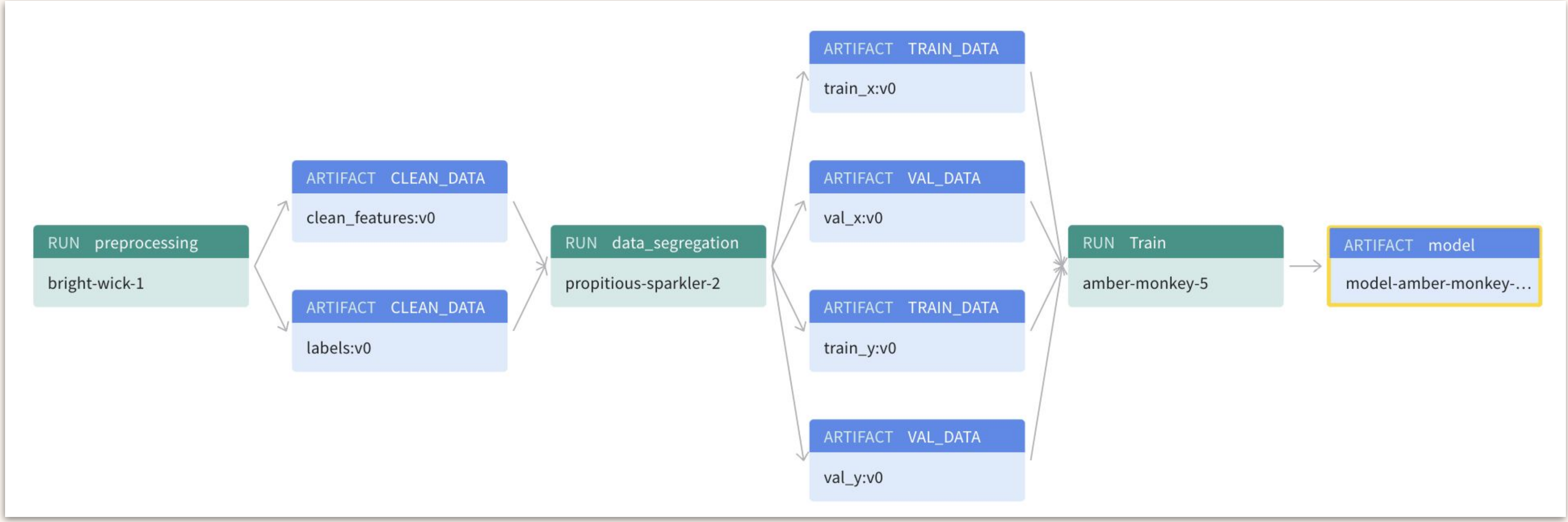
CÂMARA DOS DEPUTADOS

06/10/2021 - 20:04

What settings were used in the last experiment?



A more efficient machine learning workflow



FEATURE ARTICLE

Advancing Tiny Machine Learning Operations: Robust Model Updates in the Internet of Intelligent Vehicles

Thommas K. S. Flores, Ivanovitch Silva, Mariana B. Azevedo, Thaís de A. de Medeiros, and Morsinaldo de A. Medeiros, Federal University of Rio Grande do Norte, Natal, RN, 59078–970, Brazil

Daniel G. Costa, University of Porto, 4200-465, Porto, Portugal

Paolo Ferrari and Emiliano Sisinni, University of Brescia, 25123, Brescia, Italy

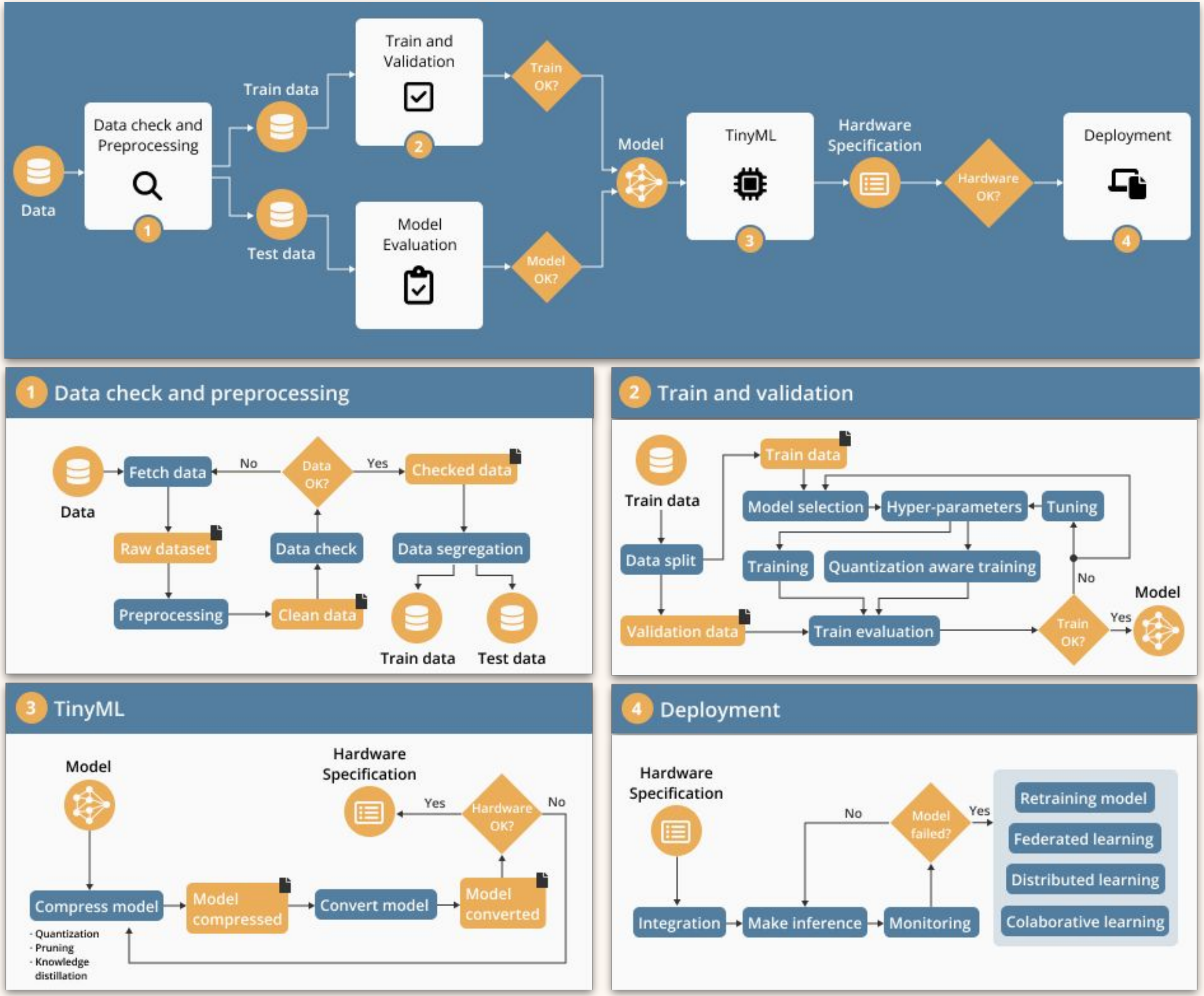
The Internet of Intelligent Vehicles is becoming increasingly important, and embedded machine learning is gaining popularity due to new development paradigms. However, the demand for machine learning model updates on embedded systems has become relevant in multiple scenarios. This article proposes a methodology for tiny machine learning operations within the context of the Internet of Intelligent Vehicles, utilizing affordable microcontrollers based on the ESP32 platform. The solution presented in the article consists of two ESP32 devices: one functioning as a radio station (RS) and the other as the microcontroller of an onboard diagnostic (OBD-II) scanner. The RS hosts the updated model and transmits it to the OBD-II scanner using the Espressif Systems Peer-to-Peer Over Wi-Fi communication protocol over 802.11 Wi-Fi. Experimental results demonstrate significant improvement in model performance postupdate, but the article also identifies critical challenges to model robustness because of the use of the interpreter method on microcontrollers.

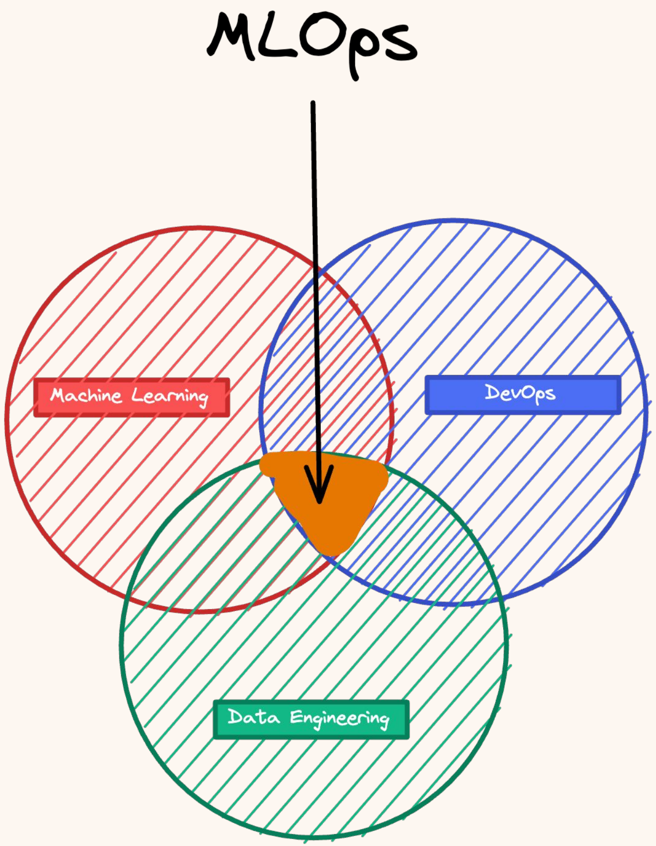
The Internet of Intelligent Vehicles (IoIV) is a concept that involves integrating vehicles with the Internet and advanced technologies to enhance their functionality, safety, and overall performance. This idea is a subset of the broader Internet of Things (IoT) paradigm, connecting everyday vehicles to the Internet for improved communication and data exchange. With features like high-speed connectivity, advanced sensors for environmental perception, and real-time data sharing, the IoIV aims to optimize traffic, improve road safety, and facilitate autonomous driving. Additionally, it contributes to efficient traffic management, provides personalized user experiences, and raises important considerations regarding cybersecurity and public acceptance of autonomous technologies.

Recently, machine learning (ML) algorithms have been gaining ground in automotive applications since they can enhance a large set of processing tasks that require very quick decisions. In this sense, the need to allow the execution of such algorithms in resource-constrained devices in embedded scenarios has also led to the creation of the tiny ML (TinyML) paradigm, with particular challenges concerning the size of the models and the processing of real-time data. Overall, the greatest innovation of the TinyML paradigm is the capacity to perform data inference through ML algorithms in tiny low-power devices,[1] which has opened a new trend of development that was not previously possible when implementing traditional ML algorithms. For the Internet of Vehicles paradigm, this has been an important revolution, although many challenges still remain when bringing TinyML to a more realistic setting.

The recent literature has presented some works that address TinyML as an effective solution for the construction of intelligent vehicles. For example, Flores et al.[2] developed a soft-sensor to estimate carbon dioxide (CO_2) emissions in vehicles using the TinyML paradigm. Additionally, Shauot et al.[3] compared several ML models embedded in different hardware to detect driver distraction from their position while driving. In

0272-1732 © 2024 IEEE
Digital Object Identifier 10.1109/MM.2024.3354323
Date of publication 17 January 2024; date of current version 21 February 2025.





Machine Learning/Deep Learning

CLI Fundamentals	Clean Code Principles	Production Ready Code	Programming	Introducton to MLOps	Model Management
Elements of the Command Line	Refactoring Documentation	Catching Errors Logging	Functions Classes Decorators	Experiment Tracking	Orchestration and ML Pipelines
Infrastructure Github Codespace, vscode, colab, terminal	Python Code Quality Authority (PCQA)	Testing	Interact with APIs and SDKs to build CLI tools	Model Deployment	Model Monitoring

MLOps course

*Advanced Undergrad

Agosto - 2025						
D	S	T	Q	Q	S	S
					1	2
3	4	5	6	7	8	9
10	**11**	12	**13**	14	15	16
17	**18**	19	**20**	21	22	23
24	**25**	26	**27**	28	29	30
31						

Setembro - 2025						
D	S	T	Q	Q	S	S
	1	2	**3**	4	5	6
7	**8**	9	**10**	11	12	13
14	**15**	16	**17**	18	19	20
21	**22**	23	**24**	25	26	27
28	**29**	30				

Outubro - 2025						
D	S	T	Q	Q	S	S
			1	2	3	4
5	**6**	7	**8**	9	10	11
12	**13**	14	**15**	16	17	18
19	**20**	21	**22**	23	24	25
26	27	28	**29**	30	31	

Novembro - 2025						
D	S	T	Q	Q	S	S
						1
2	**3**	4	**5**	6	7	8
9	**10**	11	**12**	13	14	15
16	**17**	18	**19**	20	21	22
23	**24**	25	**26**	27	28	29
30						

Dezembro - 2025						
D	S	T	Q	Q	S	S
	1	2	**3**	4	5	6
7	**8**	9	**10**	11	12	13
14	**15**	16	**17**	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Unit 01

11/08 to 24/09

Week 01
11/08 and 13/08 - Planning

Week 02
18/08 - Course Presentation
20/08 - Visualizing Gradient Descendent

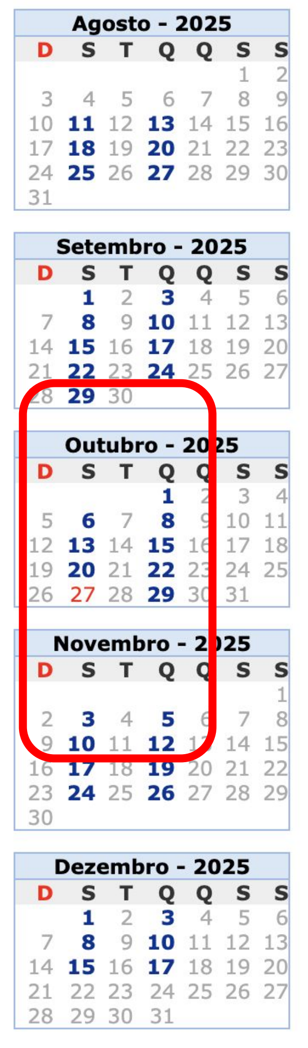
Week 03
25/08 - Rethinking the training loop
27/08 - Clean Code Principles, Documentation, Tooling, and Going Classy

Week 04
01/09 - A simple classification problem
03/09 - [Project #01] - Description

Week 05
08/09 - [Project #01] doing
10/09 - [Project #01] doing

Week 06
15/09 - [Project #01] doing
17/09 - [Project #01] doing

Week 07
22/09 - [Project #01] Presentation
24/09 - [Project #01] Presentation



Unit 02

29/09 to 12/11

Week 08
29/09 - From a shallow to a deep-ish classification model - Part I
01/10 - From a shallow to a deep-ish classification model - Part II

Week 09
06/10 - Handling Errors, writing tests and logs
08/10 - Convolutions (CNN)

Week 10
13/10 - Multiclass problem, hooks
15/10 - Rock, paper and scissor - Part I

Week 11
20/10 - Rock, paper and scissor - Part II
22/10 - [Project #02] - Description

Week 12
27/10 - no class
29/10 - [Project #02] doing

Week 13
03/11 - [Project #02] doing
05/11 - [Project #02] doing

Week 14
10/11 - [Project #02] Presentation
12/11 - [Project #02] Presentation

Unit 03

17/11 to 17/12

Week 15
17/11 - Frameworks & Tools
19/11 - Frameworks & Tools

Week 16
24/11 - Frameworks & Tools
26/11 - [Project #03] Description

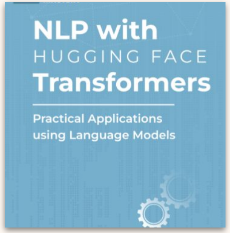
Week 17
01/12 - [Project #03] Doing
03/12 - [Project #03] Doing

Week 18
08/12 - [Project #03] Doing
10/12 - [Project #02] - Doing

Week 19
15/12 - [Project #03] Presentation
19/12 - Final Exam

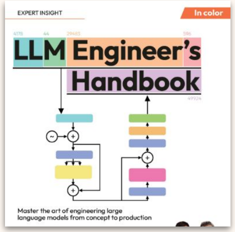
References

Main references: from MLOPs to LLMOps

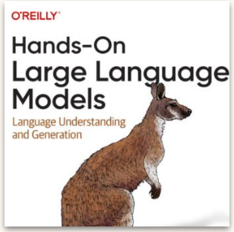


May, 2025

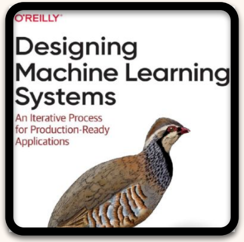
Jan, 2025



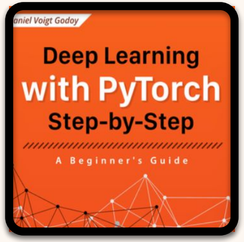
Oct, 2024



Sep, 2024



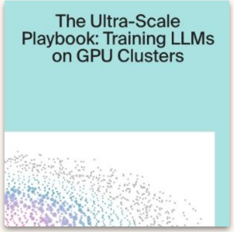
May, 2022



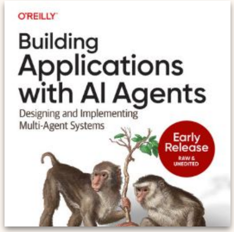
Feb, 2022

References

Further References: from LLM to Agents

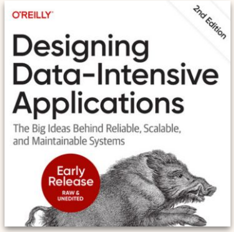


Jul, 2025



Sep, 2025

Dec, 2025



Jan, 2026

Clone me!!!!

https://github.com/ivanovitchm/mlops