

This is my personal diary which contains resources, which I know, learned or people have told me to experiment with. This is the second part of the diary. First diary is 392 pages and it is upto 16-Dec-22.

<https://dasarpai.com/dsresources/ai-ml-resources-from-my-diary>

This diary is related to my work/learning in data science, AI, ML, NLP, DL, GNN, GAN, Statistics, etc. A few links related to Software Development, People Management, Project Management are also kept in this diary. The resources here are python/R library link, blog articles, YouTube video links, article links, AI products links, architecture, images etc. The Table of Content and Content of this diary is not any specific order. Whatever I was/am getting I keep adding into this, mostly towards the end but not always.

You can connect with me on linkedin: <https://www.linkedin.com/in/harithapliyal/>

My blog: <https://dasarpai.netlify.app>, <https://dasarpai.com>

You can refer my github repo about AI resource: [DataScience-Github-Repo](#)

I didn't put any effort into making this document grammatically and spelling correct. So, there may be grammar and spelling errors.

#####

Date Updated: Thu, 15-Dec-22 to Thu, 30-Nov-23

Table of Contents

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

1. [In my Control and Out of My Control](#)
2. [The Mathematical Modeling of Black Swan Events](#)
3. [Analyzing Transformers in Embedding Space Explained](#)
4. [Multimodality is the future! Here are 10 multimodal models that can be used to build futuristic #AI apps:](#)
5. [2000+ new AI tools were released in the last 30 days.](#)
6. [Generative AI, OpenAI, and ChatGPT: What are these?](#)
7. [MIT university is offering FREE education in Data Science.](#)
8. [A Complete Guide to Creating and Storing Embeddings for PostgreSQL Data](#)
9. [A New Era of GPTs: 10 Game-Changing Creations](#)
10. [Comparing the Performance of LLMs: A Deep Dive into Roberta, Llama 2, and Mistral for Disaster Tweets Analysis with Lora](#)
11. [Multi-Modal RAG : Llamalndex Blog](#)
12. [CS50's Introduction to Artificial Intelligence with Python](#)
13. [Artificial Intelligence in Business: Creating Value with Machine Learning](#)
14. [CS50's Introduction to Computer Science](#)
15. [Data Analytics can be divided into 5 types based on the questions it can answer.](#)
16. [Learn the power of LangChain: A comprehensive guide to building custom Q&A chatbots](#)
17. [LLM Collection | Prompt Engineering Guide](#)
18. [In Machine learning. We have 2 types of errors: Reducible and Irreducible.](#)
19. [Data characteristics](#)
20. [Autoencoders are Neural Networks!](#)
21. [Best YouTube Courses for Programming:](#)
22. [Mathematics for Machine Learning](#)

- 23. [How to Learn System Design?](#)
- 24. [Here's a list of favorite recent LLM papers to read:](#)
- 25. [Language Translation using PyTorch Transformer](#)
- 26. [How to Fine-tune LayoutLMv3](#)
- 27. [Malaria Classification with Vision Transformer and PyTorch](#)
- 28. [Generative AI Models vs Discriminative AI Models](#)
- 29. [Infographic: 50 Cognitive Biases in the Modern World](#)
- 30. [MIT Researchers Introduce PFGM++](#)
- 31. [I Scanned 1000+ Prompts so You Don't Have to: 10 Need to Know Techniques](#)
- 32. [The ChatGPT list of Lists: A collection of 3000+ Prompts, examples, usecases](#)
- 33. [Why We Should Be Careful Using One-Tailed Hypothesis Tests in Research](#) 
- 34. [Top Large LLM Interview Questions & Answers](#)
- 35. [Tough SQL questions](#)
- 36. [Graph Convolutional Networks: Introduction to GNNs](#)
- 37. [Relationships Between t and F, and z and Chi-square](#)
- 38. [You can now fine-tune an LLM without writing a single line of code!](#)
- 39. [Introducing no-code LLM FineTuning with Monster API](#)
- 40. [BERT applications you should know about:](#)
- 41. [Revolutionizing Language Models Unleashing the power of retrieval augmented generation rag](#)
- 42. [Code Llama | NVIDIA NGC](#)
- 43. [Agriculture usecases](#)
- 44. [dhblog - Why and How to Fine-tune CLIP](#)
- 45. [ICML 2023 Tutorials](#)
- 46. [Similarity Algorithms :: TigerGraph Graph Data Science Library](#)
- 47. [TQDM UseCases](#)
- 48. [When Companies were founded](#)
- 49. [Enterprises are deploying LLMs in these four ways.](#)
- 50. [Exploring the promise of generative models in chemistry an introduction to diffusion-models](#)
- 51. [Graph of Thoughts: Solving Elaborate Problems with Large Language Models](#)
- 52. [The Definitive Guide to Einstein GPT | Salesforce Ben](#)
- 53. [LLM: A CLI utility and Python library for interacting with Large Language Models](#)
- 54. [Tensor Algebra and its Applications to Data Science and Statistics](#)
- 55. [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)
- 56. [Retrieval-Augmented Multimodal Language Modeling](#)
- 57. ["GraphGPT converts unstructured natural language into a knowledge graph."](#)
- 58. [Deep Bidirectional Language-Knowledge Graph Pretraining](#)
- 59. [Code Llama: Open Foundation Models for Code](#)
- 60. [Introducing Code Llama, a state-of-the-art large language model for coding](#)
- 61. [Salesforce+ Salesforce AI Day: Calling all Trailblazers](#)
- 62. [Tutorials and Unpublished Papers](#)
- 63. [Salesforce's CodeT5 system can understand and generate code](#)
- 64. [Tryage: Real-time, intelligent Routing of User Prompts to Large Language Model](#)
- 65. [LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models through Parameter-Efficient Fine-Tuning \(Practical Experience Report\)](#)
- 66. [MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models](#)
- 67. [Dive into LoRA Adapters](#)
- 68. [AutoGen: Enabling Next-Gen GPT-X Applications | FLAML](#)
- 69. [Aman's AI Journal • Reinforcement Learning from Human Feedback \(RLHF\)](#)
- 70. [The Role of AI in Biomarker-Driven Oncology Clinical Trials](#)
- 71. [What is Sensor Data? Examples of Sensors and Their Uses | Definition from TechTarget](#)
- 72. [Img2Img: Stable Diffusion Image-to-image Explained - Dataconomy](#)
- 73. [AI Glossary: Artificial Intelligence Terms - Dataconomy](#)
- 74. [Chunking for LLMS using Haystack Huggingface](#)
- 75. [Fine-Tune Your Own Llama 2 Model in a Colab Notebook | Towards Data Science](#)
- 76. [WizardLM/WizardLM-13B-V1.1 · Hugging Face](#)
- 77. [Imsys/vicuna-13b-v1.3 · Hugging Face](#)

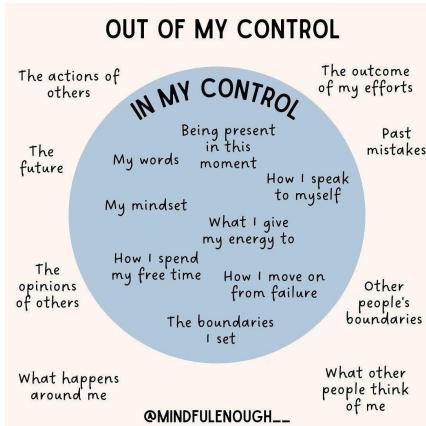
78. [LlamaIndex - Data Framework for LLM Applications](#)
79. [Complex Earth Moon Rotation](#)
80. [Learning to Identify Critical States for Reinforcement Learning from Videos](#)
81. [Tools for literature review project](#)
82. [ARM vs Intel: What Are They? What is the Difference Between Them?](#)
83. [A Gentle Introduction to Calculating the BLEU Score for Text in Python](#)
84. [Statistical source expansion for question answerin](#)
85. [Inc42 Media: The Game Of Multiples: Analysing Startup Valuations](#)
86. [Analytics India Magazine: Top 7 Generative AI Courses by AWS](#)
87. [Swarajya: How India Is Using AI To Build The Internet For Local Languages](#)
88. [CNBCTV18: How multiple Indian farmers are cultivating world's most expensive Miyazaki mangoes to earn millions](#)
89. [Deccan Herald: DH Toon: 'Kartavya' limited to campaigns](#)
90. [Analytics India Magazine: Andrew Ng & HuggingFace Introduce New Course on Generative AI](#)
91. [Top 10 Free Specialised Courses by Andrew Ng](#)
92. [Analytics Insight: Sam Altman's Worldcoin Surpasses 2 Million Sign-ups, InQubeta \(QUBE\) Set to Deliver Impressive 50x ROI to Its Investors](#)
93. [Best Transformer Based LLMs on HuggingFace \(Part 1\)](#)
94. [Web 3.0: The Future of Internet](#)
95. [Long Sequence Modeling with XGen: A 7B LLM Trained on 8K Input Sequence Length](#)
96. [Meet DORSal: A 3D Structured Diffusion Model for the Generation and Object-Level Editing of 3D Scenes](#)
97. [Step-by-Step BERT Implementation Guide - Analytics Vidhya](#)
98. [Beebom: 12 Best Large Language Models \(LLMs\) in 2023](#)
99. [Unity Announce the Release of Muse: A Text-to-Video Games Platform that lets you Create Textures, Sprites, and Animations with Natural Language](#)
100. [The Indian Express: ChatGPT is just the tip of the iceberg: 10 AI tools that are way cooler than OpenAI's chatbot](#)
101. [Free Neo4j Beginners Courses](#)
102. [Using Langchain and Open Source Vector DB Chroma for Semantic Search wit...](#)
103. [Machine Learning Pipelines for Kubeflow](#)
104. [Video series on Machine learning theory and how to apply the algorithms](#)
105. [Abacus.AI - Effortlessly Embed Cutting Edge AI In Your Applications.](#)
106. [Explore the ChatLLM feature by #abacusai.](#)
107. [ChatGPT can help you write a standout resume in seconds—here's how](#)
108. [Let's learn Data Science in 2023, All are FREE](#)
109. [FREE certification course from Harvard University:](#)
110. [List of Awesome AI Agents](#)
111. [ML for economics](#)
112. [Noteable Plugin: The ChatGPT Plugin That Automates Data Analysis - KDnuggets](#)
113. [From Google Colab to a Ploomber Pipeline: ML at Scale with GPUs - KDnuggets](#)
114. [ColabCode: Deploying Machine Learning Models From Google Colab - KDnuggets](#)
115. [Running Redis on Google Colab - KDnuggets](#)
116. [Abacus.AI - Effortlessly Embed Cutting Edge AI In Your Applications.](#)
117. [What's going on with the Open LLM Leaderboard?](#)
118. [Become a Full Stack Developer for FREE](#)
119. [Language is remarkably nuanced](#)
120. [Techniques to Cut the Costs of Using ChatGPT and GPT-4](#)
121. [How to Draw Graphs, Charts, and Diagrams in ChatGPT | Beebom](#)
122. [How to Write A Business Plan PDF - PDF Gate](#)
123. [Database Sharding Explained](#)
124. [Meet Gorilla: UC Berkeley and Microsoft's API-Augmented LLM Outperforms GPT-4, Chat-GPT and Claude.](#)
125. [How to Build ML Applications on the AWS Cloud with Kubernetes and oneAPI](#)
126. [TestNYC-Benchmark-GPT4](#)
127. [Neural Networks](#)

128. [GPT4-Multimodal](#)
129. [How to evaluate and compare results between GPT-3, GPT-4, ChatGPT?](#)
130. [LLama2](#)
131. [DocPrompting: Generating Code by Retrieving the Doc](#)
132. [Awesome Transformers](#)
133. [HuggingFace released PEFT](#)
134. [Meta-Learning Fast Weight Language Models](#)
135. [Cisco Networking Academy is offering this Certifications and Courses \(FREE\)](#)
136. ["How Cloud Computing is transforming from raw infra to app-centric services"](#)
137. [Data Science Deconstructed!](#)
138. [Top No-Code Machine Learning Platforms](#)
139. [Data Analyst Roadmap](#)
140. [Question Answer Generation](#)
141. ['Lessons Learned Applying NLP to Create a Web-Scale Knowledge Graph'](#)
142. [All the ways to Deploy a ML Model](#)
143. [Selection of Appropriate Statistical methods For Data Analysis](#)
144. [Contrastive learning](#)
145. [EDA Visuals](#)
146. [AI Decision Tree](#)
147. [Grab these top-rated AI/ Machine Learning books for FREE! 📚🔥](#)
148. [Trick for better LLM context retrieval for those who haven't seen it:](#)
149. [HypotheticalDocumentEmbedder](#)
150. [Transformer Model Chronology](#)
151. [Distributed Backpropagation Algorithm](#)
152. [Overview of MLOps in Dev, Stage, Production Env.](#)
153. [GPT Progression](#)
154. [Google GAN](#)
155. [Data Science, Machine Learning & Deep learning Topics](#)
156. [Takeaways From the Linkedin Algorithm Report](#)
157. [Machine learning debugging models](#)
158. [Generative AI Apps](#)
159. [Digital Twin](#)
160. [How to install, use ChatGPT](#)
161. [AI Explainability 360](#)
162. [Renewable Energy](#)
163. [Curve Fitting Methods](#)
164. [\(LLMs\) going to impact search engine providers](#)
165. [Information Retrieval Metrics](#)
166. [Data and Analytics Platform - 1](#)
167. [Data and Analytics Platform - 2](#)
168. [How to Modernize Enterprise Data](#)
169. [MLOPS Tools](#)
170. [Data Engineering](#)
171. [VALL-E, that generates audio from text #MSFTResearch!](#)
172. [GitHub : 7 quality learning resources](#)
173. [Handling Missing Data](#)
174. [Learn For Free](#)
175. [Top 10 Freelancing Websites](#)
176. [Unexplored Python Libraries for Time Series Analysis](#)
177. [12 Time Series Python Libraries incl Deep Learning](#)
178. [Sanskrit text extraction](#)
179. [Pdfplumber python library](#)
180. [OpenAI Introduces Point-E](#)
181. [MLIR Compiler Infrastructure](#)
182. [Vision MLOps Assistors](#)
183. [Understanding Bias-Variance Trade-off from a Bayesian Perspective](#)

184. [Deep Learning Paper Implementations](#)
185. [Can we solely rely on LLMs' memories](#)
186. [CMU Advanced NLP Lectures playlist](#)
187. [30m global terrain dataset](#)
188. [Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles](#)
189. [Microsoft is offering online courses for FREE](#)
190. [The Code Review Pyramid](#)
191. [30 Cybersecurity search engines for Researchers:](#)
192. [Intro Article on Web3.0](#)
193. [SOAP and REST APIs](#)
194. [PyRCA](#)
195. [CS 329S: Machine Learning Systems Design](#)
196. [Microsoft Machine Learning Studio \(classic\)](#)
197. [Top 5 Notable Humanoid Robots Made in India](#)
198. [TransUNet — Revolutionize Traditional Image Segmentation](#)
199. [Steamship GPT3 based Chatbot](#)
200. [Bootstrapping Labels with GPT-4](#)
201. [10 Most Used Machine Learning Algorithms In Python With Code](#)
202. [Customer Reviews NLP Spacy Analysis and ML/AI Demand Forecasting of the Steam PC Video Game Service – Digital High Science](#)
203. [Neo4j and Google Cloud Join Forces to Empower Generative AI with Knowledge Graphs - Multiplatform AI](#)
204. [Introducing DistMult and ComplEx for PyTorch Geometric](#)
205. [Integrate LLM workflows with Knowledge Graph using Neo4j and APOC](#)
206. [qiskit](#)
207. [Understanding Deep Learning](#)
208. [CMU Advanced NLP 2022](#)
209. [Synthetic Data Explainable AI](#)
210. [Roadmap to learn Web3](#)
211. [Aeroponic Potato Farming Technique And Its Advantages](#)
212. [How to Grow Potatoes in The Air With a Technique That Increases The Yield by 10x](#)
213. [GRC](#)
214. [Beyond ChatGPT: The Future Of AI At Work](#)
215. [Data generator](#)
216. [How To Create Animated Videos For YouTube Using 'Text to Video' AI Tools](#)
217. [Essence of linear algebra](#)
218. [Video lectures, Optimization for AI by Se-Young Yun](#)
219. [Video lectures, Statistical Learning by Se-Young Yun](#)
220. [Video lectures, Deep Learning by Se-Young Yun](#)
221. [Speech Synthesis, Recognition, and More With SpeechT5](#)
222. [A New Deep Reinforcement Learning \(DRL\) Framework can React to Attackers in a Simulated Environment and Block 95% of Cyberattacks Before They Escalate](#)
223. [fast.ai - Making neural nets uncool again](#)
224. [Awesome Quantum Machine Learning](#)
225. [InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval](#)
226. [Here's a roadmap to Master Python!](#)
227. [Read paper :Constitutional AI: Harmlessness from AI Feedback](#)
228. [Index of /lecture_slides](#)
229. [MLU-Explain](#)
230. [Learn Data Structure using these 46 Best Resources](#)
231. [Transformer models: an introduction and catalog](#)
232. [Leta transcripts – Dr Alan D. Thompson – Life Architect](#)
233. [How to produce factually accurate automatic text summaries - Amazon Science](#)
234. [Understanding Large Language Models -- A Transformative Reading List](#)
235. [Stanford CRFM Introduces PubMedGPT 2.7B](#)

- 236. [ChatGPT competitors: Amazon jumps into fray with generative AI better than GPT-3.5](#)
- 237. [Multimodal Chain-of-Thought Reasoning in Language Models](#)
- 238. [8 Types of Google Cloud Analytics: How to Choose?](#)
- 239. [RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning](#)
- 240. [Researchers at Stanford Introduce ControlNet: A Neural Network Structure to Control Pre-Trained Large Diffusion Models to Support Additional Input Conditions - MarkTechPost](#)
- 241. [nlpconnect/vit-gpt2-image-captioning · Hugging Face](#)
- 242. [Imx sentiment demo.ipynb - Colaboratory](#)
- 243. [Large Language Models and Where to Use Them: Part 1](#)
- 244. [A deep reinforcement learning model that allows AI agents to track odor plumes](#)
- 245. [Vertex AI Foundations for secure and compliant ML/AI deployment | Google Cloud Blog](#)
- 246. [Using PyGWalker to Enhance Your Jupyter Notebook EDA Experience](#)
- 247. [Meet The Sketchy Cousin of Stable Diffusion, 'Scribble Diffusion'](#)
- 248. [Algebra and Trigonometry 2e - OpenStax](#)
- 249. [Calculus Volume 1 - OpenStax](#)
- 250. [Introductory Business Statistics - OpenStax](#)
- 251. [Generative AI: What are the potential applications and ethical concerns?](#)
- 252. [Zero-shot image-to-text generation with BLIP-2](#)
- 253. [50+ ML Papers Explained](#)
- 254. [Awesome Community Detection Research Papers](#)
- 255. [Top 15 Use Cases of Object Detection in Computer Vision](#)
- 256. [Understanding Vision Transformers \(ViTs\): Hidden properties, insights, and robustness of their representations | AI Summer](#)
- 257. [Evaluating explainability for graph neural networks | Scientific Data](#)
- 258. [Explicit Planning Helps Language Models in Logical Reasoning](#)
- 259. [How AI Will Transform Project Management](#)
- 260. [CI/CD for Machine Learning \(GitOps\) - W&B Courses](#)
- 261. [Use Open AI \(ChatGPT\) On Your Own Large Data!](#)
- 262. [DTensor Concepts | TensorFlow Core](#)
- 263. [Download C3 AI Center of Excellence Data Sheet](#)
- 264. [Instruction Tuning with GPT-4](#)
- 265. [Unlocking the Potential of ChatGPT: A Comprehensive Exploration of its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing](#)
- 266. [Inner \(Dot\) product of two Vectors. Applications in Machine Learning](#)
- 267. [Task-driven Autonomous Agent Utilizing GPT-4, Pinecone, and LangChain for Diverse Applications](#)
- 268. [How to Use Large Language Models \(LLM\) in Your Own Domains](#)
- 269. [chatGPT Telegram](#)
- 270. [Open Source vs Proprietary LLM](#)
- 271. [GPT4Video: A Unified Multimodal Large Language Model](#)
- 272. [Explore Notus-7B released model.](#)
- 273. [How to Effectively use Docker!](#)
- 274. [Data Science, Machine Learning you can find multiple data distributions](#)
- 275. [12 Tools for Writers](#)
- 276. [Data Storytelling: A Visual Guide to Chart Selection](#)
- 277. [When to use What Data Science Tools](#)
- 278. [How Many Open-Source LLMs do you Know?](#)
- 279. [A/B Testing Methods](#)
- 280. [Data Distribution](#)
- 281. [Where can you find each Data Distribution?](#)
- 282. [API Security Checklist](#)
- 283. [Noisy Model Learning](#)
- 284. [Machine Learning Engineer](#)
- 285. [Circle of Controls](#)
- 286. [Llamaindex + pptx-api](#)
- 287. [Video lectures, Stanford CME 302 Numerical Linear Algebra autumn 2007](#)
- 288. [You need to be careful with Time Bias](#)

- 289. [Data Leakage is more dangerous than Overfitting.](#)
- 290. [First-ever course on Data-Centric AI](#)
- 291. [Anomaly detection with pytimetk](#)
- 292. [Boxplots vs Raincloud](#)
- 293. [Data visualization and exploration](#)
- 294. [Multi-Modal Retrieval](#)
- 295. [Top Important Computer Vision Papers for the week from 28-8-to-3-9](#)
- 296. [Do You Need to Know All Design Pattern?](#)
- 297. [Embedding In OpenAI API](#)
- 298. [API Architectural Styles](#)
- 299. [Databases](#)
- 300. [How Github Works?](#)
- 301. [10 AI Tools, 10 websites that will save you hundreds of hours](#)
- 302. [Discriminative vs Generative Models](#)
- 303. [Most Used Design Patterns Cheatsheet](#)
- 304. [Stay ahead of the curve with these 12 powerful GitHub repositories for learning data science, analytics, and engineering](#)
- 305. [Introduction MMMU, a Massive Multi-discipline Multimodal Understanding](#)
- 306. [Introduction to Important AI Tools/Courses](#)
- 307. [LLM+LLMops APIs](#)
- 308. [Generating or engineering features from Time Series data](#)
- 309. [How Many LLMs do you know?](#)
- 310. [The latest ML research:](#)
- 311. [Hashing Vs Encryption vs Encoding](#)
- 312. [Checklist for Data Scientists or Machine Learning or AI practitioners](#)
- 313. [Activation functions while working with neural nets.](#)
- 314. [Critical Considerations in Model Development and Evaluation](#)
- 315. [Open-Source LLMs vs. ChatGPT](#)
- 316. [What is Table Transformer \(TATR\)?](#)
- 317. [Open Source LLMs, Fine-Tunes and RAG Based Vector Store APIs - The Abacus.AI Blog](#)
- 318. [Trip down the GPU lane with Machine Learning | PPT](#)
- 319. [LLM Capabilities](#)
- 320. [The 40-hour LLM application roadmap: Learn to build your own LLM applications from scratch](#)
- 321. [What's next in large language model \(LLM\) research? Here's what's coming down the ML pike](#)
- 322. [Feature Engineering](#)
- 323. [Accelerating Generative AI with PyTorch II: GPT, Fast | PyTorch](#)
- 324. [Google Cloud Skills Boost](#)
- 325. [Table detection and recognition with Table Transformer \(TATR\).](#)
- 326. [Glossary of artificial intelligence - Wikipedia](#)
- 327. [Machine learning - Wikipedia](#)
- 328. [Data Collection Techniques](#)
- 329. [Document Embedding](#)



In my Control and Out of My Control

The Mathematical Modeling of Black Swan Events

https://medium.com/@gp_pulipaka/the-mathematical-modeling-of-black-swan-events-52dbd44814b9

Analyzing Transformers in Embedding Space Explained

<https://towardsdatascience.com/analyzing-transformers-in-embedding-space-explained-ef72130a6844>

Multimodality is the future! Here are 10 multimodal models that can be used to build futuristic #AI apps:

Open-source:

1. LLaVa - [Inkd.in/gN4eca2D](https://huggingface.co/LLaVA) : LLaVA is the most popular open-source multimodal framework.
2. OpenFlamingo: [Inkd.in/gnzc7wSi](https://huggingface.co/OpenFlamingo)
3. Idefics-80b: [Inkd.in/gr-qYVsH](https://huggingface.co/Idefics-80b)
4. SPHINX: [Inkd.in/gvZRSmVV](https://huggingface.co/SPHINX)
5. LLaMA-Adapter: [Inkd.in/guX635zA](https://huggingface.co/LLaMA-Adapter)
6. Fuyu-8B: [Inkd.in/grVbfhY8](https://huggingface.co/Fuyu-8B)
7. Mini-GPT4: [Inkd.in/gHXM6-KF](https://huggingface.co/Mini-GPT4)
8. InstructBLIP: [Inkd.in/gpuU_dRM](https://huggingface.co/instructblip)

Closed:

1. OpenAI GPT-4V: openai.com/gpt-4; has API access with heavy rate limit
2. Google BARD: bard.google.com, but with unofficial access [Inkd.in/gW6vR88m](https://huggingface.co/bard)

Are there any other multimodal models you like?

2000+ new AI tools were released in the last 30 days.

Here's 20 cutting-edge AI Tools for :

- Designer
- Writing
- Video
- Audio
- Coding
- Productivity
- Images

• Design Tools:

- 1) Auto Draw - autodraw.com
- 2) Flair AI - flairai.io
- 3) Pictory - pictory.ai

• Content Creation Tools:

- 4) Writesonic - writesonic.com
- 5) ChatGPT - chat.openai.com
- 6) Stable Diffusion - lnkd.in/gaRJqgbH

• Writing Tools:

- 7) Copy AI - copy.ai
- 8) Jasper AI - jasper.ai
- 9) Quillbot - quillbot.com

• Coding Tools:

- 10) DeepCode - deepcode.ai
- 11) 10Web - 10web.io
- 12) Akkio - akkio.com

1. Social Media Video :

A game-changingAI tool that creates social media videos.

A single prompt can be transformed into high-quality content.

Welcome to seamless, innovative video production with @lattesocialism

platform.latte.social/vid-gen

2. Coding Tabnine

The perfect AI coding assistant.

Enabling developers to:

- Generate code
- Talk to their code
- Translate code and more

3. Website Builder - 10Web_io

Create Wordpress websites using AI in a matter of minutes.

4. Excel on Steroids - RowsHQ

An incredible AI assisted Excel spreadsheet replacement, do almost anything with spreadsheets using AI.

5. API Integrations - leap_api

Add AI to whatever you're building quickly and easily. Image generators and other AI models available to integrate fast

6. Writing Assistant - c_omposeai

A free chrome extension that uses AI to help you write anything faster saving you tons of time.

7. UI/UX Design - uizard

Create engaging software designs with AI in a matter of minutes, not weeks.

8. Personal Assistant - heypi_ai

An intuitive AI personal assistant to help you maximize productivity

OpenSource version of OpenAI GPTs is here 

Less than 72 hours have passed since OpenAI released "GPTs" - ChatGPT augmented with custom tools and custom instructions that anyone can build with Nocode.

Opensource community already came up with OpenGPT built on top of LangChain, LangServe, and LangSmith to emulate similar experience and functionality as OpenAI GPTs.

What makes OpenGPT stand out?

1. Choice of LLMs: Users can select from over 60+ LLMs provided by LangChain, ensuring a tailored AI experience to suit varied project needs.
2. Prompt Engineering: With LangSmith, developers have the flexibility to fine-tune and debug prompts, leading to enhanced accuracy and performance of the LLMs.
3. Extensive Tool Integration: The platform boasts a library of over 100 tools from LangChain, alongside the flexibility to create and incorporate custom tools, offering unparalleled versatility.

Learn more :

<https://github.com/langchain-ai/opengpts>

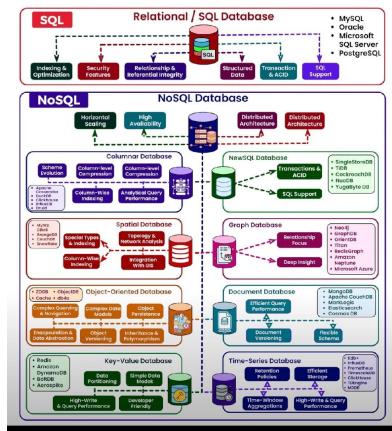
ChatGPT is a web app (you can access it in your browser) designed specifically for chatbot applications—and optimized for dialogue. It relies on GPT to produce text, like explaining code or writing poems. GPT, on the other hand, is a language model, not an app.

ChatGPT is primarily designed to generate human-like responses to text input, while OpenAI Playground is intended to experiment with different types of machine learning models. ChatGPT can be used for a variety of applications, including chatbots, language translation, and text summarization.

Generative AI, OpenAI, and ChatGPT: What are these?

<https://medium.com/data-science-at-microsoft/generative-ai-openai-and-chatgpt-what-are-they-3c80397062c>

Types of Databases



Types of Databases

rasbt/machine-learning-book: Code Repository for Machine Learning with PyTorch and Scikit-Learn

<https://github.com/rasbt/machine-learning-book>

MIT university is offering FREE education in Data Science.

Courses cover:

- Python
- Data Science
- Machine Learning
- Statistics
- Linear Algebra &
- Deep Learning

Learn from the best at a free of cost!

1. Introduction to Computer Science and Programming in Python

A great course to learn the fundamentals of computer science in Python

<https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>

2. Linear Algebra

Get an overview of the important and basic concepts of Linear Algebra required for Machine Learning

<https://ocw.mit.edu/courses/res-18-010-a-2020-vision-of-linear-algebra-spring-2020/>

3. Introduction To Data Science

Learn all the fundamental concepts of Data Science and Machine Learning.

<https://ocw.mit.edu/courses/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/>

4. Statistics

Learn the important stats concepts and their practical use cases.

<https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2014/>

5. MIT's Introduction to Deep Learning

This course will give you a complete idea of Deep Learning, Computer vision, and Natural Language Processing

<http://introtodeeplearning.com/>

Video lectures, UC Berkeley Math 54 Linear Algebra & Differential Equations summer 2020, summer 2021, by Luvreet Sangha, Peter Koroteev

<https://math.berkeley.edu/~pkoroteev/math54.html>

YT Playlist :

<https://youtube.com/playlist?list=PLShth7hrtLHO2U1Xkrl6ZgMyuPHDxRcob&si=0RP1WtddFDwe48bL>

Math H54 Honors Linear Algebra & Diff Eqns fall 2020, by Yu-Wei Fan

A Complete Guide to Creating and Storing Embeddings for PostgreSQL Data

<https://www.timescale.com/blog/a-complete-guide-to-creating-and-storing-embeddings-for-postgresql-data/>

A New Era of GPTs: 10 Game-Changing Creations

<https://www.analyticsinsight.net/a-new-era-of-gpts-10-game-changing-creations/>

Comparing the Performance of LLMs: A Deep Dive into Roberta, Llama 2, and Mistral for Disaster Tweets Analysis with Lora

<https://huggingface.co/blog/Lora-for-sequence-classification-with-Roberta-Llama-Mistral>

Multi-Modal RAG : Llamaindex Blog

<https://blog.llamaindex.ai/multi-modal-rag-621de7525fea>

CS50's Introduction to Artificial Intelligence with Python

What you'll learn

Graph search algorithms

Reinforcement learning

Machine learning

Artificial intelligence principles

How to design intelligent systems

How to use AI in Python programs

HarvardX: CS50's Introduction to Programming with Python

What you'll learn

Functions, Variables

Conditionals

Loops

Exceptions

Libraries

Unit Tests

File I/O

Regular Expressions

Object-Oriented Programming

Et Cetera

Artificial Intelligence in Business: Creating Value with Machine Learning

Fundamentals of TinyML

What you'll learn

Fundamentals of Machine Learning (ML)

Fundamentals of Deep Learning

How to gather data for ML

How to train and deploy ML models

Understanding embedded ML

Responsible AI Design

CS50's Introduction to Computer Science

broad and robust understanding of computer science and programming

How to think algorithmically and solve programming problems efficiently

Concepts like abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development

Familiarity in a number of languages, including C, Python, SQL, and JavaScript plus CSS and HTML

How to engage with a vibrant community of like-minded learners from all levels of experience

How to develop and present a final programming project to your peers

LLMs for Chip Design

Proposes using LLMs for industrial chip design by leveraging domain adaptation techniques.

Evaluates different applications for chip design such as assistant chatbot, electronic design automation, and bug summarization.

Overall, domain adaptation significantly improves performance over general-purpose models on a variety of design tasks. Using a domain-adapted LLM for RAG further improves answer quality.

More results:

The authors report achieving "a score of 7.4 out of 10 point scale for engineering assistant chatbot responses based on expert evaluations, achieve more than 50% correctness in EDA script generation, and expert evaluation rating of 4 to 5 out of 7 point scale for summarizations and assignment identification tasks."

Another interesting insight reported in the paper is that domain adaptation combined with a retriever mechanism closes the gap between a state-of-the-art LLaMA2 70B model and a much smaller 13B model. LLMs struggle with generating engineering/technical responses which often lead to hallucinations. RAG has the potential to mitigate these issues and ensure the usefulness of the proposed engineering assistant chatbot.

Furthermore, "fine-tuning the ChipNeMo retrieval model with domain-specific data improves the retriever hit rate by 30% over a pre-trained state-of-the-art retriever, in turn improving the overall quality of RAG responses."

This paper only highlights the tremendous opportunities in deeply exploring fine-tuned models and retriever-based approaches for difficult domain-specific tasks.

arxiv.org/abs/2311.00176

Data Analytics can be divided into 5 types based on the questions it can answer.

Descriptive analytics

What happened?

Descriptive analytics answers questions about what happened.

Descriptive analytics techniques summarize large datasets to present insights to stakeholders.

The presentation of data related to those KPIs is descriptive analytics.

Diagnostic analytics

Why did it happen?

Diagnostic analytics helps answer questions about why things happened and is the next step in data analytics after descriptive analytics.

Analysts take findings from descriptive analytics and dig deeper to find the cause.

Diagnostic analytics generally occurs in three steps:

Identify anomalies in the data.

Collect data that are related to these anomalies.

Use statistical techniques to discover relationships and trends that explain these anomalies.

Predictive analytics

What will happen in the future?

Predictive analytics helps answer questions about what will happen in the future.

Predictive analytics techniques include statistical and machine learning techniques such as forecasting, neural networks, decision trees, and regression.

Predictive analytics is often performed by data scientists.

Prescriptive analytics

What actions should be taken?

Prescriptive analytics takes predictive analytics one step further and helps answer questions about what actions should be taken to achieve a goal or target.

This technique allows businesses to make data-informed decisions in the face of uncertainty.

Algorithmic content recommendations are a common implementation of prescriptive analytics.

Cognitive analytics

How can the problem be solved best?

Cognitive analytics combines artificial intelligence, machine learning, and data analytics approaches to guide decision-making.

A self-learning feedback loop, this feedback loop enables cognitive applications to become more precise over time.

Learn the power of LangChain: A comprehensive guide to building custom Q&A chatbots

<https://datasciencedojo.com/blog/understanding-langchain/#>

LLM Collection | Prompt Engineering Guide

<https://www.promptingguide.ai/models/collection>

In Machine learning, We have 2 types of errors: Reducible and Irreducible.

Reducible error

1. Bias and variance are the main sources of reducible error.
2. A model with high bias pays little attention to the training data and oversimplifies the model.
3. A model with high variance pays too much attention to the data, leading to overfitting.
4. By balancing well the Bias-Variance duo, we can minimize the error.
5. But minimizing the error isn't always enough for a good result. Sometimes the minimum is still too high.

Irreducible error

1. Irreducible error refers to an error that cannot be reduced even if the model is well-trained.
2. The problem originates from the complexity and randomness of real-world data.
3. There's no model that could perfectly predict the world.
4. Random noise is the variability in data that cannot be captured by a model.
5. Thanks to complexity, there will always be unknown variables that we cannot measure or include in the model.

If the effect of these factors is huge, the final model will have a too high irreducible error.

You must know about this concept since it is a crucial part of every model!

Data characteristics

- Value
- Variety
- Variability
- Veracity
- Velocity
- Visualization
- Volume

Autoencoders are Neural Networks!

They are simple yet so powerful!

Today, We'll understand:

- What Autoencoders are ?
- How they are trained
- The applications of Autoencoders
- And how to implement them from scratch using PyTorch Lightning! ⚡

Let's go! 🚀

Autoencoders have two main parts:

1 Encoder: Compresses the input into a dense representation (latent space)

2 Decoder: Reconstructs the input from this dense representation.

The idea is to make the reconstructed output as close to the original input as possible: 👍

Applications of Autoencoders:

- Dimensionality Reduction: Like PCA but cooler. 😎
- Anomaly Detection: If reconstruction error is high, something's fishy!
- Data Denoising: Clean noisy data by training on noise.

A glimpse on how a denoising autoencoder is trained: 🎉

We'll use PyTorchLightning ⚡ for this.

- ◆ First, let's define our autoencoder!

A LightningModule enables your PyTorch nn.Module to play together in complex ways inside the t

Best YouTube Courses for Programming:

› SQL

https://youtube.com/playlist?list=PLD20298E653A970F8&si=OBUHxORqzd_FzNJ0

› Python

https://youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&si=T1k6QMcoTfMc_6Gm

› JavaScript

https://youtube.com/playlist?list=PLDyQo7g0_nsXISfu0BpG5Fgz0Qe3lVWhA&si=atoSLDgo4gkP_VXO

› Go

https://youtube.com/playlist?list=PL4cUxeGkcC9gC88BEo9czgyS72A3doDeM&si=mk5WssGEfxI_dtEM

› C

https://youtube.com/playlist?list=PL9IEJIKnBJjG5H0yIFAzpzs9qSmW_eICB&si=6mev2xdbMbkhFO2C

› C++

<https://youtube.com/playlist?list=PLlrATfBNZ98dudnM48yfGUIDqGD0S4FFb&si=rUCCPjcNZO4a2L-7>

› Java

https://youtube.com/playlist?list=PLsyeobzWxl7pe_IiTfNyr55kwJPWbgxB5&si=umNd5YI7WzS0tvzC

› C#

<https://youtube.com/playlist?list=PLdo4fOcmZ0oVxKLQCHpiUWun7vlJJvUiN&si=21YX9vVcFcVIVUzK>

› PHP

https://youtube.com/playlist?list=PLr3d3QYzkw2xabQRUpcZ_IBk9W50M9pe-&si=psW4pcAeM1ZD7CCY

› Swift

<https://youtube.com/playlist?list=PLMRghzcHGw1b89DXHOVA77ozWXWmuBkWX&si=UtxGDz0ii1qTe8gD>

› Kotlin

https://youtube.com/playlist?list=PLQkwcJG4YTCRSQikwhApYs9ij_Hc5Z9&si=oVyKecPWZ1JO_0d8

› Dart

https://youtube.com/playlist?list=PLptHs0ZDJKt_fLp8ImPQVc1obUJKDSQL7&si=sY2tyOmEhlykhmU

› TypeScript

<https://youtube.com/playlist?list=PLCbdBdyNHZXL3grXeUPZwTn1lo7pVaUdH&si=Imub50XnpKacPZIS>

› Ruby

<https://youtube.com/playlist?list=PLoQwKu-bfLSYRKLgzfhhiBlxDOigUFySP&si=cGuOCxtBmwsFs27z>

› Rust

<https://youtube.com/playlist?list=PLzMcBGfZo4-nyLTISRBvo0zjSnCnqjHYQ&si=mv7V13b5YI37ePqR>

› Scala

https://youtube.com/playlist?list=PLJGDHERh23x-YBJ8LmYU_IGBFflvsKfLu&si=lYorFAluFWblh4yt

› R

https://youtube.com/playlist?list=PL6gx4Cwl9DGCzVMGCPi1kwvABu7eWv08P&si=y0Es_nkQcsR6NSAb

› Julia

<https://youtube.com/playlist?list=PLP8iPy9hna6Qpx0MgGyEIJ5qFlaIXYf1R&si=mJZuPJHj6-TEBi1>

› MATLAB

https://youtube.com/playlist?list=PLYmlEoSHldN4bz5WY7e0OvXQ90E_xUOmz&si=cPtur79w1rMfaJOA

› Lua

<https://youtube.com/playlist?list=PLYBJzqz8zpWavt37pA6NANJTGSTlHpybU&si=E3buWt745ZLMla79>

› Pascal

<https://youtube.com/playlist?list=PLYBJzqz8zpWZUDwnt3TNi-8HkrR-B9RJe&si=z9CiViKd8gnmS5EB>

› Assembly

<https://youtube.com/playlist?list=PLmxT2pVYo5LB5EzTPZGfFN0c2GDiSXgQe&si=dTast1ox9C-H2oLq>

› Bash

https://youtube.com/playlist?list=PLT98CRI2KxKGj-VKtApD8-zCqSaN2mD4w&si=sdXuKm7OGKSG_GeV

Mathematics for Machine Learning

Companion webpage to the book “Mathematics for Machine Learning”. Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press.

<https://mml-book.com/>

How to Learn System Design?

If you're working as a software developer and you want to move towards a software architect role, or you're preparing for coding interviews for more senior roles, systems design is an important skill you need to have. Without knowing it, you will have a hard time designing new software systems and understanding existing ones.

System design refers to the process of defining a system's components. Architecture, modules, components, interfaces, and data are a few examples of these aspects. The process of defining, creating, and designing systems that suit the particular objectives and requirements of an organization is what you need to understand when it comes to system design.

To understand system design, you will need to know the following:

1. Performance vs Scalability
2. Latency vs Throughput and Availability vs Consistency (with patterns)

- 3. Availability patterns**
- 4. Background jobs**
- 5. Domain Name Systems**
- 6. CDNs**
- 7. Load Balancers**
- 8. Caching**
- 9. Asynchronism**
- 10. Communication patterns**
- 11. Performance antipatterns**
- 12. Monitoring**
- 13. Cloud design patterns**

To learn this, here are some **books** where you can start learning:

1. System Design Interview

One of the best books in this field, where you will learn how to design systems such as web crawlers or even YouTube.

Link: amzn.to/3lefBiD

2. Designing Data-Insensitive Applications

In this book, the author writes about different technologies used to store and process data. By reading it, you will gain insight into different algorithms used in the database world.

Link: amzn.to/3Y7I61c

3. Head First Design Patterns

A great book on OO design patterns, written in a simple style with examples in Java.

Link: amzn.to/3Ymwq9w

4. Cracking the Coding Interview

It is a general-purpose coding interview book where the author shares her insights into programming interviews at big tech companies like Microsoft and Google. It covers all basic topics like algorithms, data structures, SQL, etc.

Link: amzn.to/3E4TTEH

5. Fundamentals of Software Architecture

As System Design is related to Software Architecture, this book will give you an introduction to how to architect software systems. This book examines architecture patterns, components, soft skills, modernity, architecture as engineering, and many more.

Link: amzn.to/3JJLKco

Check the System Design Roadmap in the image (by roadmap .sh).

#softwareengineering

https://mina-ghashami.github.io/posts/2023-01-10-transformer/?utm_source=substack&utm_medium=email

Here's a list of favorite recent LLM papers to read:

1/ Zephyr LLM - a 7B parameter model with competitive performance to ChatGPT on AlpacaEval; applies distilled supervised fine-tuning to improve task accuracy and distilled direct performance optimization on AI feedback data to better align the model; shows performance comparable to 70B-parameter chat models aligned with human feedback.

arxiv.org/abs/2310.16944

2/ LLMs Meet New Knowledge - presents a benchmark to assess LLMs' abilities in knowledge understanding, differentiation, and association; benchmark results show

arxiv.org/abs/2310.14820

3/ Llemma - an LLM for mathematics which is based on continued pretraining from Code Llama on the Proof-Pile-2 dataset; the dataset involves scientific paper, web data containing mathematics, and mathematical code; Llemma outperforms open base models and the unreleased Minerva on the MATH benchmark; the model is released, including dataset and code to replicate experiments.

arxiv.org/abs/2310.10631

4/ LLMs for Software Engineering - a comprehensive survey of LLMs for software engineering, including open research and technical challenges.

arxiv.org/abs/2310.03533

5/ Self-RAG - presents a new retrieval-augmented framework that enhances an LM's quality and factuality through retrieval and self-reflection; trains an LM that adaptively retrieves passages on demand, and generates and reflects on the passages and its own generations using special reflection tokens; it significantly outperforms SoTA LLMs.

arxiv.org/abs/2310.11511

6/ Instruct-Retro - introduces Retro 48B, the largest LLM pretrained with retrieval; continues pretraining a 43B parameter GPT model on an additional 100B tokens by retrieving from 1.2T tokens.

arxiv.org/abs/2310.07713

7/ Overview of Factuality in LLMs - a survey of factuality in LLMs providing insights into how to evaluate factuality in LLMs and how to enhance it.

arxiv.org/abs/2310.07521

8/ LLMs Represent Space and Time - discovers that LLMs learn linear representations of space and time across multiple scales; the representations are robust to prompt variations and unified across different entity types; demonstrate that LLMs acquire fundamental structured knowledge such as space and time, claiming that language models learn beyond superficial statistics, but literal world models.

arxiv.org/abs/2310.02207

9/ StreamingLLM - a framework that enables efficient streaming LLMs with attention sinks, a phenomenon where the KV states of initial tokens will largely recover the performance of window attention; the emergence of the attention sink is due to strong attention scores towards the initial tokens; this approach enables LLMs trained with finite length attention windows to generalize to infinite sequence length without any additional fine-tuning.

arxiv.org/abs/2309.17453

10/ Retrieval meets Long Context LLMs - compares retrieval augmentation and long-context windows for downstream tasks to investigate if the methods can be combined to get the best of both worlds; an LLM with a 4K context window using simple RAG can achieve comparable performance to a fine-tuned LLM with 16K context; retrieval can significantly improve the performance of LLMs regardless of their extended context window sizes; a retrieval-augmented LLaMA2-70B with a 32K context window outperforms GPT-3.5-turbo-16k on seven long context tasks including question answering and query-based summarization.

arxiv.org/abs/2310.03025

You can find more interesting papers for this and past months here:

<https://github.com/dair-ai/ML-Papers-of-the-Week>

Language Translation using PyTorch Transformer

<https://debuggercafe.com/language-translation-using-pytorch-transformer/>

<https://towardsdatascience.com/graph-transformer-generalization-of-transformers-to-graphs-ead2448cff8b>

Levi (@levikul09)

<https://typefully.com/levikul09>

How to Fine-tune LayoutLMv3

Fine-tune LayoutLMv3 with Your Custom Data ...

<https://youtu.be/sZauGswJvas?si=6JvztURr7PuE-fyh>

Malaria Classification with Vision Transformer and PyTorch

<https://debuggercafe.com/malaria-classification-with-vision-transformer-and-pytorch/>

https://raw.githubusercontent.com/nltk/nltk_data

<https://raw.githubusercontent.com/>

Generative AI Models vs Discriminative AI Models

Discriminative AI models can be applied to classification or regression problem statements and produce predictions based on unobserved data using conditional probability. A Generative AI model, on the other hand, relies on the dataset's distribution to provide a probability for a given case.

While generative AI models comprehend how the data is embedded into the space, discriminative AI models classify the data space by understanding its boundaries. Due to their stark differences, both strategies are best suited for particular jobs.

Assuming a functional form for $P(Y)$ and $P(X|Y)$, generative classifiers produce estimated parameters from the data and apply the Bayes theorem to determine $P(Y|X)$, or posterior probability. Discriminative classifiers, on the other hand, make the assumption that $P(Y|X)$ has a functional form and directly estimate the parameters from the given data.

Discriminative AI model

Most conditional AI models, also known as discriminative AI models, are employed in supervised machine learning. They carry out what they 'literally' claim, classifying the data points into distinct groups and figuring out the boundaries using probability estimates and maximum likelihood.

On these AI models, outliers have little to no impact. Although they are a better option than generative AI models, misclassification issues can be a significant disadvantage.

Here are some examples and a brief description of the widely used discriminative AI models: Logistic Regression, Support Vector machines, Decision Trees.

A few other examples are commonly-used neural nets, k-nearest neighbor (KNN), conditional random field (CRF), random forest, etc.

Generative AI model

As the name suggests, generative models can be used to generate new data points. These models are usually used in unsupervised machine learning problems.

Generative AI models go in-depth to model the actual data distribution and learn the different data points, rather than model just the decision boundary between classes.

These AI models are prone to outliers, which is their only drawback when compared to discriminative AI models. The mathematics behind generative AI models is quite intuitive too. The method is not direct like in the case of discriminative AI models. To calculate $P(Y|X)$, they first estimate the prior probability $P(Y)$ and the likelihood probability $P(X|Y)$ from the data provided.

Putting the values into Bayes' theorem's equation, we get an accurate value for $P(Y|X)$.

Here are some examples as well as a description of generative AI models: Bayesian network, Autoregressive model, GAN.

Some other examples include Naive Bayes, Markov random field, hidden Markov model (HMM), latent Dirichlet allocation (LDA), etc.

Discriminative AI vs Generative AI: Which is the best fit for Deep Learning?

Artificial neural networks (ANNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are examples of supervised machine learning algorithms that have been used extensively in deep learning. The oldest of the three, ANN uses artificial neurons, weights, biases, and backpropagation to find patterns in the inputs. The majority of CNN's applications are in computer vision and image recognition. It functions by combining significant elements from an input picture. The most recent of the three, RNN, is employed in sophisticated domains such as time series analysis, handwriting recognition, natural language processing, etc.

These are the fields where discriminative AI models are effective and better used for deep learning as they work well for supervised tasks.

In addition to this, image clustering based on similarities can be achieved using deep learning and neural nets. Popular unsupervised deep learning techniques include autoencoder, Boltzmann machine, and self-organizing maps. For tasks like image denoising, compression, anomaly detection, exploratory data analysis (EDA) of high dimensional datasets, and even image generation, they use Generative AI models.

Infographic: 50 Cognitive Biases in the Modern World

<https://www.visualcapitalist.com/50-cognitive-biases-in-the-modern-world/>

MIT Researchers Introduce PFGM++

A Groundbreaking Fusion of Physics and AI for Advanced Pattern Generation - MarkTechPost

<https://www.marktechpost.com/2023/10/05/mit-researchers-introduce-pfgm-a-groundbreaking-fusion-of-physics-and-ai-for-advanced-pattern-generation/>

I Scanned 1000+ Prompts so You Don't Have to: 10 Need to Know Techniques

<https://medium.com/mlearning-ai/i-scanned-1000-prompts-so-you-dont-have-to-10-need-to-know-techniques-a77bcd074d97>

The ChatGPT list of Lists: A collection of 3000+ Prompts, examples, usecases

Read this story from Maximilian Vogel on Medium:

<https://medium.com/mlearning-ai/the-chatgpt-list-of-lists-a-collection-of-1500-useful-mind-blowing-and-strange-use-cases-8b14c35eb>

Why We Should Be Careful Using One-Tailed Hypothesis Tests in Research

1/ Introduction: Hypothesis testing is a cornerstone of scientific research. But did you know there's a debate about using one-tailed vs. two-tailed tests? Let's dive into why one-tailed tests might not be the best choice.



2/ What is a One-Tailed Test?: A one-tailed hypothesis test checks if a parameter is greater than (or less than) a specific value. It's looking in ONE direction. E.g., "Is drug A more effective than drug B?" 

3/ Limitation #1 - Biased Expectations: One-tailed tests assume we know the direction of the effect beforehand. But research should be exploratory! By choosing one direction, we might be biased by our expectations or hopes. 

4/ Limitation #2 - Missing Out: What if the effect is in the opposite direction? With a one-tailed test, we might miss out on significant findings just because they're not in the direction we expected. 

5/ Limitation #3 - False Positives: One-tailed tests can inflate the risk of false positives. If we're only looking in one direction, we might find "significant" results that aren't truly there. This can mislead and misinform. 

6/ Two-Sided Tests - A Broader View: Two-sided tests examine both directions. They're more general, asking, "Is there a difference?" rather than specifying a direction. They're often preferred for exploratory research. 

7/ Ethical Considerations: Science thrives on objectivity. By choosing a one-tailed test based on hopes or hunches, we risk compromising the integrity of our research. It's essential to approach tests without a predetermined direction. 

8/ But... Are One-Tailed Tests Always Bad?: No! There are situations where a one-tailed test is appropriate. E.g., if previous research overwhelmingly supports a particular direction. But these cases are exceptions, not the rule. 

9/ Conclusion: While one-tailed tests have their place, they come with limitations. For robust, unbiased research, two-tailed tests are often the better choice. Always consider the context and aim for objectivity. 

10/ Final Thoughts: Research is a journey of discovery. Let's ensure our methods are as open-minded as our quest for knowledge. Avoid biases, stay curious, and let the data speak for itself! 

#DataScience #Statistics

Top Large LLM Interview Questions & Answers

Read this story from Youssef Hosni on Medium:

<https://levelup.gitconnected.com/top-large-language-models-llms-interview-questions-answers-d7b83f94c4e>

Tough SQL questions

Below are the few conceptual questions from SQL.

If you are able to answer more than 80% questions, it means you are good at SQL else you need to work on it.

1. Why Rank skips the sequence?
2. Why Order by fails in Minus Query?
3. Why Right Join when Left Join can suffice the requirement?
4. Can we use aggregate functions without Group by clause?
5. Can we use Group by clause without aggregate functions?
6. In which scenario IN operator fails?
7. What's the limitation of MINUS operator?
8. How to create an infinite loop using recursive cte?
9. Why WHERE fails to filter aggregate functions?
10. Name an example of Cross Join.
11. In which scenario NOT IN operator fails.
12. Name a scenario where sequence of columns matter in GROUP BY clause.
13. Name a reserved keyword which has two different functionality in SQL.
14. If two NULLS are not same then why distinct column shows only 1 null in the output?
15. How to find existence of NULL in a column without using IS NULL keyword.
16. Name two different keywords which performs the same task.
17. What's the difference between GROUP BY and PARTITION BY?
18. What is the difference between DELETE,DROP and TRUNCATE?
19. Why CTEs are better than SUB QUERIES as far as performance is concerned?
20. Why CORRELATED QUERY badly impacts the performance?
21. Why TRUNCATE is a DDL command although it does not change the structure of table?
22. How you can prove that TRUNCATE resets the HIGH WATER MARK(HWM)?
23. Under what condition COALESCE will behave as NVL?
24. Why SQL is known as Structured Query Language?
25. Why the symbol '<>' corresponds to not equal?
26. What is the difference between IN and EXIST operator?
27. In which scenario AVG gives the incorrect result?
28. What is the difference between INTERSECT and INNER JOIN?
29. What is the difference between UNION ALL and FULL OUTER JOIN?

30. What is the difference between PRIMARY KEY and UNIQUE KEY?
31. What is the difference between INDEPENDENT AND CORRELATED QUERY?
32. Why window functions can't be used in WHERE CLAUSE?
33. Why column alias can't be used in WHERE CLAUSE but can be used in ORDER BY CLAUSE?
34. What are the SQL keywords which have inbuilt distinct functionality apart from DISTINCT?
35. Why WHERE CLAUSE nullify the impact of LEFT/RIGHT outer join in the output?

Graph Convolutional Networks: Introduction to GNNs

"Graph Neural Networks (GNNs) represent one of the most captivating and rapidly evolving architectures within the deep learning landscape. As deep learning models designed to process data structured as graphs, GNNs bring remarkable versatility and powerful learning capabilities.

Among the various types of GNNs, the Graph Convolutional Networks (GCNs) have emerged as the most prevalent and broadly applied model. GCNs are innovative due to their ability to leverage both the features of a node and its locality to make predictions, providing an effective way to handle graph-structured data.

In this article, Read into the mechanics of the GCN layer and explain its inner workings. Furthermore, author will explore its practical application for node classification tasks, using PyTorch Geometric as our tool of choice.

PyTorch Geometric is a specialized extension of PyTorch that has been created specifically for the development and implementation of GNNs. It is an advanced, yet user-friendly library that provides a comprehensive suite of tools to facilitate graph-based machine learning. To commence journey, the PyTorch Geometric installation will be required. If you are using Google Colab, PyTorch should already be in place, so all need to do is execute a few additional commands."

[<https://towardsdatascience.com/graph-convolutional-networks-introduction-to-gnns-24b3f60d6c95>] (<https://towardsdatascience.com/graph-convolutional-networks-introduction-to-gnns-24b3f60d6c95>)

Relationships Between t and F, and z and Chi-square

1/ The Statistical Family:

In the world of statistics, many tests are interconnected! Today, let's explore the relationships between t and F, and z and chi-square tests.

2/ t & F Tests:

The t-test examines differences between two group means, while the F-test (used in ANOVA) compares variances across multiple groups. But did you know one can be framed as the other?

3/ t-squared:

A cool fact is that if you square the t-statistic from a two-sample t-test, you get the F-statistic! Essentially, $t^2 = F$. This shows the direct relationship between the two tests.

Note: It's worth noting that this relationship is specific to a two-sample t-test scenario. F-tests in general (as used in multi-way ANOVAs or regression) do not have this direct t-squared relationship.

4/ Why does it matter?

When comparing only two groups in ANOVA (which usually handles 3+ groups), the F-test is equivalent to the square of the t-test. This provides flexibility in selecting tests and interpreting results!

5/ Switching Gears: z & chi-square

Now, onto the z-test and chi-square test! The z-test deals with population means, while the chi-square is all

about observed vs expected frequencies.

6/ z-squared:

Another interesting fact? If you square the z-statistic from a one-sample z-test, you get the chi-square value! So, z^2 = chi-square. An inherent link again!

7/ Real-World Application:

Consider testing if a die is biased. Using a z-test, we see if observed frequencies match expectations.

Square the z-value, and voila! We have a chi-square test result, commonly used for this purpose.

Note: The z-test is generally not used to test if a die is biased (that's more commonly a chi-square goodness-of-fit test). The connection can be seen when considering a test of a single proportion against a known value, where the squared z-statistic equals the chi-square.

8/ Puzzling Observation:

Ever wondered why these relationships exist? Both pairs (t & F, z & chi-square) compare observed values to expected ones under the null hypothesis. The squaring represents a sum of squared deviations.

9/ Insightful Takeaway:

Understanding these relationships deepens our grasp of statistical tests. It reveals the interconnected framework underpinning many hypothesis tests.

10/ Closing Thoughts:

Statistics might seem overwhelming with its myriad tests. However, recognizing these intertwined relationships can simplify our perspective and enhance our analytical prowess.

#Statistics #DataScience

You can now fine-tune an LLM without writing a single line of code!

A breakthrough in the open-source LLM space that can increase the speed of AI development and adoption by an order of magnitude.

I'll show you a demo you can follow, but first, let me start from the beginning:

A Large Language Model comes out of the factory knowing many things but with little expertise.

Take GPT-4, for example. It can speak without pause about mathematics but struggles to solve most problems. It knows about photography but not about my photos. It knows about business but can't say a thing about yours.

Fortunately, we can teach these models specialized knowledge. For example, we can force a model to always answer in a specific way or show it facts about a domain it didn't know before.

We call this process "fine-tuning."

For many use cases, fine-tuning a model is the difference between getting mediocre answers or feeling the tool is pure magic.

While OpenAI's models attract much attention, many companies use open-source models like Llama and Falcon. This gives them more control over their data, expenses, and how the model responds.

Unfortunately, fine-tuning a model is neither a simple process nor cheap. It takes a lot of time and GPU computing. It's also hard to find experienced people who know how to work with these models.

While you can choose your adventure and do everything yourself, the #monsterapis team released the ability to fine-tune a Large Language Model without writing any code. You can use it with any of the following models:

- Llama and Llama 2
- Falcon 7B, 40B
- OPT 125m, 6B
- GPT J 6B
- Stable LM 3B, 7B
- GPT 2 XL

Besides the obvious advantages of not dealing with code, complexity, or hardware, using their no-code tool will also let you fine-tune a model at a fraction of the cost! Their secret is using a decentralized GPU platform, which makes the process much more cost-efficient.

There's a simple step-by-step demo you can follow that will show you how simple the process is. Explore :

Introducing no-code LLM FineTuning with Monster API

<https://blog.monsterapi.ai/no-code-fine-tuning-llm/>

1/ 📈 Unpacking the 95% Confidence Interval (CI): Why it doesn't mean there's a 95% chance of containing the mean 😐

2/ ⚙️ Fixed vs. Variable: The true population parameter (like the mean) is a fixed, unknown value. The CI, on the other hand, can vary from one sample to another.

3/ ✨ Before vs. After: Before taking a sample & calculating a CI, we can say there's a 95% chance the next interval we calculate will contain the mean. But once it's calculated, the interval either contains the true mean or it doesn't.

4/ 🔪 Repetition Concept: The 95% confidence level means that if we were to take 100 different samples and compute a 95% CI for each one, we would expect about 95 of those intervals to contain the true mean.

5/ 🤷 Common Misconception: It's tempting to think of the CI as a probability interval after it's been calculated. But remember, probability pertains to the process (before the fact), not the specific interval outcome (after the fact).

6/ 💼 Visual Analogy: Imagine shooting arrows at a target. The bullseye is the true mean. If your bow is "95% confident", 95 out of 100 arrows will hit somewhere inside the bullseye. But for any single shot, it either hits or misses, no in-between.

7/💡 Why It Matters: Proper understanding of CI ensures that we interpret data correctly. Misunderstanding can lead to overconfidence in our results, potentially leading to incorrect decisions.

8/ 📚 In Conclusion: Confidence Intervals offer a way to capture the uncertainty in estimates, but interpreting them requires a clear understanding of the underlying concepts. Always remember: it's about potential outcomes in repeated sampling!

#Statistics #DataScience #ConfidenceInterval

BERT applications you should know about:

- BERTSUM
- Google Smart Search
- SciBERT
- BioBERT
- ClinicalBERT

Links to the articles 

BERTSUM

An innovative document-level encoder offers a fresh perspective by leveraging BERT to capture the semantic meaning of documents and generate sentence representations.

<https://t.co/giGoCwTO4H>

Google Smart Search

Google applied BERT models to both ranking and featured snippets in Search to do a much better job helping you find useful information.

<https://blog.google/products/search/search-language-understanding-bert/>

SciBERT

Maximizes performance on various scientific NLP tasks by capitalizing on unsupervised pretraining with a vast collection of scientific publications from diverse domains.

<https://t.co/WvWoHrjvIX>

BioBERT

A domain-specific language model trained on biomedical corpora. Outperforms BERT and previous models in various biomedical text mining tasks when pre-trained on biomedical corpora.

<https://t.co/p2e4RgOI2B>

ClinicalBERT

Discovers relationships between medical concepts, as judged by physicians. It outperforms baselines in predicting hospital readmission using discharge summaries and ICU notes.

<https://t.co/0XZj1L45a4>

Revolutionizing Language Models Unleashing the power of retrieval augmented generation rag

<https://ainomictech.medium.com/revolutionizing-language-models-unleashing-the-power-of-retrieval-augmented-generation-rag-bac5181fa162>

Code Llama | NVIDIA NGC

<https://catalog.ngc.nvidia.com/orgs/nvidia/teams/playground/models/codellama>

Agriculture usecases

How Machine Learning, AI Tools Helps in Agriculture?

For a variety of reasons, machine learning is a rapidly expanding technology with numerous applications in agriculture. The most intriguing ML use cases and how they help organizations will be covered in this article.

Due to climatic variations, insect and disease outbreaks, financial difficulties, and constantly shifting consumer preferences, the agriculture industry is subject to a variety of risks and uncertainties. As a result, the industry experiences considerable production losses, resource waste, and slow growth. It is here that ML is used. The primary issues in agriculture that ML technology address are:

Crop Diseases;
Drastic Weather changes , Earthquakes
Soil degradation;
Water scarcity;
Low yields;
Pest and disease outbreaks;
Weather changes.

Through automation and analysis of real-time data, ML can assist in resolving these problems as well as others. AI-harvesting bots, sensors, autonomous vehicles, GPS navigation and control systems, and other IoT gadgets can all be used in conjunction with machine learning. With the help of IoT-based apps, farmers can gather real-time data, see trends, and make educated decisions about weather changes, crops, and pests.

Precision agriculture (PA) is encouraged by machine learning (ML), which delivers useful insights and enhances decision-making processes. This phrase describes a crop management strategy that maximizes crop production by utilizing cutting-edge sensors and analysis tools. Therefore, ML gives farmers access to more precise data so they may automate agricultural processes, enhance crop yields, reduce production costs, and make better decisions.

Advantages of Machine Learning in Agriculture

Machine learning technology brings the following benefits:

- 1) Risk management: Farmers can apply machine learning to predict which weather- or environment-related variables would negatively affect their operations and take action to lessen their effects and reduce waste;
- 2) Data-driven decision-making: To make wise decisions about how to handle plants or animals and manage crops, ML models can analyze enormous volumes of historical and current data;
- 3) Resource management optimization: By quickly analyzing a variety of data and optimizing agricultural processes, ML makes precise and efficient resource management possible, resulting in lower costs and shorter turnaround times;
- 4) Safety enhancement: By automating numerous tasks like handling pesticides and large gear, ML systems enable farmers to maintain a better and safer working environment;
- 5) Crop resilience enhancement: In order to reduce crop losses, ML can provide early warnings and solutions

by evaluating data on weather patterns, disease outbreaks, or insects;

6)Market and price prediction: Machine learning models can predict crop prices and market demand based on historical trends. As a result, farmers can maximize their profit from crop selection, volume production, and marketing strategies.

The above benefits result in more productive, profitable, and sustainable farming operations. To provide businesses with these capabilities, ML relies on a variety of technologies.

Machine Learning use cases in agriculture

Agriculture applications based on ML can be divided into four main groups:

Crop management(Weed Detection, Diseases Detection, Yields Prediction, Selective Breeding, etc) ;

Soil management(Use of fertilisers,Soil monitoring, Insect Detection, etc) ;

Water management(Irrigation, Detecting leaks, Weather monitoring,etc) ;

Livestock management(Detecting animal diseases, Tracking feeding, Grazing control, etc) ;

The main Machine learning technologies in agriculture:

There are a lot of technologies that make up Machine learning, but the most beneficial for agriculture are: Time-series Forecasting for Weather, Earthquakes, Forecasting Crop Diseases, Drug Discovery using Deep Learning, Predictive Analytics, Computer Vision, GPT, Open AI.

dhblog - Why and How to Fine-tune CLIP

https://dienhoa.github.io/dhblog/posts/fine_tune_clip.html

ICML 2023 Tutorials

<https://icml.cc/virtual/2023/events/tutorial>

Similarity Algorithms :: TigerGraph Graph Data Science Library

<https://docs.tigergraph.com/graph-ml/current/similarity-algorithms/>

<https://medium.com/serpdotai/lora-low-rank-adaptation-of-large-language-models-82869419b7b8>

<https://www.businessstoday.in/magazine/deep-dive/story/from-byjus-to-oyo-why-retaining-investor-confidence-is-the-new-battleground-392054-2023-07-31>

prime example is edtech major BYJU'S, which was once hailed for achieving a valuation of \$22 billion in 2022. But today, it is getting panned for everything from not complying with local laws, not releasing its financial results, and engaging in aggressive sales tactics, among other things. Not only that, many of its marquee investors have marked down its valuation by more than half from \$22 billion on their books; do note that these are not a result of any down round.

Other names that have joined the mark-down party include hotel-room booking platform OYO, ride-hailing service Ola Cabs, food delivery app Swiggy, medtech start-up PharmEasy and fintech services provider Pine Labs, among others. All these firms were once widely celebrated for becoming unicorns (start-ups with a valuation of \$1 billion or more) and decacorns (valued above \$10 billion), not too long ago.

TQDM Usecases

<https://medium.com/@harshit4084/track-your-loop-using-tqdm-7-ways-progress-bars-in-python-make-things-easier-fcbbb9233f24>

When Companies were founded

- 🇫🇮 Nokia: 1865
- 🇯🇵 Toshiba: 1875
- 🇯🇵 Nintendo: 1889
- 🇺🇸 IBM: 1911
- 🇯🇵 Panasonic: 1918
- 🇺🇸 Disney: 1923
- 🇺🇸 Motorola: 1928
- 🇰🇷 Samsung: 1938
- 🇺🇸 HP: 1939
- 🇯🇵 Sony: 1946
- 🇺🇸 Microsoft: 1975
- 🇺🇸 Apple: 1976
- 🇺🇸 Dell: 1984
- 🇨🇳 Huawei: 1987
- 🇺🇸 Amazon: 1994
- 🇺🇸 Netflix: 1997
- 🇺🇸 Google: 1998
- 🇺🇸 Salesforce: 1999
- 🇺🇸 SpaceX: 2002
- 🇺🇸 Tesla: 2003
- 🇺🇸 Facebook: 2003
- 🇺🇸 YouTube: 2005
- 🇺🇸 Reddit: 2005
- 🇺🇸 Twitter: 2006
- 🇺🇸 Airbnb: 2008
- 🇺🇸 WhatsApp: 2009
- 🇺🇸 Instagram: 2010
- 🇺🇸 Zoom: 2011
- 🇺🇸 Twitch: 2011
- 🇺🇸 OpenAI: 2015
- 🇨🇳 TikTok: 2016

Enterprises are deploying LLMs in these four ways.

1. Prompt Engineering with Context
2. Retrieval Augmented Generation (RAG)
3. Fine Tuned Model
4. Trained Model

<https://www.fiddler.ai/blog/four-ways-that-enterprises-deploy-langs>

Machine Learning – Google AI

https://ai.google/build/machine-learning?utm_source=twitter&utm_medium=paid_social&utm_campaign=ml_hub_offnet_23_in&utm_content=fd-dev-discover&twclid=2-4rin3lpfb0a7z99h40b87y3bw

Machine Learning - Toolkit LMM – Google AI

<https://ai.google/build/machine-learning/toolkits/lm-android>

Exploring the promise of generative models in chemistry an introduction to diffusion-models

<https://medium.com/@cch57/exploring-the-promise-of-generative-models-in-chemistry-an-introduction-to-diffusion-models-31530e9d1dcb>

Graph of Thoughts: Solving Elaborate Problems with Large Language Models

<https://arxiv.org/abs/2308.09687>

The Definitive Guide to Einstein GPT | Salesforce Ben

<https://www.salesforceben.com/the-definitive-guide-to-einstein-gpt/>

LLM: A CLI utility and Python library for interacting with Large Language Models

<https://llm.datasette.io/en/stable/>

Tensor Algebra and its Applications to Data Science and Statistics

<https://arxiv.org/abs/2210.16182>

What is Retrieval-Augmented Generation?

Retrieval-augmented generation (RAG) is an AI framework for improving the quality of LLM-generated responses by grounding the model on external sources of knowledge to supplement the LLM's internal representation of information.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

<https://arxiv.org/abs/2005.11401>

Retrieval-Augmented Multimodal Language Modeling

<https://arxiv.org/abs/2211.12561>

Models like ChatGPT and others have built-in restrictions. One of the biggest problems is that they are unable to incorporate new information after the deadline for their last training, which leaves them in the dark about anything that has happened or developed since then. Additionally, even if they have a wealth of broad knowledge, they lack access to private or confidential company data, which is frequently essential for firms seeking specialized insights or decision-making.

This is where Retrieval Augmented Generation (RAG) steps in as a game-changer. RAG bridges the knowledge gap by dynamically retrieving relevant information from external sources, ensuring that the

generated responses are not only factual but also up-to-date.

Vector databases play an integral role in the RAG mechanism by enabling efficient and semantic retrieval of information. These databases store information as vectors, allowing RAG to quickly and accurately fetch the most relevant documents or data points based on the semantic similarity of the input query, enhancing the precision and relevance of the LLM's generated responses.

Implementing Question Answering Pipeline with LangChain

On top of the elements of the LLM ecosystem, LangChain provides a uniform interface and abstraction layer, streamlining the creation of generative AI applications. Developers can reduce boilerplate code and concentrate on providing value by using LangChain.

Your vector database integration with LangChain, now the AI application developers can easily leverage the superior performance provided by your application while building the knowledge retrieval component in the AI applications.

Your vector database integration with LangChain signifies a leap forward in the domain of question-answering systems. By leveraging the strengths of both LLMs and vector databases, this integration promises richer, more accurate, and context-aware answers. As AI continues to reshape our world, tools like LangChain, coupled with powerful vector database, will be at the forefront of this transformation.

#AI #MachineLearning #DeepLearning #DataScience #RAG #RetrievalAugmentedLearning
#VectorDatabase

"GraphGPT converts unstructured natural language into a knowledge graph."

GitHub: <https://github.com/varunshenoy/GraphGPT>
#ArtificialIntelligence #ChatGPT #DeepLearning

Deep Bidirectional Language-Knowledge Graph Pretraining

abs: arxiv.org/abs/2210.09338
github: <https://github.com/michiyasunaga/dragon>

Code Llama: Open Foundation Models for Code

<https://ai.meta.com/research/publications/code-llama-open-foundation-models-for-code/>

Github : facebookresearch
/codellama
Inference code for CodeLlama models

<https://github.com/facebookresearch/codellama>

Introducing Code Llama, a state-of-the-art large language model for coding

<https://ai.meta.com/blog/code-llama-large-language-model-coding>

Salesforce+ Salesforce AI Day: Calling all Trailblazers

https://www.salesforce.com/plus/experience/salesforce_ai_day/series/salesforce_ai_day

Tutorials and Unpublished Papers

<https://faculty.csbsju.edu/frioux/workinprogress.html>

"The Role of #QuantumComputing in #ArtificialIntelligence"

=> #QC can improve #NLP accuracy and speed

=> Maintain performance with large amounts of data.

Advanced LLM Machine Learning quantum computing, SuperAI , we can achieve better QML Models

New research reveals that machine learning on quantum computers can be accomplished with simpler and smaller datasets than previously thought

#AI #artificialintelligence #Electronics #intermediatescalequantumcomputers #llm #machinelearning

<https://multiplatform.ai/new-research-reveals-that-machine-learning-on-quantum-computers-can-be-accomplished-with-simpler-and-smaller-datasets-than-previously-thought/>

Researchers combine machine learning and quantum-classical computational molecular design to accelerate the discovery of efficient OLED emitters

#AI #artificialintelligence #computationalchallenges #deuterated #llm #machinelearning #OLEDemitters

<https://multiplatform.ai/researchers-combine-machine-learning-and-quantum-classical-computational-molecular-design-to-accelerate-the-discovery-of-efficient-oled-emitters/>

Salesforce Introduces XGen-7B, a new 7B LLM trained on 8K seq. length for 1.5T tokens.

Better or comparable results with MPT, Falcon, LLaMA, OpenLLaMA in text & code tasks.

Blog:<https://blog.salesforceairresearch.com/xgen/>

github: <https://github.com/salesforce/xgen>

On standard NLP benchmarks, XGen achieves comparable or better results when compared with state-of-the-art open-source LLMs (e.g. MPT, Falcon, LLaMA, Redpajama, OpenLLaMA) of similar model size.

Salesforce targeted evaluation on long sequence modeling benchmarks show benefits of 8K-seq models over 2K- and 4K-seq models.

XGen-7B achieves equally strong results both in text (e.g., MMLU, QA) and code (HumanEval) tasks. Training cost of \$150K on 1T tokens under Google Cloud pricing for TPU-v4.

Salesforce's CodeT5 system can understand and generate code

https://venturebeat.com/ai/salesforces-codet5-system-can-understand-and-generate-code/?utm_campaign=60ef0282c81ed40001b931ee&utm_content=613fd23853042200014dd831&utm_medium=socialshare&utm_source=twitter

Home of CodeT5: Open Code LLMs for Code Understanding and Generation

Blog :<https://blog.salesforceairresearch.com/codet5/>

Github :<https://github.com/salesforce/CodeT5>

Paper :<https://arxiv.org/abs/2305.07922>

Hugging Face blog :

<https://huggingface.co/Salesforce/codet5-small>

Tryage: Real-time, intelligent Routing of User Prompts to Large Language Model

<https://arxiv.org/abs/2308.11601v1>

LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models through Parameter-Efficient Fine-Tuning (Practical Experience Report)

<https://arxiv.org/abs/2308.11148v1>

MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models

<https://arxiv.org/abs/2308.09729v2>

Dive into LoRA Adapters

<https://towardsdatascience.com/dive-into-lora-adapters-38f4da488ede>

AutoGen: Enabling Next-Gen GPT-X Applications | FLAML

<https://microsoft.github.io//FLAML/docs/Use-Cases/Autogen/>

Aman's AI Journal • Reinforcement Learning from Human Feedback (RLHF)

<https://aman.ai/primers/ai/RLHF/>

The Role of AI in Biomarker-Driven Oncology Clinical Trials

<https://www.biopharmatrend.com/post/705-how-ai-empowers-biomarker-driven-clinical-trials/>

What is Sensor Data? Examples of Sensors and Their Uses | Definition from TechTarget

<https://www.techtarget.com/iotagenda/definition/sensor-data#:~:text=Sensor%20data%20is%20the%20output,just%20about%20any%20physical%20element>

Img2Img: Stable Diffusion Image-to-image Explained - Dataconomy

<https://dataconomy.com/2023/08/28/img2img-stable-diffusion-image-to-image/>

AI Glossary: Artificial Intelligence Terms - Dataconomy

<https://dataconomy.com/2022/04/23/artificial-intelligence-terms-ai-glossary/>

How sensors data helps to detect, predict anomalies, defects using machine learning . #BigData #Analytics #DataScience #MachineLearning #NLProc #IoT #IIoT #PyTorch #Python #RStats #Sensors

How to collect data from sensors using machine learning?

For gathering the dataset, first you get raw data from sensors on the sensor node, and then transmit these data to a machine for storage. In the next step, you use preprocessing methods to integrate automatically collected data with manually labeled data for training ML algorithms.

What is sensor data in machine learning?

Sensor data is the output of a device that detects and responds to some type of input from the physical environment. The output may be used to provide information to an end user or as input to another system or to guide a process. Sensors can be used to detect just about any physical element.

For e.g Chandrayaan - 3 too has lot of sensors inbuilt to detect and conduct experiments on the moon using AI, Machine Learning algorithms.

How does sensor data work?

Sensors gather and generate information based on the physical conditions surrounding them. Sensors usually include the following:

1. A processor to convert physical signals into digital data;
2. Communications capabilities to transmit data to people or machines; and
3. A power source.

IoT sensor data exists in three stages on the network that involve elements of data management:

1. Creation. The sensor collects signals and turns them into data.
2. Transmission. Data generated is sent to other machines using network protocols, such as MQ Telemetry Transport, Hypertext Transfer Protocol and Constrained Application Protocol. Transmission methods vary based on loss-tolerance, security and timeliness requirements.
3. Storage. Data is stored in various formats and accessed for use, data analysis and forecasting. In some cases, it is sent in real time immediately after creation. In others, it is stored for a period of time before being sent to its next destination in batches. Storage and bandwidth limitations can dictate the amount of data transmitted and the way it's sent. Cloud-based storage is used for high-volume sensor data.

Read more at

References

<https://www.techtarget.com/iotagenda/definition/sensor-data#:~:text=Sensor%20data%20is%20the%20output,just%20about%20any%20physical%20element>

<https://colab.research.google.com/drive/1tLjOg2ZQuICIfuWrAC2LdiZHCoV8oUbs?usp=sharing#scrollTo=3mTyrfkxTJBQ>

Haystack and Huggingface are both popular frameworks in the field of natural language processing (NLP) and machine learning.

1. Haystack:

Haystack is an open-source Python library developed by Deepset AI. It is specifically designed to simplify and accelerate the development of question-answering systems and search applications. Haystack provides a set of powerful tools and pre-built components for building end-to-end pipelines for tasks like document retrieval, passage ranking, and answer extraction. It integrates well with popular NLP libraries like Transformers (from Huggingface) and Elasticsearch, making it easier to leverage existing models and search technologies.

2. Huggingface:

Huggingface is a renowned organization that focuses on democratizing and advancing NLP technologies. They have developed a wide range of NLP libraries and tools that have gained significant popularity in the community. One of their most notable contributions is the Transformers library, which provides a comprehensive collection of pre-trained models for tasks like text classification, named entity recognition, language translation, and more. Transformers allows users to easily load, fine-tune, and utilize state-of-the-art models such as BERT, GPT, and RoBERTa. Huggingface also offers a model hub called the Huggingface Model Hub, where users can discover and share pre-trained models.

Accepted papers will be given guidelines in preparing and submitting the final manuscript(s) together with the notification of acceptance.

International Journal of Game Theory and Technology (IJGTT)

International Journal of Ambient Systems and applications (IJASA)

International Journal of Ubiquitous Computing (IJU)

Machine Learning and Applications: An International Journal (MLAIJ)

International Journal of Artificial Intelligence & Applications (IJAIA)

International Journal on Soft Computing (IJSC)

International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)

Advances in Vision Computing: An International Journal (AVC)

Chunking for LLMS using Haystack Huggingface

<https://cobusgreyling.medium.com/chunking-for-langs-using-haystack-huggingface-81255776d71>

Fine-Tune Your Own Llama 2 Model in a Colab Notebook | Towards Data Science

https://mlabonne.github.io/blog/posts/Fine_Tune_Your_Own_Llama_2_Model_in_a_Colab_Notebook.html

WizardLM/WizardLM-13B-V1.1 · Hugging Face

<https://huggingface.co/WizardLM/WizardLM-13B-V1.1>

lmsys/vicuna-13b-v1.3 · Hugging Face

<https://huggingface.co/lmsys/vicuna-13b-v1.3>

https://github.com/tatsu-lab/stanford_alpaca

Llamaindex - Data Framework for LLM Applications

<https://www.llamaindex.ai/>

Complex Earth Moon Rotation

<https://twitter.com/JohnTsla/status/1693967852682764329?s=20>

Learning to Identify Critical States for Reinforcement Learning from Videos

abs: <https://arxiv.org/abs/2308.07795>

paper page: <https://huggingface.co/papers/2308.07795>

Recent work on deep reinforcement learning (DRL) has pointed out that algorithmic information about good policies can be extracted from offline data which lack explicit information about executed actions [51, 32]. For example, videos of humans or robots may convey a lot of implicit information about rewarding action sequences, but a DRL machine that wants to profit from watching such videos must first learn by itself to identify and recognize relevant states/actions/rewards. Without relying on ground-truth annotations, our new method called Deep State Identifier learns to predict returns from episodes encoded as videos. Then it uses a kind of mask-based sensitivity analysis to extract/identify important critical states. Extensive experiments showcase our method's potential for understanding and improving agent behavior.

Learning to Identify Critical States for Reinforcement Learning from Videos

Haozhe Liu^{1†}, Mingchen Zhuge^{1†}, Bing Li^{1*}, Yuhui Wang¹, Francesco Faccio^{1,2}

Bernard Ghanem¹, Jürgen Schmidhuber^{1,2,3}

¹AI Initiative, King Abdullah University of Science and Technology

²The Swiss AI Lab IDSIA/USI/SUPSI, "NNAIENSE

(haozhe.liu, mingchen.zhuge, bing.li, yuhui.wang,

francesco.faccio, bernard.ghanem, juergen.schmidhuber}@kaust.edu.sa

Abstract

Recent work on deep reinforcement learning (DRL) has pointed out that algorithmic information about good policies can be extracted from offline data which lack explicit information about executed actions [51, 32]. For example, videos of humans or robots may convey a lot of implicit information about rewarding action sequences, but a DRL machine that wants to profit from watching such videos must first learn by itself to identify and recognize relevant states/actions/rewards. Without relying on ground-truth annotations, our new method called Deep State Identifier learns to predict returns from episodes encoded as videos. Then it uses a kind of mask-based sensitivity analysis to extract/identify important critical states. Extensive experiments showcase our method's potential for understanding and improving agent behavior. The source code and the generated datasets are available at GitHub.

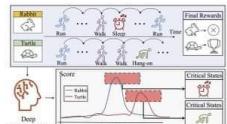


Figure 1. Motivation of the proposed method. In the illustrated race between a turtle and a rabbit, the deep state is critical in determining the winner of the race. Our method is proposed to identify such critical states.

note, or estimate [76, 37]. To avoid this pitfall, in this work we explicitly study the relationship between common

Tools for your Literature Review Project			
Literature Search	Knowledge Maps	Writing	Citations
PubMed PubMed is a free tool by NIH that supports the search and retrieval of biomedical and life sciences literature.	Research Rabbit Research Rabbit is a free, literature review tool that visualizes complex relationships in the literature.	Grammarly Grammarly is a writing assistant that checks for spelling, grammar, punctuation errors, and vocabulary usage.	Zotero Zotero is a powerful, easy-to-use research tool that helps you organize, analyze and cite academic literature sources.
Google Scholar Google Scholar is a free tool that lets you search the latest scholarly articles, documents and books.	Connected Papers Connected Papers is a tool that helps researchers find and explore papers and trends relevant to their field of work.	PaperPal PaperPal helps researchers search faster with real-time suggestions for in-depth language and grammar correction.	Mendeley Mendeley is a citation manager that lets you collect & organize citations, & then easily insert them into documents.
Elicit Elicit is a free app that helps you find papers, extract data, summarize, brainstorm ideas, and more.	LitMaps LitMaps helps you find articles and papers to your interests. It generates a map of the most relevant articles related to your seed paper.	Quillbot Quillbot's AI-powered paraphrasing tool helps to rewrite, edit, and change the tone of their text to improve clarity.	EN EndNote is a personal citation/bibliography manager tool that helps researchers create bibliographies and format references in MS word.

Tools for literature review project

ARM vs Intel: What Are They? What is the Difference Between Them?

<https://www.atatus.com/blog/arm-vs-intel-what-are-they-what-is-the-difference-between-them/#:~:text=Intel%20or%20AMD%2C%20which%20both,a%20run%20for%20its%20money.>

A Gentle Introduction to Calculating the BLEU Score for Text in Python

<https://machinelearningmastery.com/calculate-bleu-score-for-text-python/>

Statistical source expansion for question answerin

<https://dl.acm.org/doi/10.1145/2063576.2063632>

A source expansion algorithm automatically extends a given text corpus with related content from large external sources such as the Web. The expanded corpus is not intended for human consumption but can be used in question answering (QA) and other information retrieval or extraction tasks to find more relevant information and supporting evidence. We propose an algorithm that extends a corpus of seed documents with web content, using a statistical model to select text passages that are both relevant to the topics of the seeds and complement existing information.

Inc42 Media: The Game Of Multiples: Analysing Startup Valuations

<https://inc42.com/resources/the-game-of-multiples-analysing-startup-valuations/>

Analytics India Magazine: Top 7 Generative AI Courses by AWS

<https://analyticsindiamag.com/top-7-generative-ai-courses-by-aws/>

Swarajya: How India Is Using AI To Build The Internet For Local Languages

<https://swarajyamag.com/science/how-india-is-using-ai-to-build-the-internet-for-local-languages>

CNBCTV18: How multiple Indian farmers are cultivating world's most expensive Miyazaki mangoes to earn millions

<https://www.cnbctv18.com/travel/food-and-drinks/how-multiple-indian-farmers-are-cultivating-worlds-most-expensive-miyazaki-mangoes-to-earn-millions-17378681.htm>

Deccan Herald: DH Toon: 'Kartavya' limited to campaigns

<https://www.deccanherald.com/national/dh-toon-kartavya-limited-to-campaigns-1241826.html>

Analytics India Magazine: Andrew Ng & HuggingFace Introduce New Course on Generative AI

<https://analyticsindiamag.com/andrew-ng-huggingface-introduce-new-course-on-generative-ai/>

Top 10 Free Specialised Courses by Andrew Ng

<https://analyticsindiamag.com/top-10-free-specialised-courses-by-andrew-ng/>

Analytics Insight: Sam Altman's Worldcoin Surpasses 2 Million Sign-ups, InQubeta (QUBE) Set to Deliver Impressive 50x ROI to Its Investors

<https://www.analyticsinsight.net/sam-altmans-worldcoin-surpasses-2-million-sign-ups-inqubeta-qube-set-to-deliver-impressive-50x-roi-to-its-investors/>

Best Transformer Based LLMs on HuggingFace (Part 1)

<https://analyticsindiamag.com/best-transformer-based-langs-on-huggingface-part-1/>

Web 3.0: The Future of Internet

https://www.linkedin.com/pulse/web-30-future-internet-dr-rabi-prasad-padhy-phd-computer-science?utm_source=share&utm_medium=member_android&utm_campaign=share_via

Official research release for the family of XGen models (7B) by Salesforce AI Research:

Long Sequence Modeling with XGen: A 7B LLM Trained on 8K Input Sequence Length

<https://huggingface.co/Salesforce/xgen-7b-8k-base?ref=blog.salesforceairresearch.com>

Meet DORSal: A 3D Structured Diffusion Model for the Generation and Object-Level Editing of 3D Scenes

<https://www.marktechpost.com/2023/06/30/meet-dorsal-a-3d-structured-diffusion-model-for-the-generation-and-object-level-editing-of-3d-scenes/>

Step-by-Step BERT Implementation Guide - Analytics Vidhya

<https://www.analyticsvidhya.com/blog/2023/06/step-by-step-bert-implementation-guide/>

Beebom: 12 Best Large Language Models (LLMs) in 2023

<https://beebom.com/best-large-language-models-langs/>

Unity Announce the Release of Muse: A Text-to-Video Games Platform that lets you Create Textures, Sprites, and Animations with Natural Language

<https://www.marktechpost.com/2023/06/30/unity-announce-the-release-of-muse-a-text-to-video-games-platform-that-lets-you-create-textures-sprites-and-animations-with-natural-language/>

The Indian Express: ChatGPT is just the tip of the iceberg: 10 AI tools that are way cooler than OpenAI's chatbot

<https://indianexpress.com/article/technology/artificial-intelligence/ai-tools-that-are-way-cooler-than-openais-chatgpt-8694202/>

Free Neo4j Beginners Courses

https://graphacademy.neo4j.com/categories/beginners/?utm_content=day5&utm_medium=email&utm_source=freeflow

Using Langchain and Open Source Vector DB Chroma for Semantic Search wit...

<https://youtu.be/5NG8mefEsCU>

Machine Learning Pipelines for Kubeflow

<https://github.com/kubeflow/pipelines>

Video series on Machine learning theory and how to apply the algorithms

https://youtube.com/playlist?list=PLw0PWVv7CMzmzGtVeDXmR3qbf_0ifXPfl

Abacus.AI - Effortlessly Embed Cutting Edge AI In Your Applications.

https://abacus.ai/chat_llm

How can you build your own custom ChatGPT-like system on your data?

This is not easy as it could require complex architecture and pipelines.

Explore the ChatLLM feature by #abacusai.

Let's take a look at how it works:

Everyone has a knowledge base or data sitting around, like wiki pages, documentation, customer tickets, etc.

With ChatLLM you can quickly create a chat app, like ChatGPT, that helps you discover and answer questions about your data.

With #abacusai you point your system to a knowledge base or set of documents.

It ingests the data and creates the necessary pipelines, chunks the data, and sets up the essential components like embedding lookup.

You can use LLM APIs or open-source models.

#abacusai offers you many options for how to build LLM-powered chat systems.

You can fine-tune your own base models on your data.

Or you can use LLM APIs without needing to worry about the technical solution.

Your ChatLLM can be trained and is deployable in hours. It's also scalable and retrainable!

The ChatLLM also remembers the context, which is useful for longer chats.

The model responses can be tuned to a certain style and allow easy navigation to source content.

Here is a nice demo from the team showing how easy it is to create a custom LLM chat app.

<https://t.co/vvWuIuqGKT>

Nice #abacusai has this feature.

It's a really quick way for anyone to discover value in data using LLMs.

If you have a ton of data just sitting around and you want to explore it with LLMs, I highly recommend you try out ChatLLM: <https://t.co/zWDrRpVpXH>

Caution : "Make sure your organisation private data is protected, followed GDPR policies and regulations and acts"

https://github.com/nani2357/Customer_Personality_Analysis-Clustering/blob/c941566945ef20fd056b799b28b9c79320dc6f99/Final_ModelDev.ipynb

ChatGPT can help you write a standout resume in seconds—here's how

<https://www.cnbc.com/2023/03/22/chatgpt-can-help-you-write-a-standout-resume-in-seconds-heres-how.html>

Let's learn Data Science in 2023, All are FREE

- Harvard
- Stanford
- MIT
- Microsoft
- Google
- IBM

› Python

<http://learn.microsoft.com/en-us/training/patterns/beginner-python/>

› SQL

<http://online.stanford.edu/courses/soe-yd>

> PowerBI (Data Analytics)

<http://learn.microsoft.com/en-us/users/collinschedler-0717/collections/m14nt4rdwnwp04>

› Data Wrangling

<http://pll.harvard.edu/course/data-science-wrangling>

› Data Analysis

<http://cognitiveclass.ai/courses/course-v1:CognitiveClass+DA0101EN+v2>

› Data Visualization

<http://pll.harvard.edu/course/data-science-visualization>

› R

<http://pll.harvard.edu/course/data-science-r-basics>

› Statistics

<http://cognitiveclass.ai/courses/statistics>

> Linear Algebra

<http://pll.harvard.edu/course/data-analysis-life-sciences-2-introduction-linear-models-and-matrix-algebra>

› Probability

<https://pll.harvard.edu/course/data-science-probability>

› Machine Learning

<http://developers.google.com/machine-learning/crash-course>

› Deep Learning

introtodeeplearning.com

FREE certification course from Harvard University:

1. Computer Science

<https://pll.harvard.edu/course/cs50-introduction-computer-science?delta=0>

2. Programming with Scratch

<https://pll.harvard.edu/course/cs50s-introduction-programming-scratch?delta=0>

3. Web Programming with Python and JavaScript

<https://pll.harvard.edu/course/cs50s-web-programming-python-and-javascript?delta=0>

4. Artificial Intelligence

<https://pll.harvard.edu/course/cs50s-introduction-artificial-intelligence-python?delta=0>

5. Programming with Python

<https://edx.org/course/cs50s-introduction-to-programming-with-python>

6. Mobile App Development with React Native

<https://pll.harvard.edu/course/cs50s-mobile-app-development-react-native?delta=0>

7. Introduction to Data Science with Python
<https://pll.harvard.edu/course/introduction-data-science-python?delta=0>

8. Introduction to Game Development
<https://pll.harvard.edu/course/cs50s-introduction>

List of Awesome AI Agents

List of Awesome AI Agents like AutoGPT and BabyAGI
27 open-source Agents with code included!
<https://github.com/e2b-dev/awesome-ai-agents>

Auto-GPT - Github: <https://github.com/Significant-Gravitas/Auto-GPT>
BabyAGI -Github: <https://github.com/yoheinakajima/babyagi>
Voyager - Github: <https://github.com/MineDojo/Voyager> / Paper: <https://arxiv.org/abs/2305.16291>
Super-AGI - Website: <https://superagi.com/> Youtube Video about the project: <https://youtu.be/Unj5NLNTkLY>

ChemCrow: Augmenting large-language models with chemistry tools
Github: <https://github.com/ur-whitelab/chemcrow-public/> Paper: <https://arxiv.org/abs/2304.05376>

ML for economics

Explore crash course on #MachineLearning specifically tailored for economists? 🤔💡

Colin Cameron from #ucdavis has some very nice slides offering a clear and accessible overview on the subject! 

Links:

Basics: selection, shrinkage, dim reduction, LASSO
<https://t.co/26Bw32SrOH>

Flexible methods: incl. random forests, classification & cluster analysis
<https://t.co/axSGkJdxem>

Use in economics of ML (incl. prediction, causal inference)
<https://t.co/w57Xx93iTa>

More stuff on Colin's #MachineLearning page:

<https://t.co/zrfJonZrf>

Noteable Plugin: The ChatGPT Plugin That Automates Data Analysis - KDnuggets

<https://www.kdnuggets.com/2023/06/noteable-plugin-chatgpt-plugin-automates-data-analysis.html>

From Google Colab to a Ploomber Pipeline: ML at Scale with GPUs - KDnuggets

<https://www.kdnuggets.com/2022/03/google-colab-ploomber-pipeline-ml-scale-gpus.html>

ColabCode: Deploying Machine Learning Models From Google Colab - KDnuggets

<https://www.kdnuggets.com/2021/07/colabcode-deploying-machine-learning-models-google-colab.html>

Running Redis on Google Colab - KDnuggets

<https://www.kdnuggets.com/2022/01/running-redis-google-colab.html>

Abacus.AI - Effortlessly Embed Cutting Edge AI In Your Applications.

<https://abacus.ai/featurestore>

What's going on with the Open LLM Leaderboard?

<https://huggingface.co/blog/evaluating-mmlu-leaderboard>

Become a Full Stack Developer for FREE

HTML → <http://html.spec.whatwg.org/multipage/>

CSS3 → <http://web.dev/learn/css/>

Javascript → <http://LearnJavaScript.online>

React → <http://reactjs.org>

Python → <http://python.org>

Java → <http://java67.com>

Ruby → <http://gorails.com>

SQL → <http://SQLbolt.com>

MongoDB → <http://learn.mongodb.com>

AWS → <http://aws.amazon.com/training>

Azure → <http://learn.microsoft.com/en-us/training>

Git & GitHub → <http://LearnGitBranching.js.org>

Google Cloud → <http://cloud.google.com/edu>

AdobeXD → <http://youtu.be/WEljsc2jorI>

Get study materials, cheatsheets, interview questions on data science, machine learning and deep learning

#datascience #machinelearning #ai #Python #python3 #sql #deeplearning

#computervision #computerscience #programming #bigdata #chatgpt #deeplearningalgorithms

Language is remarkably nuanced

<https://blog.google/technology/ai/lamda/>

Language is remarkably nuanced and adaptable. It can be literal or figurative, flowery or plain, inventive or informational. That versatility makes language one of humanity's greatest tools — and one of computer science's most difficult puzzles.

Techniques to Cut the Costs of Using ChatGPT and GPT-4

<https://www.analyticsinsight.net/techniques-to-cut-the-costs-of-using-chatgpt-and-gpt-4/>

How to Draw Graphs, Charts, and Diagrams in ChatGPT | Beebom

<https://beebom.com/how-draw-graphs-charts-diagrams-chatgpt/>

How to Write A Business Plan PDF - PDF Gate

<https://pdfgate.net/how-to-write-a-business-plan-pdf/>

Database Sharding Explained

<https://architecturenotes.co/database-sharding-explained/>

Meet Gorilla: UC Berkeley and Microsoft's API-Augmented LLM Outperforms GPT-4, Chat-GPT and Claude.

<https://link.medium.com/B0B80qKzvAb>

How to Build ML Applications on the AWS Cloud with Kubernetes and oneAPI

<https://link.medium.com/4JiHWLeNqyb>

TestNYC-Benchmark-GPT4

https://github.com/jerryliu/llama_index/blob/main/examples/test_wiki/TestNYC-Benchmark-GPT4.ipynb

Neural Networks

Solving dense prediction tasks in computer vision (e.g., object detection or segmentation) requires a combination of semantic and spatial information. But, typical neural network architectures fail to capture this naturally. Here's how we can fix this...

When we look at a typical neural net architecture such as a ResNet, it is separated into several “sections”. Within each section, we reduce the spatial resolution of our features (i.e., via strided convolution or pooling) and increase the channel dimension.

Earlier layers with high spatial dimension capture spatial information. Later layers with higher channel dimension capture more semantic information. For more info on what different neural net layers learn, check out the link below.

🔗: <https://t.co/2iJ7yWUIf>

Dense prediction tasks require both spatial and semantic information. Feature pyramid networks (FPNs) extract this information by combining feature representations at different network layers. Features are resized and added together to capture diverse information.

This multi-scale approach is also useful for deep learning on video. Multiscale vision transformers (MViTs) aggregate features from different transformer layers to synthesize features that capture spatial, semantic, and motion info in a video.

🔗: <https://t.co/vCATXqSIUc>

For more details on multi-scale neural network architectures such as FPN and MViT, check out recent newsletter post below.

🔗: <https://t.co/oxlbeiLW1W>

GPT4-Multimodal

GPT-4 is Powerful as it is now multimodal and is capable of understanding both text and image inputs🔥

Here is how you can access GPT-4 model with just a few lines of code and use it:👉

Not just accessing GPT-4, you can create production-ready APIs and deploy apps powered by GPT-4 with #GetSteamship

Check out the Plugin here:👉

<https://t.co/FiCGea3dtm>

How to evaluate and compare results between GPT-3, GPT-4, ChatGPT?

For this analysis, explore with a simple vector index (<https://t.co/m1qnXSaKpi>) over the NYC Wikipedia article.

Researchers benchmark on a small set of queries that 🐾LLAMAIndex (GPT Index) gets wrong. These queries ask specific facts buried in the article. So they require accurate retrieval of relevant context.

For LLM-assisted evaluation, researchers pass the query, the retrieved context, and the answer to GPT-4, ChatGPT, and GPT-3 for comparison.

Then they prompt the LLMs to evaluate 1) if the context is *relevant* to the question, and 2) if the answer is *correct*.

See below attached pics

First Q: "When was the first documented visit into New York Harbor?"

GPT-4: matches human eval (irrelevant context but correct answer)

ChatGPT: thinks "there's an earlier visit in 1609"

GPT-3: thinks "the establishment of the Five Boroughs" is relevant

Winner: *GPT-4*

See below attached pics

Second Q: "What are the airports in New York City?"

Again, retrieves an irrelevant context, but hallucinates the correct answer.

GPT-4: matches human eval

ChatGPT: matches human eval

GPT-3: thinks "sporting venues in NYC" is relevant

Winner: GPT-4 & ChatGPT

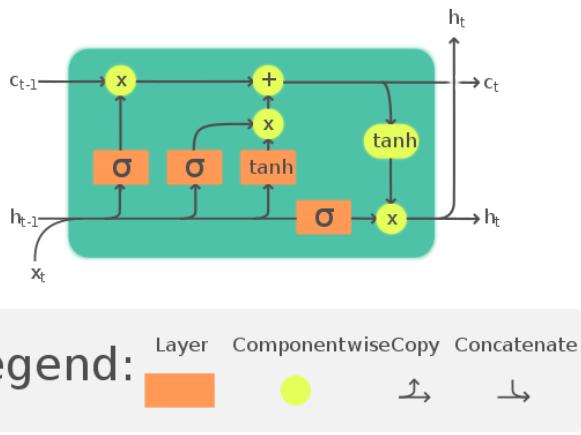
See pic

Over a small benchmark, GPT-4 consistently gives better evaluation results than GPT-3 and ChatGPT.

See this demo notebook for more details: <https://t.co/PSZ4ht3wZs>

While not always perfect, GPT-4 shows impressive critical reasoning.

Researchers are beyond excited about the (near) future where LLMs not only help with retrieval & synthesis, but also automated evaluation and optimization.



<https://github.com/Yimeng-Zhang/feature-engineering-and-feature-selection/blob/master/A%20Short%20Guide%20for%20Feature%20Engineering%20and%20Feature%20Selection.md>

<https://link.medium.com/d45qrOxcYxb>

<https://link.medium.com/gDMRvllqWxb>

<https://github.com/llyasviel/ControlNet>

Learn to use famous Deep RL libraries such as Stable Baselines3, RL Baselines3 Zoo, Sample Factory and CleanRL.

Train agents in unique environments such as SnowballFight, Huggy the Doggo, VizDoom (Doom) and classical ones such as Space Invaders, PyBullet and more.

LLama2

<https://github.com/facebookresearch/llama>

DocPrompting: Generating Code by Retrieving the Doc

<https://github.com/shuyanzhou/docprompting>

Awesome Transformers

<https://github.com/abacaj/awesome-transformers>

HuggingFace released PEFT

Explore #HuggingFace released PEFT, a library for parameter-efficient fine-tuning!

It enables machine learning models to achieve performance comparable to full fine-tuning while only having a small number of trainable parameters.

Large models are expensive to fine-tune on downstream tasks. What if we could achieve the same performance with a small fraction of the trainable parameters 😊?

Introducing 😊 PEFT: library for "parameter-efficient fine-tuning".

<https://t.co/Rj8eM9tlR>

<https://t.co/Mxs2BqSu5G>

With PEFT, you can fit the fine-tuning of very large models within consumer GPUs with support from other techniques like CPU offloading!

For example, you can fine-tune an OPT-6.7b with PEFT and `bitsandbytes` in @GoogleColab ... fast ⚡⚡

<https://t.co/EYmH0swfkg>

PEFT currently supports the following techniques for parameter-efficient fine-tuning:

- 1 LoRA by @edwardjhu and @WeizhuChen et. al.
- 2 Prefix tuning
- 3 Prompt tuning by @noahconst et. al.
- 4 P-tuning

Meta-Learning Fast Weight Language Models

Presents Fast Weight Layers (FWLs), a neural component that provides the benefits of dynamic evaluation much more efficiently by expressing gradient updates as linear attention.

<https://t.co/HoKY34emvd>

Salesforce open sources DeepTIME - DeepTIME, a deep time-index based model trained via a meta-learning formulation yielding an efficient and accurate forecasting model using PyTorch

Paper:

<https://t.co/06Xxfv4XxO>

Python GitHub:

<https://t.co/4YeWqdV1Vk>

Learn more ↗

<https://link.medium.com/aQ35aieWAxb>

<https://github.com/fastai/fastbook>

Cisco Networking Academy is offering this Certifications and Courses (FREE)

1. Introduction to Cybersecurity

<https://t.co/8qqoqpgG1g>

2. Cybersecurity Essentials

<https://t.co/DGHxEUaUB1>

3. Networking Essentials

<https://t.co/LtdmmnSgIh>

"How Cloud Computing is transforming from raw infra to app-centric services"

① The pre-cloud/early cloud Era:

This is the time of monolithic apps and static raw on-prem infrastructure.

- In this era, the monolith contains and does everything it needs to. Devs responsible for most.
- The OS primitives are the interface with infra (static IPs and VMs)

② The infrastructure/compute centric cloud era

This is the time of transition to microservices & raw cloud infra

- Dynamic compute, networking, and storage cloud services (not app centric)
- Containers emerging as de facto app packaging & compute format, & HTTP for networking

③ The application-centric cloud era

Raw infra moves into the background and app-first services emerge

- Container and function based compute
- Orchestration: Step Functions, GCP Workflows
- Eventing: EventBridge, EventGrid, EventArc
- DevX, logs, monitoring: all are app-first

④ Cloud-bound applications

The new cloud svc are created for Devs rather than Ops. They offer app-first compute & integration capabilities that blend with the app at much deeper levels

This changes how we look at apps from a single unit, into one that is blended with cloud.

sar

<https://towardsdatascience.com/training-xgboost-on-a-1tb-dataset-8790e2bc8672>

Data Science Deconstructed



AJGoldstein.com

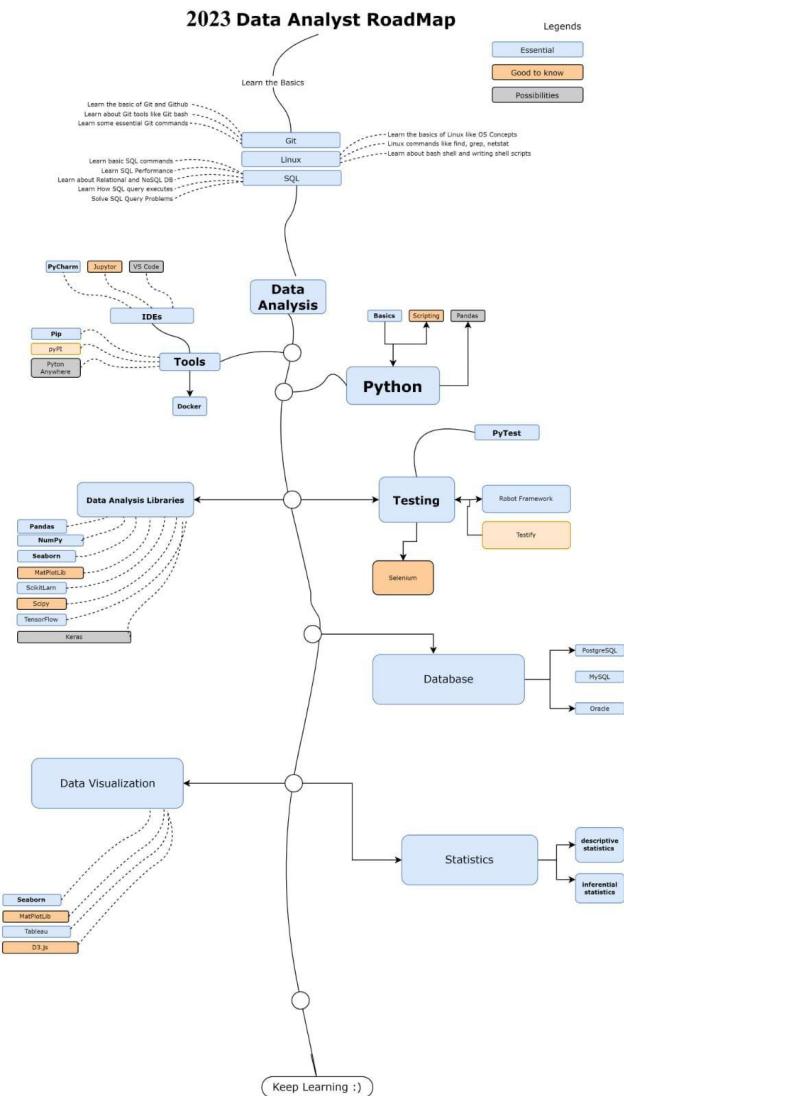
Data Science Deconstructed!

Top No-Code Machine Learning Platforms

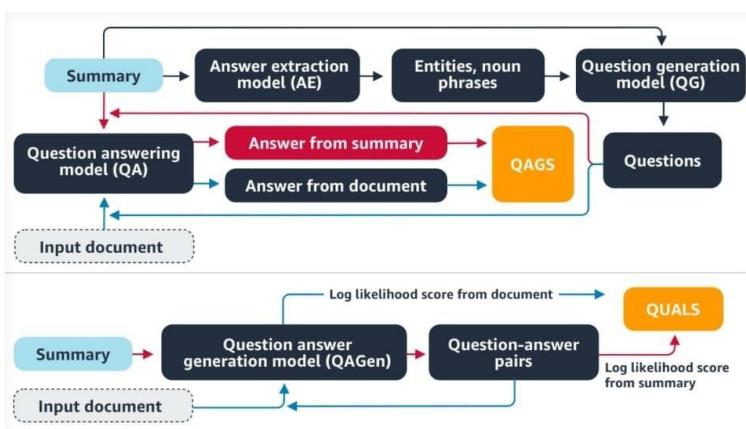
- Big ML
- Create ML
- Data Robot
- Fritz AI
- Google Cloud AutoML
- Google ML Kit
- Make ML
- Microsoft Azure Automated Machine Learning
- Obviously AI
- Runway ML
- Super Annotate
- Teachable Machine

#MachineLearning #DeepLearning

<https://github.com/mindsdb/lightwood>



Data Analyst Roadmap



Question Answer Generation

'Lessons Learned Applying NLP to Create a Web-Scale Knowledge Graph'

This session was presented by @filipe_mesquita, Vice President of Research at Diffbot at #Healthcare #NLPSummit 2022

Watch the entire video: <https://t.co/q0kb4kBGSF>

#mlops #ai #datascience #ml #dl #textmining

Using spaCy & combine with a Knowledge Graph to build an autonomous QA system. Just awesome? Yes.

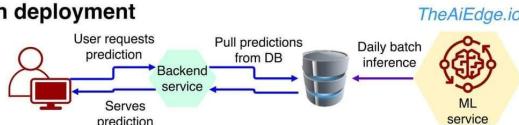
Video <https://t.co/HWK5HiF1T5>

Code <https://t.co/a8nucqoXHz>

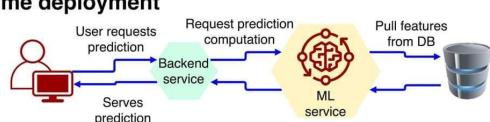
#machinelearning #QA #datascience #python #spacy 🎉

All the ways to deploy a ML model

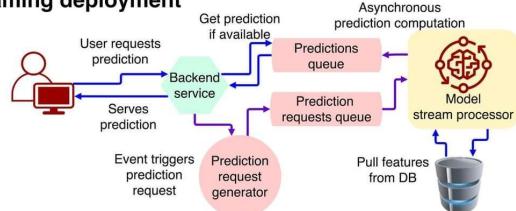
Batch deployment



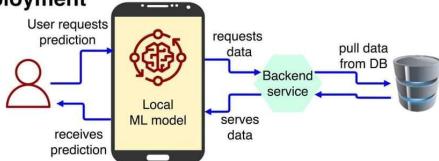
Real-time deployment



Streaming deployment



Edge deployment



All the ways to Deploy a ML Model

#MathGPT

Use GPT3 to solve math problems, and get the code behind each solution: <https://mathgpt.streamlit.app/> #Artificialintelligence #ChatGPT #DeepLearning #GPT #MachineLearning

Similarly we can explore ScienceGPT, etc... N number of GPTs, GPT++ can be expected in near future.



Selection of Appropriate Statistical Methods for Data Analysis

Prabhaker Mishra, Chandra Mani Pandey, Uttam Singh, Amit Keshri, and Mayilvaganan Sabaretnam
Ann Card Anesth. 2019 Jul-Sep; 22(3): 297–301.
doi: 10.4103/aca.ACA_248_18

Open Access Article. I share in the tweet the link for download it !
@Verokita

Table 1: Parametric and their Alternative Nonparametric Methods

Description	Parametric Methods	Nonparametric Methods
Descriptive statistics	Mean, Standard deviation	Median, Interquartile range
Sample with population (or hypothetical value)	One sample <i>t</i> -test ($n < 30$) and One sample Z-test ($n \geq 30$)	One sample Wilcoxon signed rank test
Two unpaired groups	Independent samples <i>t</i> -test (Unpaired samples <i>t</i> -test)	Mann Whitney U test/Wilcoxon rank sum test
Two paired groups	Paired samples <i>t</i> -test	Paired samples Wilcoxon signed-rank test
Three or more unpaired groups	One-way ANOVA	Kruskal-Wallis H test
Three or more paired groups	Repeated measures ANOVA	Friedman Test
Degree of linear relationship between two variables	Pearson's correlation coefficient	Spearman rank correlation coefficient
Predict one outcome variable by at least one independent variable	Linear regression model	Nonlinear regression model/Log linear regression model on log normal data

Table 2: Statistical Methods to Compare the Proportions

Description	Statistical Methods	Data Type
Test the association between two categorical variables (Independent groups)	Pearson Chi-square test/Fisher exact test	Variable has ≥ 2 categories
Test the change in proportions between 2/3 groups (paired groups)	McNemar test/Cochrane Q test	Variable has 2 categories
Comparisons between proportions	Z test for proportions	Variable has 2 categories

Table 3: Semi-parametric and non-parametric methods

Description	Statistical methods	Data type
To predict the outcome variable using independent variables	Binary Logistic regression analysis	Outcome variable (two categories), Independent variable (s): Categorical (≥ 2 categories) or Continuous variables or both
To predict the outcome variable using independent variables	Multinomial Logistic regression analysis	Outcome variable (≥ 3 categories), Independent variable (s): Categorical (≥ 2 categories) or continuous variables or both
Area under Curve and cutoff values in the continuous variable	Receiver operating characteristics (ROC) curve	Outcome variable (two categories), Test variable : Continuous
To predict the survival probability of the subjects for the given equal intervals	Life table analysis	Outcome variable (two categories), Follow-up time : Continuous variable
To compare the survival time in ≥ 2 groups with P	Kaplan-Meier curve	Outcome variable (two categories), Follow-up time : Continuous variable, One categorical group variable
To assess the predictors those influencing the survival probability	Cox regression analysis	Outcome variable (two categories), Follow-up time : Continuous variables, Independent variable(s): Categorical variables (≥ 2 categories) or continuous variable(s) or both
To predict the diagnostic accuracy of the test variable as compared to gold standard method	Diagnostic accuracy (Sensitivity, Specificity etc.)	Both variables (gold standard method and test method) should be categorical (2×2 table)
Absolute Agreement between two diagnostic methods	Unweighted and weighted Kappa statistics/Intra class correlation	Between two Nominal variables (unweighted Kappa). Two Ordinal variables (Weighted kappa), Two Continuous variables (Intraclass correlation)

Mishra P, Pandey GM, Singh U, Keshri A, Sabaretnam M. Selection of appropriate statistical methods for data analysis. Ann Card Anesth. 2019 Jul-Sep;22(3):297-301. doi: 10.4103/aca.ACA_248_18. PMID: 31274493; PMCID: PMC6539881.

Selection of Appropriate Statistical methods For Data Analysis

https://pubmed.ncbi.nlm.nih.gov/31274493/?utm_content=235553794&utm_medium=social&utm_source=linkedin&hss_channel=lcp-3740012

Contrastive learning

* Unsupervised visualization of image datasets using contrastive learning
<https://t.co/pR2tezxBhl>

* From t-SNE to UMAP with contrastive learning
<https://t.co/SrCPPSFGwJ>

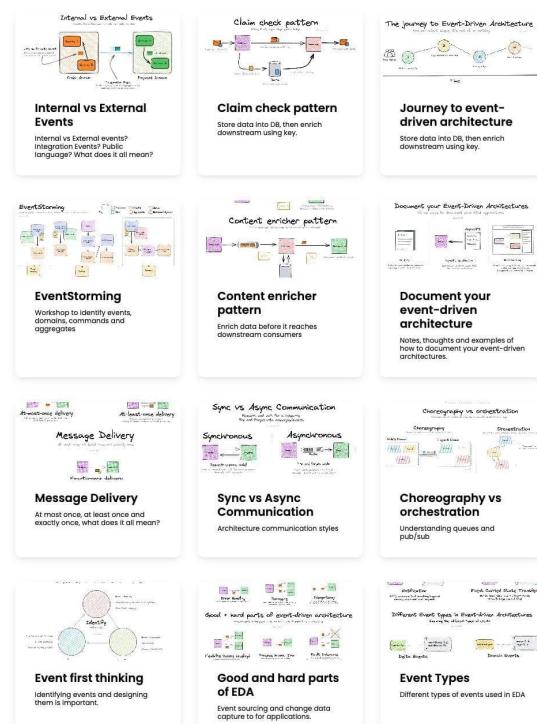
SimCLR + t-SNE = t-SimCNE 😍

Introducing t-SimCNE: a method for unsupervised 2D visualization of image datasets like CIFAR-10 or CIFAR-100.

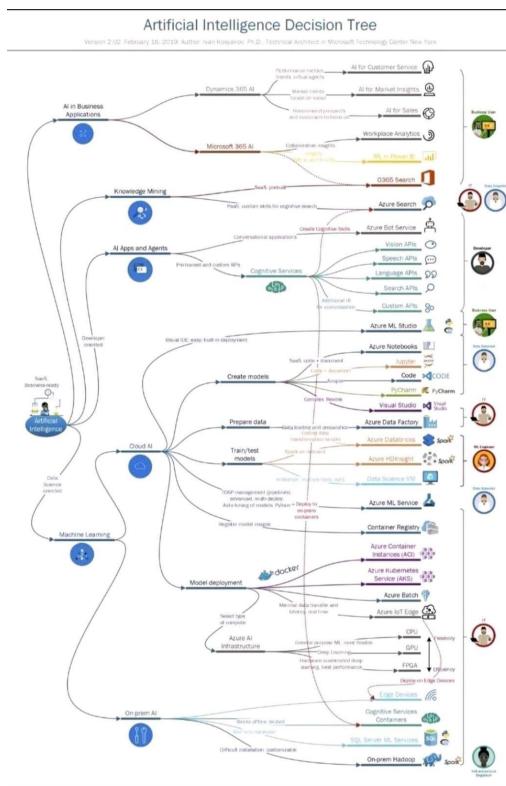
Preprint: <https://t.co/8BKfq71LCU>
Code: <https://t.co/HupGQepXfK>

EDA VISUALS

Small bite sized visuals about event-driven architectures
Designs and thoughts from [@kingsley25](#)



EDA Visuals



AI Decision Tree

Grab these top-rated AI/ Machine Learning books for FREE! 

Links in the thread

1 ISL

Tested by time and read by millions, the bible for statistical & classical machine learning.

Mathematical concepts are explained in a beginner friendly & intuitive way.

 → <https://t.co/PPyYWytnKS>

2 Deep Learning

Written By Ian Goodfellow, the creator of GANs.

Covers a broad range of topics in deep learning & takes into consideration both industrial and research perspective.

 → <https://t.co/ouEUOsx42X>

3 Data Science Handbook

The book cover all essential data science tools!

 → <https://t.co/0Yr4vzCbAI>

4 Approaching almost any ML Problem

Trick for better LLM context retrieval for those who haven't seen it:

HyDE: Hypothetical Document Embeddings

<https://t.co/8EmTIfgJFW>

Take your query => create *hypothetical* answer => embed hypothetical answer => use this to search through doc embeddings

This is great when you have a large number of documents that could serve as context (e.g. a full Google Drive or Slack history), yet the query content/embedding is semantically different than the relevant doc content.

You can implement this yourself on top of any black box LLM API and with your favorite vector search index

OR

`pip install langchain` 

<https://t.co/tFF7SsMaPf>

```
from langchain.llms import OpenAI
from langchain.embeddings import OpenAIEMBEDDINGS,
HypotheticalDocumentEmbedder
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate

base_embeddings = OpenAIEMBEDDINGS()
llm = OpenAI()

# Load with `web_search` prompt
embeddings = HypotheticalDocumentEmbedder.from_llm(llm,
base_embeddings, "web_search")

# Now we can use it as any embedding class!
result = embeddings.embed_query("Where is the Taj Mahal?")
```

HypotheticalDocumentEmbedder

// Deep Papers #1: InstructGPT //

OpenAI scientists behind InstructGPT: precursor to ChatGPT, & one of the first applications of RLHF to LLMs.

More below!

YouTube: <https://t.co/eULGxkaUhW>

Spotify: <https://t.co/xKFDEsc7TR>

Podcast highlights:

What's the motivation behind InstructGPT? What are the problems with GPT-3 that inspired InstructGPT?

Listen to the full podcast here:

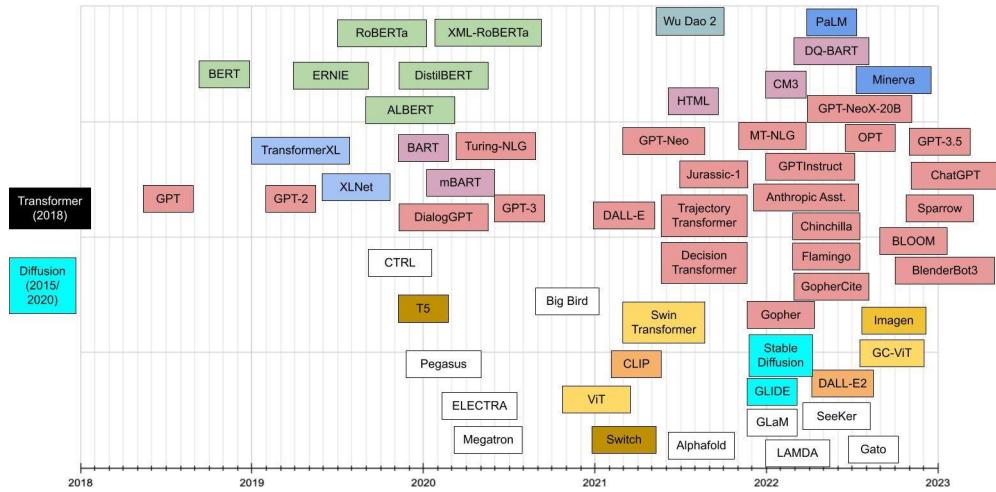
YouTube: <https://t.co/eULGxkaUhW>

Apple Podcasts:

<https://t.co/97Yk9VmRdd>

Spotify: <https://t.co/xKFDEsc7TR>

Arize AI Blog: <https://t.co/x91u2hV8F3>



Transformer Model Chronology

Lost in the coding jungle?

Find your way with these developer roadmaps:

Computer Science:

<https://t.co/VYiszD9sC4...>

Python:

<https://t.co/xx3JegCFBV>

React:

<https://t.co/kaSskSobvC>

JavaScript:

<https://t.co/wHFYD7438u>

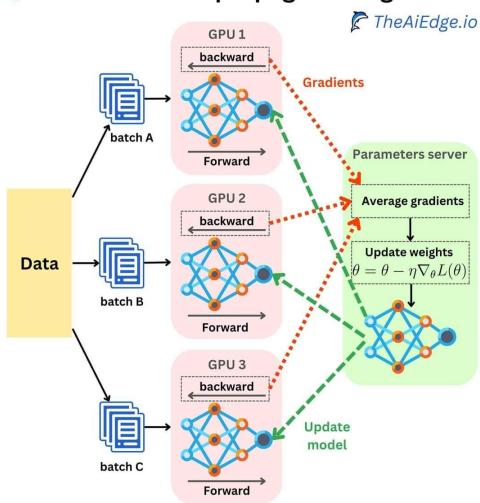
<https://github.com/microsoft/torchgeo/releases/tag/v0.4.0>

Deep Learning Meets Sparse Regularization: A Signal Processing Perspective

<https://ai.papers.bar/paper/bd7b117c9bf411ed8f9c3d8021bca7c8>

Most state-of-the-art machine learning methods are based on neural networks. Read paper which present a relatively new mathematical framework that provides the beginning of a deeper understanding of deep learning. This framework precisely characterizes the functional properties of neural networks that are trained to fit to data.

Distributed Backpropagation algorithm



Distributed Backpropagation Algorithm

Let's speed up our Machine Learning Training a notch! Do you know how the backpropagation computation gets distributed across GPUs or nodes? The typical strategies to distribute the computation are data parallelism and model parallelism.

The steps for centralized synchronous data parallelism are as follows:

1. A parameter server is used as the ground truth for the model weights. The weights are duplicated into multiple processes running on different hardware (GPUs on the same machine or on multiple machines).
2. Each duplicate model receives a different data mini-batch, and they independently go through the forward pass and backward pass where the gradients get computed.
3. The gradients are sent to the parameter server where they get averaged once they are all received. The weights get updated in a gradient descent fashion and the new weights get broadcast back to all the worker nodes.

This process is called "centralized" where the gradients get averaged. Another version of the algorithm can be "decentralized" where the resulting model weights get averaged:

1. A master process broadcasts the weights of the model.
2. Each process can go through multiple iterations of the forward and backward passes with different data mini-batches. At this point, each process has very different weights.
3. The weights get sent to the master process, they get averaged across processes once they get all received, and the averaged weights get broadcast back to all the worker nodes.

The decentralized approach can be a bit faster because you don't need to communicate between machines as much, but it is not a proper implementation of the backpropagation algorithm. Those processes are synchronous because we need to wait for all the workers to finish their jobs. The same processes can happen asynchronously, only the gradients or weights are not averaged. You can learn more about it here: <https://arxiv.org/pdf/2007.03970.pdf>

When it comes to the centralized synchronous approach, Pytorch and TensorFlow seem to follow a slightly different strategy (<https://pytorch.org/docs/stable/notes/ddp.html>) as it doesn't seem to be using a parameter server as the gradients are synchronized and averaged on the worker processes. This is how the Pytorch DistributedDataParallel module is implemented

(<https://pytorch.org/docs/stable/generated/torch.nn.parallel.DistributedDataParallel.html>), as well as the TensorFlow MultiWorkerMirroredStrategy one (https://www.tensorflow.org/api_docs/python/tf/distribute/MultiWorkerMirroredStrategy). It is impressive how simple they made training a model in a distributed fashion!

#MachineLearning #datascience #PyTorch #tensorflow

Nice overview of MLOps in dev, stage, and prod environments.

With [MLflow](#) for CI/CD/CT of ML models.

• **Dev Environment**

- Data pipeline cleans and aggregates data in a Lakehouse
- Do Exploratory Data Analysis (EDA) on that data
- Feature engineering may lead to refreshing feature tables
- Train models and track experiments
- Commit the code

• **Stage Environment**

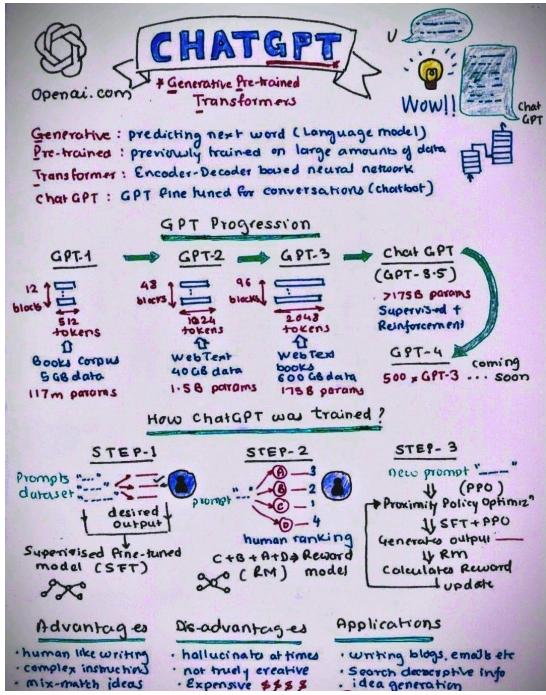
- Merge request arrives
- Run unit tests
- Run integration tests: feature store tests, model training tests, inference tests, model monitoring tests
- Commit the merge to the main branch
- Cut release branch

• **Prod Environment**

- Refresh feature table (data preparation and featurization)
- Train and tune the model: register and request model transition
- Continuous Deployment: Run compliance tests on the model, and compare staging and prod models
- Promote model to production for batch and online inference
- Monitor model performance
- Trigger model training

Link to [Databricks](#) whitepaper in the first comment.

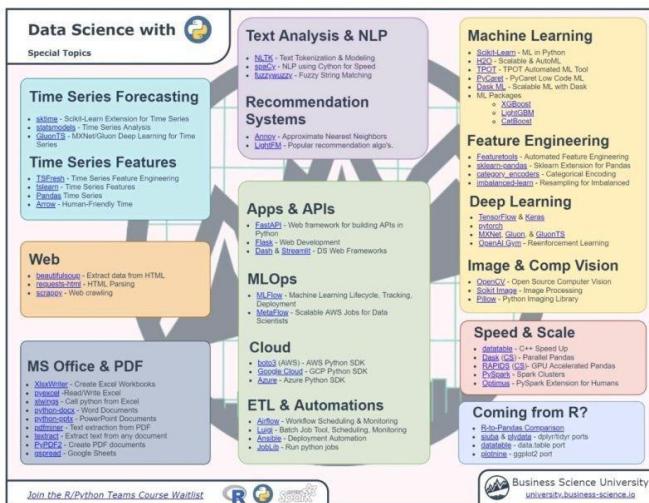
Overview of MLOps in Dev, Stage, Production Env.



GPT Progression

Google GAN

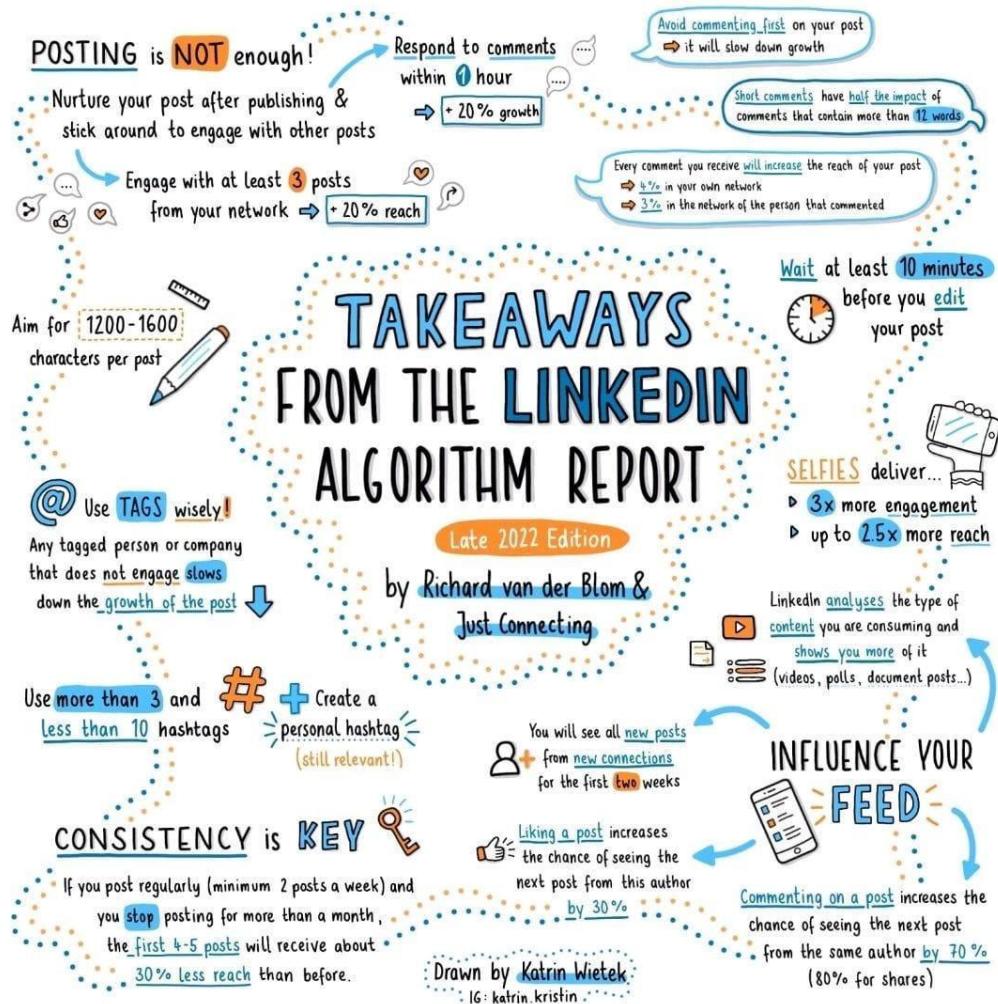
- Google party
- Google Nike
- Google imagine
- Google muse



Data Science, Machine Learning & Deep learning Topics

https://www.linkedin.com/feed/update/urn:li:activity:7022525367093534720?utm_source=share&utm_medium=member_desktop

Takeaways From the LinkedIn Algorithm Report



Machine learning debugging models

Machine Learning Model monitoring Drift

Debugging machine learning models is hard.

How to make 10x easier with end-to-end MLOps platform:

- Notebook hosting
- CI/CD
- Monitoring drift

Machine Learning Model Drift

Types, causes, detections, mitigations, and tools

<https://link.medium.com/FImarfPoNwb>

How to Detect Model Drift in MLOps Monitoring

<https://link.medium.com/9HI6bA7oNwb>

Model Monitoring and Detecting drifts in ML Models using Deepchecks

<https://www.section.io/engineering-education/model-monitoring-and-detecting-drifts-using-deepchecks/>

Did you think bringing your machine learning model to production was the hard part?

What about model drift?

Now MLOps comes into play but how does it work and what are good tools?

What is:

- Continuous integration (CI)
- Continuous deployment (CD)
- Continuous training (CT)

The full MLOps life cycle

- Data Engineering: Get and clean the data recurring if necessary
- Model Engineering: Model training, evaluation, testing, and packaging
- Model Deployment: integrating the trained model. Model serving, performance monitoring

Why is MLOps important?

Just because your model is hitting now doesn't mean it will be doing so 6 months from now

Model drift is real!

- Continuous training (CT)

Why is MLOps important?

If your machine learning model is embedded in a complete project, it is probably not alone, we have an orchestra that plays and needs to be managed

- Continuous integration (CI)

Why is MLOps important?

When the data changes, the machine learning models need to be readjusted and regularly deployed

- Continuous deployment (CD)

How do you make your life easier? With suitable tools and automation!

DAGsHub is a web platform for data version control and collaboration for data scientists & ml engineers

It's like GitHub for data science and machine learning

<https://t.co/VQHIWaFtvk>

DAGsHub in production

Continuous Delivery in Machine Learning using Jenkins and DVC an Open Source Version Control System for Machine Learning Projects

<https://t.co/iPQajVrYJ0>

#MLflow is an open source platform to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry.

<https://t.co/dPZgsO0hto>

Weights & Biases: Developer tools for machine learning

Build better models faster with experiment tracking, dataset versioning, and model management

<https://t.co/DwVOoAfaaW>

#Kubeflow the cloud-native platform for machine learning operations - pipelines, training and deployment.

Deploying and managing a complex ML system at scale.

<https://t.co/Hw85yHbF7I>

Reproducible ML pipelines for production

Key benefits of #zenml_io

- Guaranteed reproducibility of training experiments
- Cached pipeline states for faster experiment iterations
- Pre-built helpers to compare and visualize parameters and results

<https://t.co/VLHFwXX0sn>

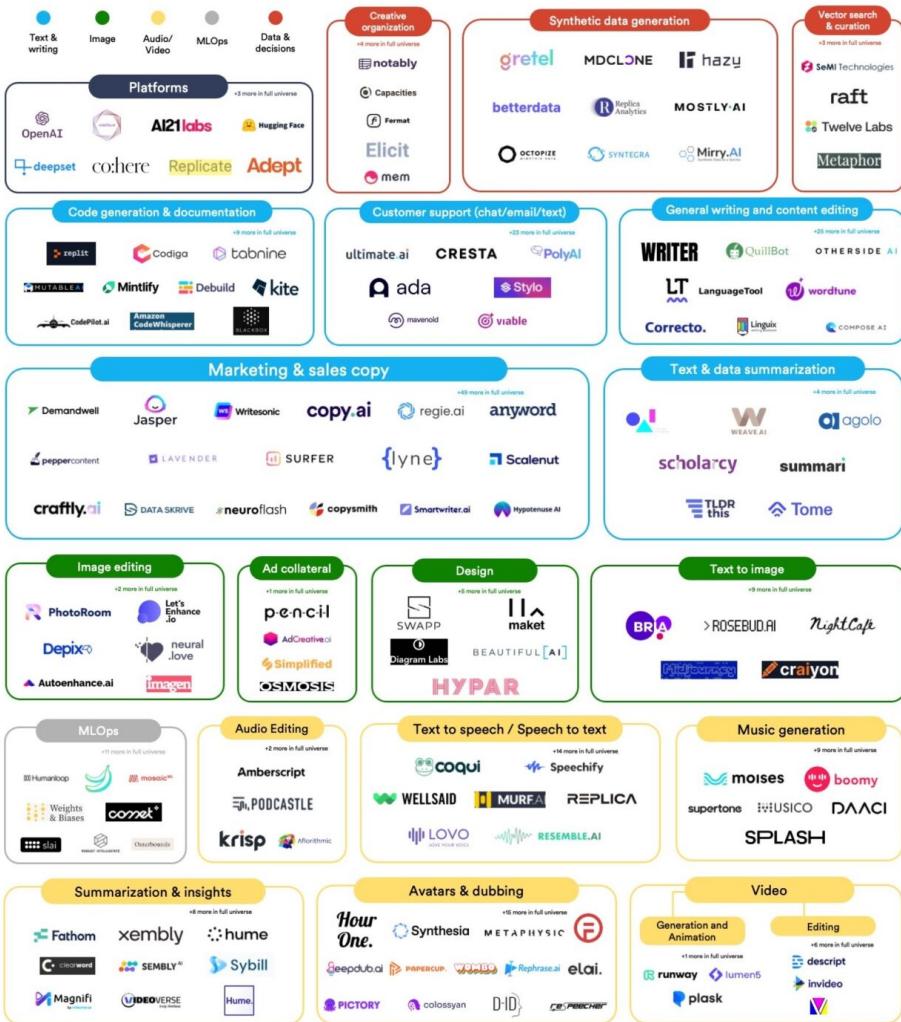
Here is the list of papers <https://t.co/atWnxmzBPU>

Can someone have consolidated of A TO Z AI, Machine Learning papers links etc like above link?

Generative AI Apps

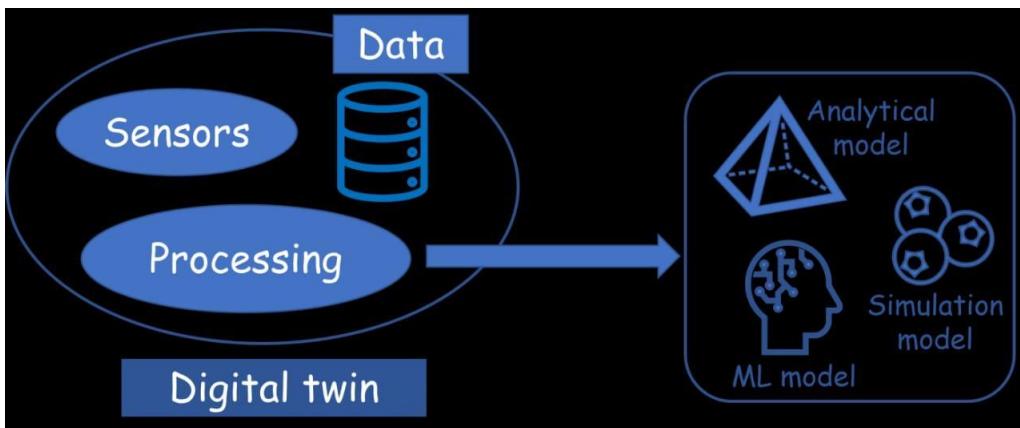
BASE10 TREND MAP: GENERATIVE AI

Companies are grouped based on medium produced and segmented by use case within each medium. Companies that offer products across segments are grouped in the segment of the core product offering.



Haikei

<https://haikei.app/>



Digital Twin

How to install, use ChatGPT

How to install, use ChatGPT <https://t.co/38hsO66aGI>

- OpenAI's ChatGPT: <https://t.co/h5zSU9nJcp>
- How to install and use Stable Diffusion: <https://t.co/P9yBZxwwwG>
- VoiceMod: <https://t.co/uih9BhPxan>

Harvard University is offering free online courses.

Learn Computer Science to Data Science for FREE in 2023

The courses will be open to anyone, with no prerequisites required.

Here are 10 FREE courses   <https://t.co/qTvbtI5loj>

1. Introduction to Computer Science

<https://t.co/mlKQfC0QIT>

2. Introduction to Programming with Scratch

An introduction to programming using Scratch, a visual programming language via which aspiring programmers can write code by dragging and dropping graphical blocks

<https://t.co/vR xv647Hqg>

3. Web Programming with Python and JavaScript

<https://t.co/HzBmdTEdkT>

4. Introduction to Programming with Python

<https://t.co/hRyuMNzlhM>

5. Understanding Technology

<https://t.co/FDKXqMe4Pj>

6. Introduction to Artificial Intelligence with Python

<https://t.co/VoINVOJQtb>

7. Introduction to Game Development

<https://t.co/wXKNw24tz8>

8. Mobile App Development with React Native

<https://t.co/e7L22cjRYn>

9. CS50's Computer Science for Business Professionals

<https://t.co/w3IxVWTNng>

10. Introduction to Data Science with Python

<https://t.co/jrip2gHGj>

AI Explainability 360

aix360 (AI Explainability 360)

aix360 includes a comprehensive set of algorithms that cover different dimensions

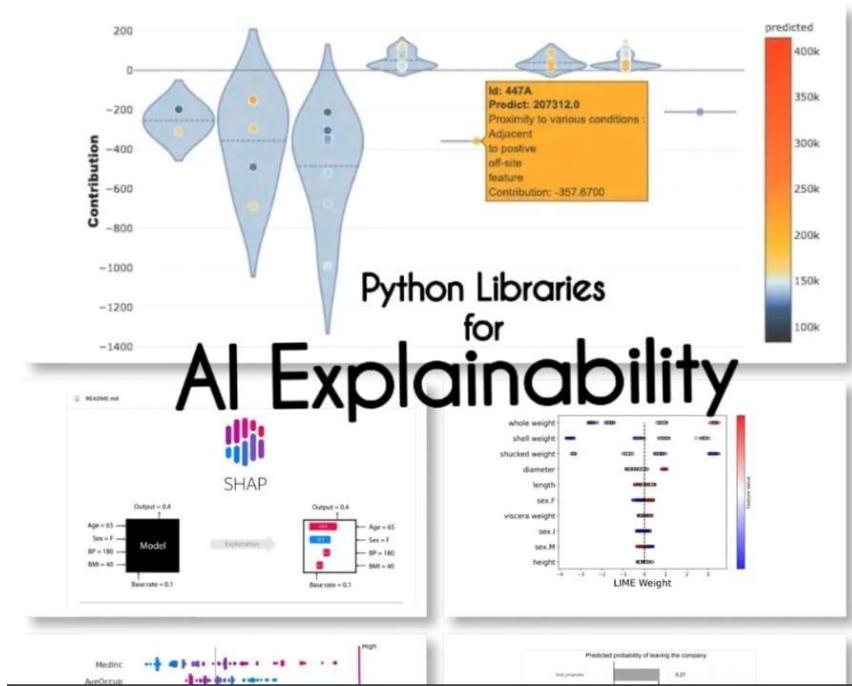
OmniXAI

OmniXAI (short for Omni eXplainable AI), addresses several problems with interpreting judgments produced by machine learning models in practice.

💡 Have I forgotten any libraries?

🔗 Source: <https://lnkd.in/dzvgSe24>

#machinelearning #artificialintelligence #ai
#deeplearning #datascience #analytics
#dataanalytics #ml #python



XAI (eXplainable AI)

XAI is a library for explaining and visualizing the predictions of machine learning models including feature importance scores.

BreakDown

BreakDown is a tool that can be used to explain the predictions of linear models. It works by decomposing the model's output into the contribution of each input feature.

interpret-text

interpret-text is a library for explaining the predictions of natural language processing models.

iml (Interpretable Machine Learning)

iml currently contains the interface and IO code from the Shap project, and it will potentially also do the same for the Lime project.

aix360 (AI Explainability 360)

aix360 includes a comprehensive set of algorithms that cover different dimensions

OmniXAI

OmniXAI (short for Omni eXplainable AI), addresses several problems with interpreting judgments produced by machine learning models in practice.

SHAP is a model agnostic and works by breaking down the contribution of each feature and attributing a score to each feature.

LIME (Local Interpretable Model-agnostic Explanations)

LIME is another model agnostic method that works by approximating the behavior of the model locally around a specific prediction.

Eli5

Eli5 is a library for debugging and explaining classifiers. It provides feature importance scores, as well as "reason codes" for scikit-learn, Keras, xgboost, LightGBM, CatBoost.

Shapash

Shapash is a Python library which aims to make machine learning interpretable and understandable to everyone. Shapash provides several types of visualization with explicit labels.

Anchors

Anchors is a method for generating human-interpretable rules that can be used to explain the predictions of a machine learning model.

XAI (eXplainable AI)

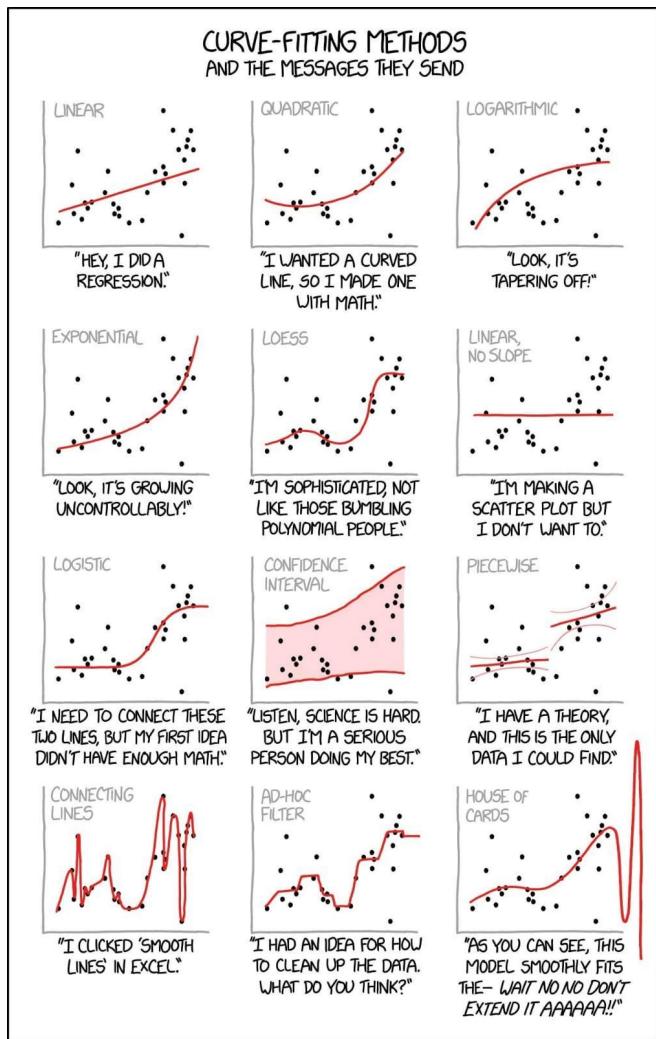
XAI is a library for explaining and visualizing the predictions of machine learning models including

Renewable Energy

What is Renewable Energy? How is machine learning used for renewable energy?
#RenewableEnergy #MachineLearning #AI #DataScience #MicroGrid

https://www.linkedin.com/posts/bhaskarareddysannapureddy_what-is-renewable-energy-how-machine-learning-activity-7020236476202307584-nwM1?utm_source=share&utm_medium=member_android

Curve Fitting Methods



(LLMs) going to impact search engine providers

How large language models (LLMs) going to impact search engine providers?

Many of the most notable publications on this topic come from Google:

1. Gopher: <https://t.co/hvqUhWIP8F>
2. Chinchilla: <https://t.co/lIkGKJVQk>
3. LaMDA: <https://t.co/km8CWz86oJ>
4. Dramatron: <https://t.co/pCeXBcID15>
5. Sparrow: <https://t.co/ahfyPSHJgy>

Improving alignment of dialogue agents via targeted human judgements
<https://arxiv.org/abs/2209.14375>

Plus, Google is one of the largest cloud platform providers along with Microsoft, Amazon, etc but they already have the needed infrastructure to train and host LLMs.

Image Source: <https://t.co/fwJDLQTCDI>

Currently, LLMs like Instruct/ChatGPT have a major problem with outputting reasonable-sounding information

that is actually false. Even the best models are likely to “hallucinate” pretty frequently.

As a solution to this problem, Google has explored teaching LLMs how to support their factual claims with information from the internet. This works relatively well, but more importantly might give humans a more dependable way to verify LLM output.

A more reasonable possibility is the use of LLMs and search engines in tandem. These tools could augment each other to create a useful hybrid. This idea is being tested by platforms like #YouSearchEngine with their YouChat feature.

Try it out: <https://t.co/1EyeJ8Zd1>

TL;DR:

- Yes, LLMs like ChatGPT are exciting.
- Are they a replacement for search? No, not yet at least.
- If LLMs replace search, Google will be in front of it.
- LLMs/search are different, their use cases probably will be too.
- Maybe LLMs + search are a good combo.

Read paper by Google <https://t.co/y7kK3Sf29n>

Let us wait and watch for LLM advancements and innovations.....

Typical Information Retrieval Metrics

Metric	Name	Formula
Hit Rate	HR	$\frac{ S_{hit} }{ S_{all} }$
Precision	P	$\frac{\text{\# of correct recommendations}}{\text{\# of recommendations}}$
Average Precision	$AP@N$	$\frac{1}{N} \sum_{i=1}^N \mathbb{I}_{hit} P_i$
Mean Average Precision	MAP	$\frac{1}{ S_{all} } \sum_{s=1}^{ S_{all} } AP@N_s$
Discounted Cumulative Gain	DCG	$\sum_{i=1}^N \frac{G_i}{\log(i+1)}$
Normalized Discounted Cumulative Gains	$NDCG$	$\frac{DCG}{IDCG}$
Average Reciprocal Hit Ratio	$ARHR$	$\frac{1}{ S_{all} } \sum_{s=1}^{ S_{all} } \sum_{i=1}^N \frac{\mathbb{I}_{hit}}{i}$

$|S_{hit}|$ Numbers of successful sessions

$|S_{all}|$ Total number of sessions

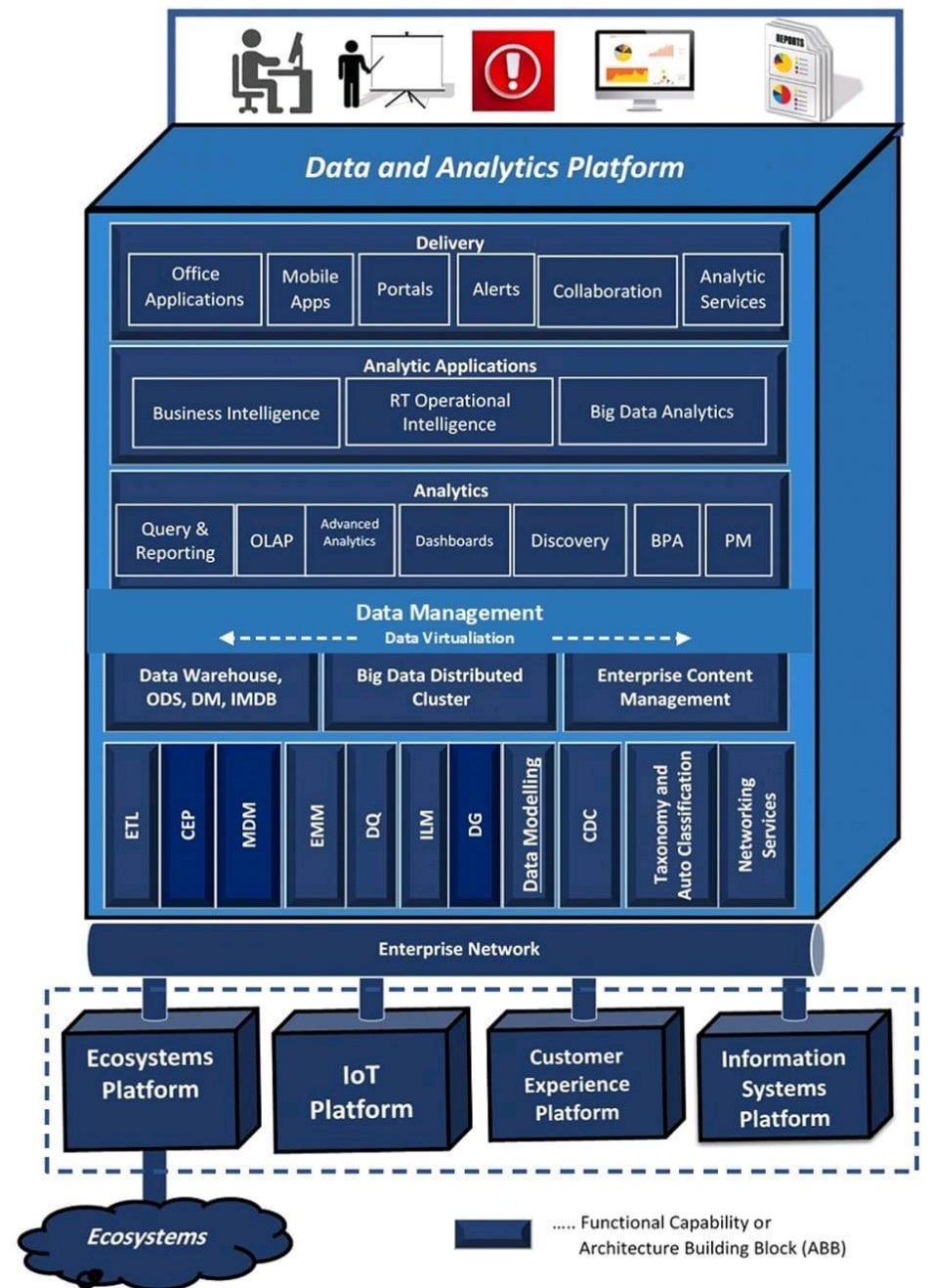
N Window size

G_i Gain at rank i

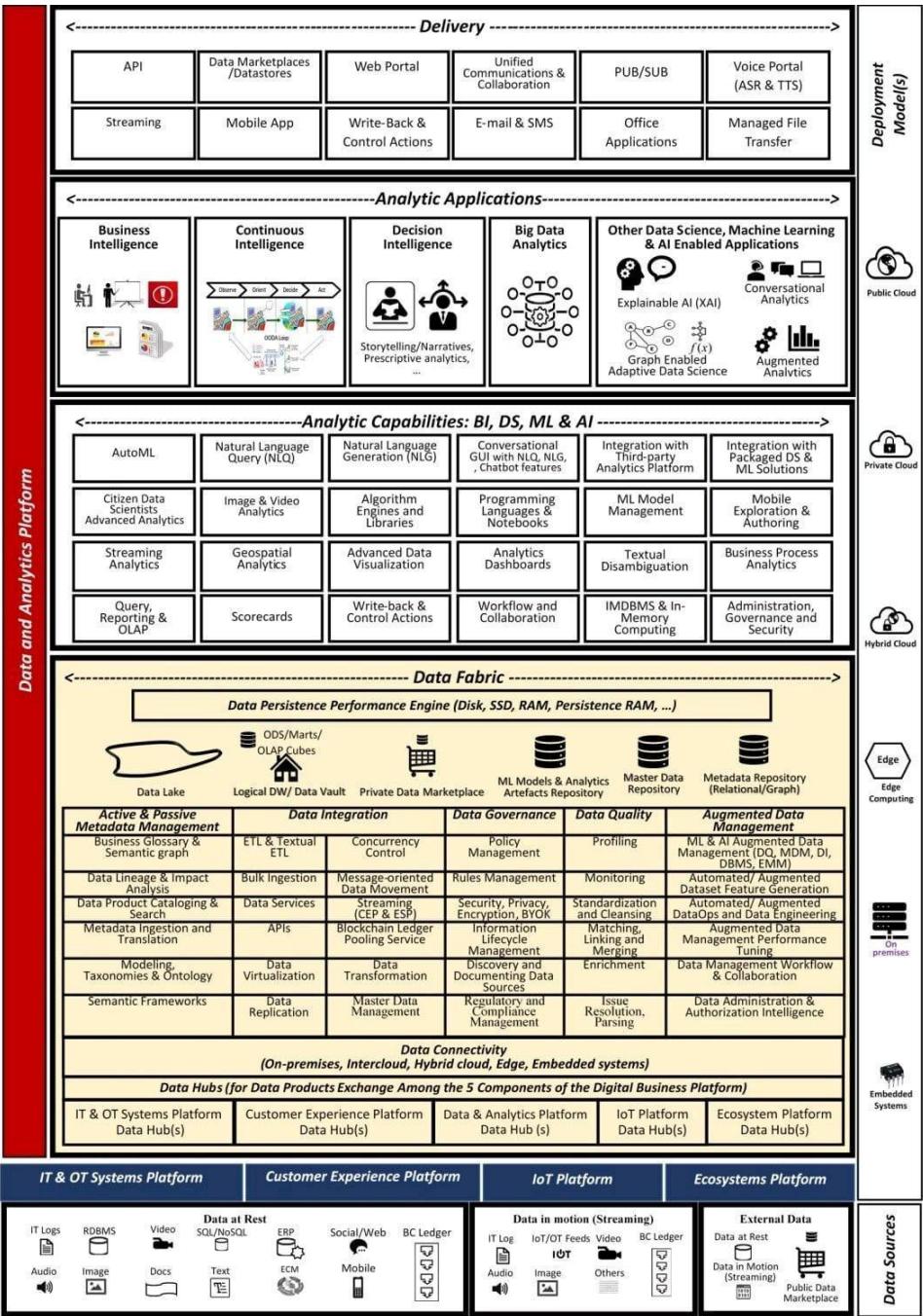
$IDCG$ Ideal Discounted Cumulative Gain

Information Retrieval Metrics

Data and Analytics Platform - 1



Data and Analytics Platform - 2



How to Modernize Enterprise Data

How to Modernize Enterprise Data and Analytics Platform (Part 1 of 4):

<https://www.datasciencecentral.com/how-to-modernize-enterprise-data-and-analytics-platform-part-1-of-4/>

How to Modernize Enterprise Data and Analytics Platform (Part 2 of 4):

<https://www.datasciencecentral.com/how-to-modernize-enterprise-data-and-analytics-platform-part-2-of-4/>

How to Modernize Enterprise Data and Analytics Platform (Part 3 of 4):

<https://www.datasciencecentral.com/part-3-of-4-toolkit-modernizing-transmission-system-operator-s/>

How to Modernize Enterprise Data and Analytics Platform (Part 4 of 4):

<https://www.datasciencecentral.com/part-4-of-4-toolkit-modernizing-transmission-system-operator-s/>

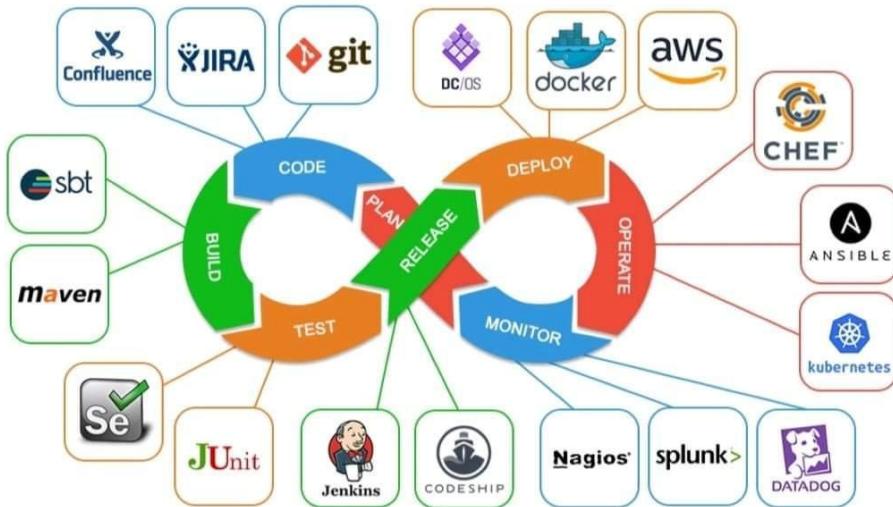
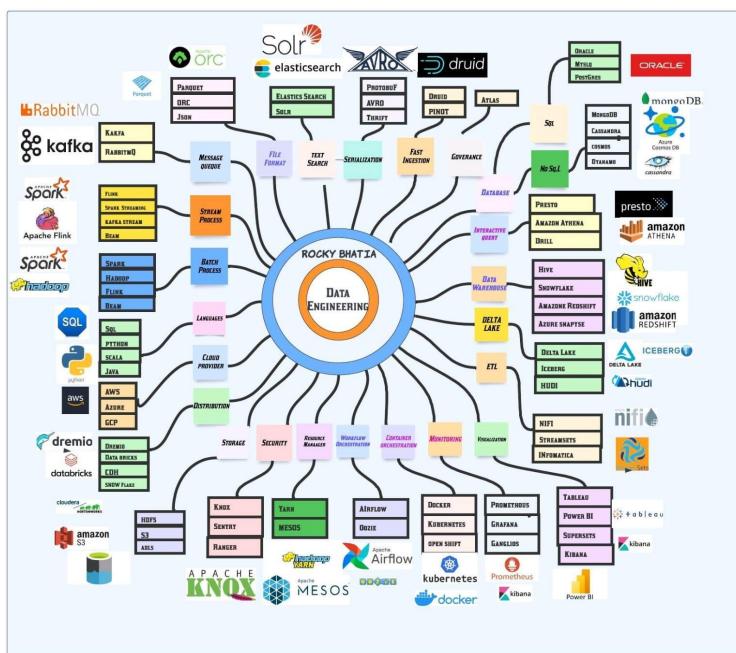


Image source: <https://testrigor.com/blog/what-is-cicd/>

MLOPS Tools



Data Engineering

VALL-E, that generates audio from text #MSFTResearch!

DALL-E generates pixels from text. Now meet its cousin, VALL-E, that generates audio from text #MSFTResearch!

VALL-E's resemblance to DALL-E v1 and Parti #GoogleAI is striking. Image and audio are both continuous signals, but they can be quantized into discrete tokens.

Read the designs of VALL-E and text2image models side by side to illustrate the similarity:

Tokenizer:

Image: VQVAE quantizes the image into a sequence of discrete tokens.

VALL-E: pre-trained discrete neural audio codec model called "EnCodec" [Défossez et al., 2022]

Architecture:

Image: conditioned on text, output discrete image codes autoregressively.

VALL-E: conditioned on text and 3-second acoustic prompt, output discrete audio codes autoregressively.

Refining generation quality:

Image: super resolution module.

VALL-E: non-autoregressive Transformer that outputs discrete codes from a cascade of late-stage quantizers.

Intuitively, the 1st level of quantizer is the most important. It captures the content and speaker identity of the audio. Later stage quantizers (level 2-8) progressively refine the signal to make it sound more natural. Each level models only the residual from the previous.

VALL-E enables stronger in-context abilities than prior works, as it can synthesize high-quality speech for unseen speakers without fine-tuning at test time. It's trained on 60K hours of English speech with over 7000 unique speakers.

Audio demo: <https://t.co/Fzpqi9VRJ9>

Note that VALL-E is similar to DALL-E v1, NOT v2.

DALL-E v2, Stable Diffusion, and Imagen rely on diffusion to regress continuous signals, instead of predicting discrete code sequences.

Paper: <https://t.co/M5k97rVoQt>

Text2speech is playing an essential role in commercial systems like Siri & Alexa. Scaling it up will have huge impact on many apps. I hope VALLE will be open-sourced soon! <https://t.co/AP3Nkv07Pn>

GitHub : 7 quality learning resources

Here are 7 programming repos you'll wish you knew sooner:

1. Coding Interview University

- Complete computer science study plan to become a software engineer.
- 242k stars

- Multi-month study plan, organized well

<https://t.co/sSkFZFjHFI>

2. Web Dev For Beginners by Microsoft

- 24 Lessons, 12 Weeks, WebDev course
- 63k stars
- Created by devs at Microsoft

<https://t.co/vwJNOkl2pp>

4. Free For Dev

- A list of SaaS, PaaS and IaaS offerings that have free tiers of interest to devops
- 64k stars
- Free tier plan details for an endless list of DevOps services

<https://t.co/P1fChn9bn9>

5. Realworld

- See how the exact same <https://t.co/TvFZg09LVS> clone is built using different frontends and backends.
- 71k stars
- Helpful for understanding other technologies in depth

<https://t.co/15hBmCFcKn>

6. Tech Interview Handbook

- Curated coding interview preparation materials for busy software engineers
- 84k stars
- Practice questions, interviewing techniques, and more

<https://t.co/2qAIDDu6Ww>

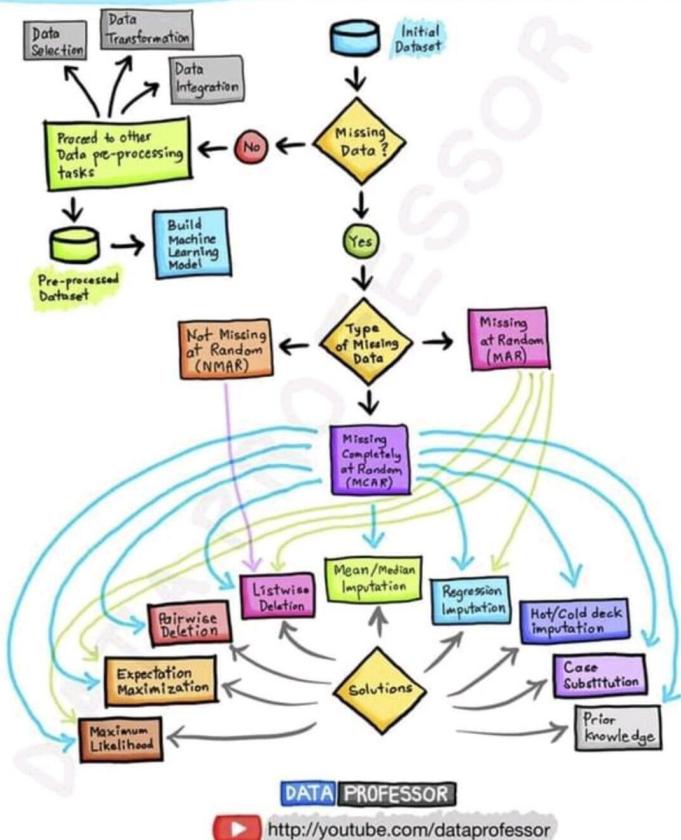
7. Free Programming Books

- Free learning resources In many languages
- 260k stars
- Good UX, organized well

<https://t.co/HZonw9MQWb>

HANDLING MISSING DATA

By: Chanin Nantasenamat



Handling Missing Data

Learn For Free

Web3 → <https://learnweb3.io/>
 JavaScript → <https://javascript.info/>
 Git → <https://lnkd.in/gcuGur3p>
 Python → <https://learnpython.org/>
 Solidity → <https://cryptozombies.io/>
 SQL <https://learnsql.com/>
 CSS → <https://web.dev/learn/css/>

HTML → <https://learn-html.org/>
 Java → <https://lnkd.in/gPMgBXKS>
 MongoDB <https://learn.mongodb.com/>
 Rust → <https://lnkd.in/gCxPXgfh>
 Tailwind CSS → <https://lnkd.in/gn7Ec2J8>

Vue JS → <https://lnkd.in/gdq46Nks>
 AWS → <https://www.aws.training/>

Top 10 Freelancing Websites

Top 10 Freelancing Websites for Software Developers:

- Upwork
- Freelancer
- Fiverr
- Toptal <https://www.toptal.com/>
- Truelancer <https://www.truelancer.com/>
- Codementor <https://www.codementor.io/>
- Guru <https://www.guru.com/>
- PeoplePerHour. <https://www.peopleperhour.com/>
- Upstack
- Gigster. <https://gigster.com/>

Which one are you using ?

Unexplored Python Libraries for Time Series Analysis

<https://towardsdatascience.com/5-unexplored-python-libraries-for-time-series-analysis-e9375962fbb2>

12 Time Series Python Libraries incl Deep Learning

https://www.linkedin.com/feed/update/urn:li:activity:7014190946372182016?utm_source=share&utm_medium=member_android

Sanskrit text extraction

<https://www.i2ocr.com/pdf-ocr-sanskrit>

How to extract Sanskrit text from PDF?

Apart from the tools mentioned in the blog above, there are many other open-source OCR tools to extract Sanskrit text from documents.

...

Here is the list of some tools you might want to check out:

I2ocr.
Convertio.
SimpleOCR.
Power Automate.
PyPDF2 (Python PDF library)
Google Drive OCR.
Google Docs OCR.

Github <https://github.com/shahrukhx01/multilingual-pdf2text>
<https://pypi.org/project/multilingual-pdf2text/>

101. Extract Text from PDF ⚡

```
[1]: !pip install pdfplumber -q

[2]: import pdfplumber as pp

with pp.open('test_data/example.pdf') as book:
    for page_no, page in enumerate(book.pages, start=1):
        print(f'{page_no = }')
        data = page.extract_text()
        print(data.strip())
        print('-'*45)
        Afiz
        @itsafiz

page_no = 1
Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do: once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, 'and what is the use of a book,'
thought Alice 'without pictures or conversation?'

So she was considering in her own mind (as well as she could,
for the hot day made her feel very sleepy and stupid), whether
the pleasure of making a daisy-chain would be worth the trouble
of getting up and picking the daisies, when suddenly a White
Rabbit with pink eyes ran close by her
-----
page_no = 2
There was nothing so VERY remarkable in that; nor did Alice
think it so VERY much out of the way to hear the Rabbit say to
itself, 'Oh dear! Oh dear! I shall be late!' (when she thought
it over afterwards, it occurred to her that she ought to have
```

Pdfplumber python library

OpenAI Introduces Point-E

OpenAI Introduces Point-E: A Machine Learning System That Can Rapidly Generate 3D Images Based On Text Prompts

Quick Read:

<https://www.marktechpost.com/2022/12/23/openai-introduces-point-e-a-machine-learning-system-that-can-rapidly-generate-3d-images-based-on-text-prompts/>

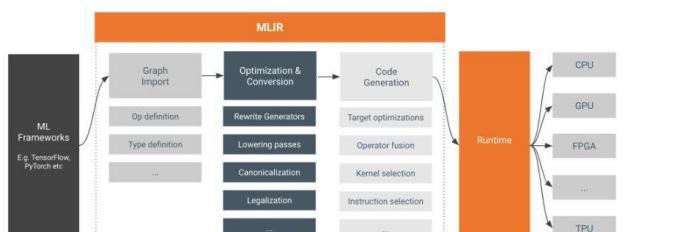
Paper: <https://arxiv.org/pdf/2212.08751.pdf>

Github: <https://github.com/openai/point-e>



MLIR Compiler Infrastructure

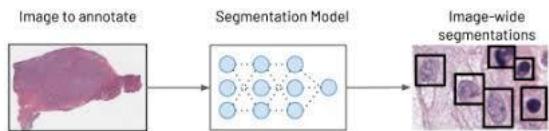
TLDR: A common graph representation and legalization framework, a common set of optimization and conversion passes and a full code generation pipeline.



MLIR Compiler Infrastructure

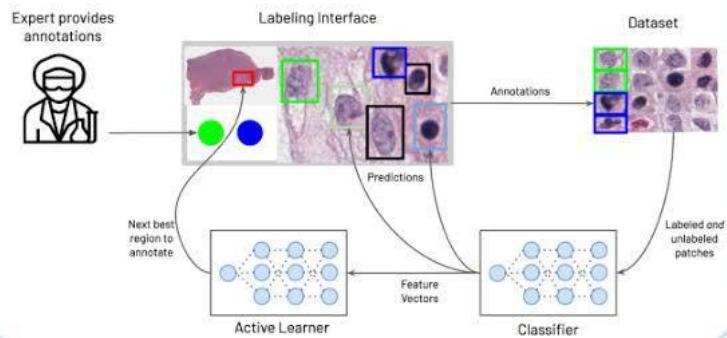
a.

Data Pre-Processing



b.

Real-time AI Augmentation



Brief list of tools and sites to help ensure a solid MLOps pipeline and reproducibility/repeatability in your computer vision (ML/DL) projects.

#mlops #computervision #artificialintelligence
#lowcode #template #frameworks #tools
#opensource

Vision MLOps Assistors

Datasets	Stock Data	EDA & Clean	Label	Augment
Kaggle datasets Robodraw datasets Codalab IEEE Dataport Google Datasets Github Dataset Repos	Pexels Pixabay Shutterstock Unsplash	OpenCV Pillow Torchvision GAN Synthesis GAN Denoising Kaggle EDA Notebooks	Label Studio Labelling CVAT LabelMe Few Shot Learning Active Learning	Torchvision Albumentation ImgAug Augmentor OpenCV
Generate	Frameworks	Models	Trackers	Hyperparams
GAN Algorithms Diffusion Models	Pytorchflow Tensorflow Mxnet Jax	OpenMMLab Huggingface PapersWithCode Framework ModelZoo Nvidia ModelZoo ImgClsMob	TensorBoard W&B MLFlow PolyAxon Neptune Comet	Hyperopt MilMachine Optuna FastAI PolyAxon RayTune
Low Code Params	Distributed Training	Validation metrics	Optimization	
ImgClsMob Monk CV Toolkit PyTorch-LB PyCaret H2O AutoML FastAI	BigDL HiveMind Horovod Deepspeed Mesh TF Ray	PaperWithCode Pytorch-Metric-Learning TorchMetrics TF Keras Metrics	DeepSparse Onnx MMDeploy TensorRT OpenVINO Tvm, LLVM Compilers FasterTransformer	
Containerization	Deployment	Interpretability		
Docker, Kubernetes Helm Docker Compose Minikube Kubeflow	Fastapi Django MMDeploy TF, Pytorch Serving Aws, Gcp, Azure Tools	Gradiance DeepLift, SHAP What If Tool Activation Adasae EL15 AIx360 Alibi Explain		

Vision MLOps Assistors

OPT-IML

OPT-IML (OPT + Instruction Meta-Learning) is a set of instruction-tuned versions of OPT, on a collection of ~2000 NLP tasks gathered from 8 NLP benchmarks, called OPT-IML Bench.

<https://github.com/facebookresearch/metaseq/tree/main/projects/OPT-IML>

Transformer Architecture

An interesting article clarifying and specifying the dimensionality and data flow of each sub component of the transformer architecture...

Into TheTransformer:

"The Transformer — a neural network architecture introduced in 2017 by researchers at Google — has proved to be state-of-the-art in the field of Natural Language Processing (NLP) and subsequently made its way into Computer Vision (CV).

Despite many resources available online explaining its architecture, I am yet to come across a resource that talks explicitly about the finer details of the data as it flows through the Transformer in the form of matrices.

This article covers the dimensions (of inputs, outputs, and weights) of all the sub-layers in the Transformer. And at the end, the total number of parameters involved in a sample Transformer model is calculated.

This article is organized as below:

- The Transformer
- Encoder
- Decoder
- The Peripheral Blocks
- Summary

<https://towardsdatascience.com/into-the-transformer-5ad892e0cee>

Understanding Bias-Variance Trade-off from a Bayesian Perspective

A comprehensive understanding of the equation $MSE = \text{bias}^2 + \text{variance}$ from a Bayesian perspective using examples and simulations

"Throughout my machine learning journey, I've come across the bias-variance trade-off several times, but I haven't been able to fully grasp it. However, after adopting a Bayesian perspective, everything started to make sense. In this article, I'll share my learnings.

First, we start with the theory and do a basic example; then we move to a harder example on Bayesian polynomial regression.

In any kind of inference problem, we want to estimate the value of an unknown parameter. This parameter is usually represented by Θ (Theta). In Bayesian inference, we assume Θ is a random variable with a prior

distribution represented by $P(\Theta)$. We start by constructing a model such that its output depends on Θ . We perform the experiment and record the values of the output. These are called observations.

Observations are random variables usually represented by X whose distribution depends on Θ . This is a conditional distribution represented by $P(X | \Theta)$.

Using $P(\Theta)$ and $P(X | \Theta)$, we can find the distribution of $P(\Theta | X)$ using Bayes Theorem. We'll discuss this in much more detail later, but for now, understand that $P(\Theta | X)$ represents the updated distribution of Θ based on the recorded observations. This is also called the posterior distribution.

From the posterior distribution, we estimate an optimal value of θ based on certain metrics. Note: θ is the realized value of Θ . Θ is a random variable whereas θ is a real value. Similarly, x is the realized value of X ."

<https://towardsdatascience.com/understanding-bias-variance-trade-off-from-a-bayesian-perspective-6c31645fcaa9>

Deep Learning Paper Implementations

A collection of simple PyTorch implementations of neural networks and related algorithms. These implementations are documented with explanations and side-by-side notes.

 <https://t.co/DbSXXEeNjJ>

Can we solely rely on LLMs' memories

Can we solely rely on LLMs' memories (eg replace search w ChatGPT)? Probably not.
Is retrieval a silver bullet? Probably not either.

Researchers analysis shows how retrieval is complementary to LLMs' parametric knowledge

 <https://t.co/UHGOmajTxV>
 <https://t.co/Miv0MHPNes>

#LLMs store a lot of factual knowledge in their parameters (parametric factual knowledge), but recent work shows that they struggle to learn less frequent facts and can often hallucinate when they don't know. How much do they memorize and what affects their memorization?

To answer these questions, researchers construct a new large open-domain QA dataset, PopQA, whose questions are grounded in Wikidata and are sampled from long-tail popularity distributions of Wikipedia to enable fine-grained analysis. We then test 10 LLMs in a zero/few-shot manner

Researchers found that there are strong correlations between subject entity popularity and accuracy, indicating that LLMs memorize well popular factual knowledge while it does not memorize less popular ones.

Surprisingly, in long-tail distributions, scaling LLMs may not be as helpful as researchers believed: GPT-3 003 performs nearly as poorly as GPT-Neo 2B on less popular entities 
Prior knowledge learning analysis often uses NQ/TriviaQA  may inflate the effectiveness of scaling.

Researchers show that augmenting LMs with non-parametric memories (retrieved text chunks) largely helps: GPT-Neo 1.3B assisted by retrieved context outperforms vanilla GPT-3 003! Even for GPT-3, retrieval gives up to 10% accuracy gains. Why are they so effective?

Researchers found that retrieval-augmented LMs (red & green lines in attached image) are particularly helpful for questions about less popular entities, where LMs suffer. On the contrary, larger models (eg GPT-3) even outperform retrieval-augmented models in well-known facts, due to retrieval errors.

In summary, LLMs indeed memorize a lot now, but they are still not good enough to completely replace non-parametric memories, esp domains with long-tail distributions. Can we get the best of both worlds? Researchers introduce a simple-yet-effective Adaptive Retrieval

Adaptive Retrieval decides when *not* to retrieve based on the subject popularity & relationship type. This approach not only gives performance improvements (up to 5%) but also largely reduces the inference time latency & API costs (e.g., halves GPT-3 API costs!).

Read below paper: More interesting results & discussions!

 <https://t.co/HITaNm4he6>

CMU Advanced NLP Lectures playlist

<https://youtube.com/playlist?list=PL8PYTP1V4I8D0UkqW2fEhgLrnIDW9QK7z>

Website :<https://phontron.com/class/anlp2022/>

References site :<https://AlphaSignal.ai>

<https://www.linkedin.com/newsletters/new-machine-learning-articles-6797568930018861057>

30m global terrain dataset

FABDEM - a new 30m global terrain dataset

Using machine learning, researchers remove building and tree artefacts from Copernicus GLO-30m DEM, to create the most accurate complete map of the Earth's terrain ever built.

Paper <https://t.co/gN8mbHbLmT>

Data <https://t.co/2WW9tlbPbu>

Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles

<https://onlinelibrary.wiley.com/doi/epdf/10.1002/nag.3293>

Microsoft is offering online courses for FREE

Microsoft is offering online courses for FREE in the below areas

Web development, IoT, AI, Data Science and Machine Learning

1. Web Development for Beginners

<https://t.co/g8QT9QpFjW>

2. IoT for Beginners

<https://t.co/YIFqFex8dc>

3. Data Science for Beginners

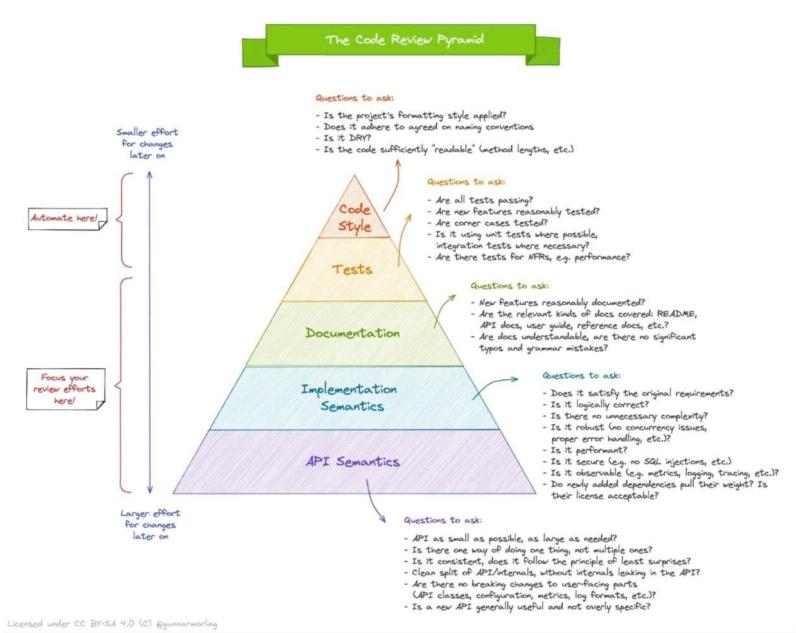
<https://t.co/3VnwY3khsH>

4. Artificial Intelligence for Beginners

<https://t.co/2pqgfSZPMn>

5. Machine Learning for Beginners

<https://t.co/8BJQd0NVeN>



The Code Review Pyramid

30 Cybersecurity search engines for Researchers:

1. Dehashed—View leaked credentials.
2. SecurityTrails—Extensive DNS data.
3. DorkSearch—Really fast Google dorking.
4. ExploitDB—Archive of various exploits.
5. ZoomEye—Gather information about targets.
6. Pulsedive—Search for threat intelligence.
7. GrayHatWarfare—Search public S3 buckets.
8. PolySwarm—Scan files and URLs for threats.
9. Fofa—Search for various threat intelligence.
10. LeakIX—Search publicly indexed information.
11. DNSDumpster—Search for DNS records quickly.
12. FullHunt—Search and discovery attack surfaces.

13. AlienVault—Extensive threat intelligence feed.
14. ONYPHE—Collects cyber-threat intelligence data.
15. Grep App—Search across a half million git repos.
16. URL Scan—Free service to scan and analyse websites.
17. Vulners—Search vulnerabilities in a large database.
18. WayBackMachine—View content from deleted websites.
19. Shodan—Search for devices connected to the internet.
20. Netlas—Search and monitor internet connected assets.
21. CRT sh—Search for certs that have been logged by CT.
22. Wigle—Database of wireless networks, with statistics.
23. PublicWWW—Marketing and affiliate marketing research.
24. Binary Edge—Scans the internet for threat intelligence.
25. GreyNoise—Search for devices connected to the internet.
26. Hunter—Search for email addresses belonging to a website.
27. Censys—Assessing attack surface for internet connected devices.
28. IntelligenceX—Search Tor, I2P, data leaks, domains, and emails.
29. Packet Storm Security—Browse latest vulnerabilities and exploits.
30. SearchCode—Search 75 billion lines of code from 40 million projects.

Intro Article on Web3.0

This article by Nader Dabit on freecodecamp is a good one to start.

<https://t.co/Hd1oUUnILS>

Blockchain Fundamentals

Basic concepts about the blockchain.

- Web1, Web2, Web3
- What is a Blockchain
- Distributed Ledger
- Crypto basics
- Smart Contracts
- Wallets
- Block Explorers
- Blockchain Oracles
- NFT Intro
- DAO Intro
- Dapp Intro

Fundamentals are fundamental

- Choose a Blockchain
- Each blockchain is an ecosystem, with pros/cons
- Choose a Blockchain to understand concepts and apply them, like deciding to use Node.js to build some API.
- Chose Ethereum for many reasons, the main one is that there is more material about it online.
- Ethereum Concepts (basics)
- Learn the basics of the Blockchain.

For Ethereum:

- Accounts
- Transactions

- Blocks
- EVM
- Gas
- Nodes & Clients
- Networks, Node Providers
- Consensus algorithms
- Beacon Chain
- Side Chains
- The Merge

Ethereum Concepts (Advanced)

If you want to dive a bit deeper, here are some articles and concepts you can look for:

- Ethereum Whitepaper
- Ethereum Trilemma
- EndGame (article by Vitalik)
- Ethereum Yellowpaper

These are optional.

Choose a language

- Solidity (good for JavaScript/C++/Python Developers)
- Vyper (good for Python Developers)
- Rust (choose this if you chose Solana as blockchain)

Choose Solidity because it's the most popular one.

Solidity (Blazing fast intro)

- It's an OO, high-level language for implementing Smart Contracts.
- It's statically typed.
- It's a curly-bracket language designed to target the Ethereum Virtual Machine.
- It supports inheritance, libraries, and complex user-defined types.

Solidity Videos

Some good YouTube channels to know more about Solidity:

- Patrick Collins Javascript Version
- Patrick Collins Python version
- Gregory (Dapp University)
- Smart Contract Programmer

Solidity support

- Remix IDE (very good to get started without setting up a local environment)
- Solidity Docs
- Solidity Cheatsheet

- Cryptozombies (a nice online tutorial to get started)

Use a framework

Once you get the basics, you can dive deep using these tools and working in a local environment:

- Metamask
- Vs Code Solidity Extension
- Truffle
- Hardhat
- Brownie
- Foundry

Connect Dapp to Frontend

Are you a front-end developer? GOOD!

For once, you don't have to learn a new framework, just use a library to interact with the blockchain!

You can use any frontend Framework (React, Vue, Angular, Svelte...)

Use:

- Web3.js
- Ether.js

Decentralized storage

Here are some examples of storage you can use to store your data on the blockchain (something similar to databases).

- IPFS
- Arweave
- FileCoin

They are a p2p network of user-operators who hold a portion of the overall data.

Mid-level tools

Most of them are not complicated. To be honest, they simplify the process, but it's better to discover them after understanding the basics.

- Testing Tools: Waffle, Ganache
- OpenZeppelin
- WalletConnect
- Chainlink
- TheGraph

SDKs

After getting an idea of what a dapp is and making some examples, you can decide you want some support and don't write everything on your own.

Here are 3 good ones you can check:

- Alchemy
- thirdweb

- Moralis

Build projects

Build projects! I leave a couple of places where you can get inspiration for building projects, but there are way more on YouTube

- Moralis
- Alchemy

Build in Public!

Not a Web3-specific thing, but if you build in public, here are some benefits:

- you get a better understanding of the concepts
- you get an audience
- you connect with people with the same interests

<https://t.co/tj0HjpHxrO>

SOAP and REST APIs

SOAP is like putting a letter in a special envelope. This envelope has clear instructions and rules for how to write the address and seal it. Inside the envelope, you place your message (usually written in XML). By using this structured and rule-bound envelope, your message can be sent safely and with assurance, just like sending a letter in a sealed and addressed physical envelope.

REST is like using a postcard because it's a lightweight and simple way to communicate, over the internet, with limited space, clear addressing, and predefined methods for interacting with resources. It's particularly well-suited for web and mobile applications that don't require the complexity of more formal protocols like SOAP.

SOAP (Simple Object Access Protocol):

- Protocol with strict rules.
 - Message format is typically XML.
 - Strong security features.
 - Supports stateful communication.
 - Standardized error handling.
 - Robust tooling in many programming languages.
- Various message bindings.

REST (Representational State Transfer):

- Architectural style, not a protocol.
- Message format is flexible, often JSON or XML.
- Relies on transport-level security (e.g., HTTPS).
- Stateless communication enhances scalability.
- Simpler and lighter weight.
- Tooling is more straightforward, leveraging HTTP.
- Well-suited for web and mobile applications.

<https://towardsdatascience.com/deploying-large-language-models-with-huggingface-tgi-981747c669e3>

<https://link.medium.com/dK0LHLbeWAb>

<https://github.com/cmhungsteve/Awesome-Transformer-Attention>

<https://github.com/ahmedbahaaeldin/From-0-to-Research-Scientist-resources-guide>

<https://github.com/padok-team/github-actions-tutorial>

https://www.linkedin.com/posts/brijpandeyji_api-apitesting-softwareengineering-activity-7078408639631245312-DrF4?utm_source=share&utm_medium=member_android

<https://link.medium.com/VMBIM7qLUAb>

<https://github.com/bodywork-ml/bodywork-core>

<https://link.medium.com/6pZYA5EKUAb>

<https://github.com/mlabonne/llm-course>

PyRCA

<https://github.com/salesforce/PyRCA>

<https://github.com/Eventual-Inc/Daft>

CS 329S: Machine Learning Systems Design

Get the whole "CS 329S: Machine Learning Systems Design" from @Stanford ~~100~~ free!

Learn about how to define the software architecture, infrastructure, algorithms, and data to meet your specific requirements.

<https://t.co/uFv9NDJXrF>

#MachineLearning #DataScience #AI

<https://github.com/Azure/Azure-Sentinel-Notebooks/blob/master/machine-learning-notebooks/Guided%20Hunting%20-%20Anomaly%20detection%20with%20Isolation%20Forest%20on%20Windows%20Logon%20data%20For%20Data%20Scientist%20.ipynb>

<https://github.com/primeqa>

Microsoft Machine Learning Studio (classic)

<https://studio.azureml.net/>

Top 5 Notable Humanoid Robots Made in India

<https://www.analyticsinsight.net/top-5-notable-humanoid-robots-made-in-india/>

TransUNet — Revolutionize Traditional Image Segmentation

<https://link.medium.com/qGtWnxom9xb>

Steamship GPT3 based Chatbot

<https://www.steamship.com/workspaces/gpt-4/plugins/gpt-4>

Bootstrapping Labels with GPT-4

<https://link.medium.com/vbJA7JhWDAb>

10 Most Used Machine Learning Algorithms In Python With Code

<https://www.theinsaneapp.com/2021/11/machine-learning-algorithms-for-beginners.html>

Machine Learning resources and tools for kids (#AI)

<https://t.co/oj8Sri6sBa>

#0-DS-Resources-Links

<https://github.com/labmlai/labml>

Customer Reviews NLP Spacy Analysis and ML/AI Demand Forecasting of the Steam PC Video Game Service – Digital High Science

<https://newdigitals.org/2023/06/17/customer-reviews-nlp-spacy-analysis-and-ml-ai-demand-forecasting-of-the-steam-pc-video-game-service/>

Neo4j and Google Cloud Join Forces to Empower Generative AI with Knowledge Graphs - Multiplatform AI

<https://multiplatform.ai/neo4j-and-google-cloud-join-forces-to-empower-generative-ai-with-knowledge-graphs/>

Introducing DistMult and ComplEx for PyTorch Geometric

<https://link.medium.com/J5G0lkHeMAb>

Integrate LLM workflows with Knowledge Graph using Neo4j and APOC

<https://link.medium.com/8hFfP1NiMAb>

https://www.linkedin.com/posts/sonalsingh96_data-sql-python-activity-7042114696023326720-V2Pt?utm_source=share&utm_medium=member_android

#0-DS-Resources-Links

Here is a List of 190 #python projects with source code, solved and explained that will help every beginner in improving Python programming skills.

#python #pythonprogramming

<https://t.co/zaayCVZjOp>

qiskit

https://github.com/theerfan/Q/tree/master/QML%20Course%2Fqiskit_version

Understanding Deep Learning

<https://udlbook.github.io/udlbook/>

CMU Advanced NLP 2022

CMU Advanced NLP 2022

YouTube playlist

<https://www.youtube.com/playlist?list=PL8PYTP1V4I8D0UkqW2fEhgLrnIDW9QK7z>

Synthetic Data Explainable AI

New Course: Synthetic Data Explainable AI

<https://www.linkedin.com/pulse/new-course-synthetic-data-explainable-ai-vincent-granville>

Roadmap to learn Web3

https://youtu.be/me0Qxa_21Yk

Aeroponic Potato Farming Technique And Its Advantages

<https://hindi.thebetterindia.com/102306/aeroponic-potato-farming-technique-advantages/>

How to Grow Potatoes in The Air With a Technique That Increases The Yield by 10x

<https://www.thebetterindia.com/281269/how-to-grow-potatoes-using-aeroponics-at-home-yield/>

GRC

https://corporater.com/blog/what-is-grc-governance-risk-and-compliance/?cmc=lk_grc-blog2&utm_source=lkads&utm_medium=grc-blog

Beyond ChatGPT: The Future Of AI At Work

<https://www.forbes.com/sites/karlmoore/2022/12/14/leverage-generative-ai-workplace-tools-with-semantic-search/>

Data generator

<https://mitblog.com/3HKnBS2>

Try Mito :Edit a spreadsheet, generate Python code
Mito makes Python data analysis 10x easier.

Join thousands using Mito in Jupyter.

<https://www.trymito.io>

How To Create Animated Videos For YouTube Using 'Text to Video' AI Tools

<https://youtu.be/YZHrKgtNbA>

Essence of linear algebra

https://youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Video lectures, Optimization for AI by Se-Young Yun

<https://t.co/mbfL0Bhc3e>

Video lectures, Statistical Learning by Se-Young Yun

<https://t.co/XwkPlgkaPi>

Video lectures, Deep Learning by Se-Young Yun

<https://t.co/S1sgVdQ0GI>

Speech Synthesis, Recognition, and More With SpeechT5

<https://huggingface.co/blog/speecht5#text-to-speech>

A New Deep Reinforcement Learning (DRL) Framework can React to Attackers in a Simulated Environment and Block 95% of Cyberattacks Before They Escalate

<https://www.marktechpost.com/2023/02/25/a-new-deep-reinforcement-learning-drl-framework-can-react-to-attackers-in-a-simulated-environment-and-block-95-of-cyberattacks-before-they-escalate/>

fast.ai - Making neural nets uncool again

<https://www.fast.ai/>

Awesome Quantum Machine Learning

<https://github.com/krishnakumarsekar/awesome-quantum-machine-learning>

InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval

<https://arxiv.org/abs/2301.01820>

Here's a roadmap to Master Python!

1 freeCodeCamp

4 hours Python bootcamp!!

What you'll learn:

- Installing Python
- Setting up an IDE
- Basics Syntax
- Variables & Datatypes
- Looping in Python
- Exception handling
- Modules & pip
- Mini hands-on projects 🔥

YouTube :

<https://lnkd.in/gm3Pz5ju>

2 CS50p: Harvard University

Beautiful explanations and great projects.
It's a complete package.

Highly recommended!!

Check this out 

<https://t.co/ucm0JbQuEb>

3 Corey Schafer

Arguably the best python channel on YouTube 

Check this out 

<https://t.co/SMrNBSLkeT>

4 Project based learning

You don't learn unless you get your hands dirty. 

This Github repo contains a curated list of great python projects across many domains and areas of interest.

Check this out 

<https://t.co/8tsodWVq29>

Read paper :Constitutional AI: Harmlessness from AI Feedback

<https://arxiv.org/abs/2212.08073>

Researchers describe how they have used Constitutional AI to train better and more harmless AI assistants without any human feedback labels for harms. This approach leads to models that are safer and also more helpful.

Constitutional AI has five motivations:

- (1) Make the goals and objectives of AI systems more transparent,
- (2) Make AI decision making more transparent,
- (3) Use a much smaller quantity of high quality human supervision when training AIs,
- (4) Fully automate red-teaming and train much more robust AI systems,
- (5) Explore “Scaling Supervision” by allowing AI systems to help humans to ensure that other AI systems remain safe.

Future Directions:

In prior work researchers have focused on training AI assistants to helpful, harmless, and honest, but otherwise researchers have allowed their behavior to be determined by generalization patterns from pretraining

that are not under our direct control.

However, the constitutional methods researchers have discussed here are very general, and in principle might be applied to steer language models in a variety of ways. For example, researchers expect they could use these methods to change the model's writing style, tone, or personality, or alter its responses to specific categories of questions (e.g. to train an AI that heavily caveats certain categories of advice, or that adopts a specific persona). The constitutional approach should thus make it much easier to study how different AI behaviors tend to generalize

and interfere, since by obviating human feedback, our methods lower the barrier to experimentation. For

example, it should be possible to generate feedback labels along dozens of behavioral axes, and then study how preference models trained from these labels are correlated or anti-correlated. This is important for AI safety, since the generalization patterns imbued by pretraining are currently something of a black box whose correlations may have unforeseen consequences.

Another remaining issue, and a major motivation for this work, is robustness—that is, can we make models essentially immune to red-team attacks? We hope that by making helpfulness and harmlessness more compatible, researchers will be able to significantly scale-up (automated) red teaming in order to improve robustness.

Furthermore, researchers should be able to perform iterated ‘online’ training with AI supervision, where researchers update the preference model with new AI feedback in order to keep it on the same distribution as the policy produces. Researchers saw that this was valuable with human feedback, and by using AI feedback we can fully automate the process. Robustness was also another motivation for using chain-of-thought reasoning in this work – Researchers would eventually like AI systems to reason through the hidden risks of certain behaviors, in order to mitigate increasingly subtle and implicit harms.

Index of /lecture_slides

https://cs330.stanford.edu/lecture_slides/

Complete PYTHON Cheat Sheet with learning notes
Get this Notes : <https://bit.ly/3M9nlho>

Computer Vision:
Building your own Object Detector from scratch with Tensorflow
<https://lnkd.in/gMDawkjx>

MLU-Explain

<https://mlu-explain.github.io/>

Learn Data Structure using these 46 Best Resources

<https://t.co/mxN7WXP34J>

Transformer models: an introduction and catalog

Transformer models: an introduction and catalog—2023 Edition - AI, software, tech, and people, not in that order... by X

<https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>

Leta transcripts – Dr Alan D. Thompson – Life Architect

<https://lifearchitect.ai/leta-transcripts/>

How to produce factually accurate automatic text summaries - Amazon Science

<https://www.amazon.science/blog/how-to-produce-factually-accurate-automatic-text-summaries>

Understanding Large Language Models -- A Transformative Reading List

<https://sebastianraschka.com/blog/2023/lm-reading-list.html>

Stanford CRFM Introduces PubMedGPT 2.7B

<https://hai.stanford.edu/news/stanford-crfm-introduces-pubmedgpt-27b>

ChatGPT competitors: Amazon jumps into fray with generative AI better than GPT-3.5

<https://www.moneycontrol.com/news/technology/chatgpt-competitors-amazon-jumps-into-fray-with-generative-ai-better-than-gpt-3-5-10063651.html>

Multimodal Chain-of-Thought Reasoning in Language Models

<https://arxiv.org/abs/2302.00923>

8 Types of Google Cloud Analytics: How to Choose?

<https://bluexp.netapp.com/blog/gcp-cvo-blg-8-types-of-google-cloud-analytics-how-to-choose>

RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning

Machine Learning Blog | ML@CMU | Carnegie Mellon University

<https://blog.ml.cmu.edu/2023/02/24/rlprompt-optimizing-discrete-text-prompts-with-reinforcement-learning/>

Researchers at Stanford Introduce ControlNet: A Neural Network Structure to Control Pre-Trained Large Diffusion Models to Support Additional Input Conditions - MarkTechPost

<https://www.marktechpost.com/2023/02/23/researchers-at-stanford-introduce-controlnet-a-neural-network-structure-to-control-pre-trained-large-diffusion-models-to-support-additional-input-conditions/>

nlpconnect/vit-gpt2-image-captioning · Hugging Face

<https://huggingface.co/nlpconnect/vit-gpt2-image-captioning>

lmx sentiment demo.ipynb - Colaboratory

<https://colab.research.google.com/drive/1SXrq-YGffg6M725hgKXY1lqxTMsa1wLI?usp=sharing>

Large Language Models and Where to Use Them: Part 1

<https://txt.cohere.ai/llm-use-cases/>

A deep reinforcement learning model that allows AI agents to track odor plumes

<https://techxplore.com/news/2023-02-deep-ai-agents-track-odor.html>

Vertex AI Foundations for secure and compliant ML/AI deployment | Google Cloud Blog

<https://cloud.google.com/blog/topics/developers-practitioners/vertex-ai-foundations-secure-and-compliant-ml-ai-deployment>

Using PyGWalker to Enhance Your Jupyter Notebook EDA Experience

<https://towardsdatascience.com/use-pygwalker-to-enhance-your-jupyter-notebook-eda-experience-d7a5ee358be9>

Meet The Sketchy Cousin of Stable Diffusion, ‘Scribble Diffusion’

<https://analyticsindiamag.com/meet-the-sketchy-cousin-of-stable-diffusion-scribble-diffusion/>

Algebra and Trigonometry 2e - OpenStax

<https://openstax.org/details/books/algebra-and-trigonometry-2e>

Calculus Volume 1 - OpenStax

<https://openstax.org/details/books/calculus-volume-1>

Introductory Business Statistics - OpenStax

<https://openstax.org/details/books/introductory-business-statistics>

Generative AI: What are the potential applications and ethical concerns?

<https://www.insightsonindia.com/2023/02/06/generative-ai-what-are-the-potential-applications-and-ethical-concerns-2/>

Zero-shot image-to-text generation with BLIP-2

https://huggingface.co/blog/blip-2?utm_source=substack&utm_medium=email

50+ ML Papers Explained

<https://github.com/dair-ai/ML-Papers-Explained>

Awesome Community Detection Research Papers

<https://github.com/benedekrozemberczki/awesome-community-detection>

[Pynecone.io](#)

[Pyweb.io](#)

[Anvil.works](#)

[Atrilabs.com](#)

Top 15 Use Cases of Object Detection in Computer Vision 😊

<https://muhammadrizwanmunawar.com/blog/f/top-15-use-cases-of-object-detection-in-computer-vision%F0%9F%A4%94>

Understanding Vision Transformers (ViTs): Hidden properties, insights, and robustness of their representations | AI Summer

<https://theaisummer.com/vit-properties/>

Evaluating explainability for graph neural networks | Scientific Data

<https://www.nature.com/articles/s41597-023-01974-x>

Explicit Planning Helps Language Models in Logical Reasoning

<https://arxiv.org/abs/2303.15714>

How AI Will Transform Project Management

<https://hbr.org/2023/02/how-ai-will-transform-project-management>

CI/CD for Machine Learning (GitOps) - W&B Courses

<https://www.wandb.courses/courses/take/ci-cd-for-machine-learning/lessons/42949551-your-instructor>

Use Open AI (ChatGPT) On Your Own Large Data!

<https://youtu.be/eNKu307k59g>

How to Create a MySQL DB on AWS RDS and Connect with Python: The Ultimate Step-by-Step Guide

<https://blog.futuresmart.ai/how-to-create-a-mysql-db-on-aws-rds-and-connect-with-python-the-ultimate-step-by-step-guide>

DTensor Concepts | TensorFlow Core

https://www.tensorflow.org/guide/dtensor_overview

Download C3 AI Center of Excellence Data Sheet

<https://c3.ai/resources/data-sheet/download-c3-ai-center-of-excellence-data-sheet/>

Instruction Tuning with GPT-4

<https://instruction-tuning-with-gpt-4.github.io/>

Unlocking the Potential of ChatGPT: A Comprehensive Exploration of its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing

<https://arxiv.org/abs/2304.02017>

Inner (Dot) product of two Vectors. Applications in Machine Learning

<https://datahacker.rs/dot-product-inner-product/>

Task-driven Autonomous Agent Utilizing GPT-4, Pinecone, and LangChain for Diverse Applications

<https://yoheinakajima.com/task-driven-autonomous-agent-utilizing-gpt-4-pinecone-and-langchain-for-diverse-applications/>

How to Use Large Language Models (LLM) in Your Own Domains

<https://towardsdatascience.com/how-to-use-large-language-models-llm-in-your-own-domains-b4dff2d08464>

chatGPT Telegram

<https://github.com/altryne/chatGPT-telegram-bot>

https://www.linkedin.com/posts/mridul-mehndiratta_phd-artificialintelligence-research-activity-7051519414571020291-wCf/?utm_source=share&utm_medium=member_android

https://www.linkedin.com/posts/ravena-o-04b39b68_i-still-think-you-can-learn-a-lot-from-these-activity-7051242278266241024-fEeN?utm_source=share&utm_medium=member_android

Open-source LLM is free and available for anyone to access, and developers and researchers are free to use, improve, or modify the model.

Proprietary LLM is owned by a company and may include a license that restricts how the model can be used.

Generally, proprietary LLMs have far larger parameter sizes than open-source models.

But this is not always the case, as evidenced by Falcon 180B, the largest open-access model, with 180B parameters, surpassing most known LLMs.

The size of parameters in an LLM is a measure of its complexity.

In general, LLMs with more parameters tend to perform better on a variety of tasks.

💡 Open-source LLMs have their own ecosystem, with unique advantages.

Open-source LLMs offer **greater transparency**, allowing us to gain better insights into the model's workings, architecture, and the training data used.

Another advantage of open-source LLMs is the ability to benefit from **community contributions**. Researchers and developers can experiment and utilize contributions from individuals with diverse perspectives.

🤖 Which open-source LLMs have you tried? What is your favorite one?

Open Source vs Proprietary LLM

GPT4Video: A Unified Multimodal Large Language Model

GPT4Video: A Unified Multimodal Large Language Model for instruction-Followed Understanding and Safety-Aware Generation

abs: arxiv.org/abs/2311.16511
repo: <https://gpt4video.github.io/>

Explore Notus-7B released model.

Explore Notus-7B released model.

<https://huggingface.co/argilla/notus-7b-v1>

Notus 7B v1 is a DPO fine-tuned version of Zephyr 7B Beta SFT fine-tuned on UltraFeedback, but using the average of the different attributes to binarize the data, instead of the critique score; so that the chosen response is based on the average rather than on the critique score. After the DPO fine-tuning for intent alignment we surpass Zephyr 7B Beta in both AlpacaEval and LM Eval Harness, while almost on par for MT-Bench. All the training code and configuration has been adapted / ported from huggingface/alignment-handbook and is available at argilla-io/notus.

<https://huggingface.co/blog/alvarobartt/notus-7b-v1>

Check Mistral-7B-v0.1

<https://huggingface.co/mistralai/Mistral-7B-v0.1>

#TsinghuaNLP released the UltraChat dataset, a large-scale, multi-round dialogue dataset.

<https://github.com/thunlp/UltraChat>

#OpenBMB released the UltraFeedback dataset, a large-scale, fine-grained, diverse preference dataset for RLHF and DPO

A large-scale, fine-grained, diverse preference dataset (and models).

<https://github.com/OpenBMB/UltraFeedback>

#HuggingFace H4 team fine-tuned Zephyr using UltraChat (supervised fine tuning) and UltraFeedback (DPO for alignment). They released the full recipe, parameters, and code to make it reproducible in the Alignment Handbook

Robust recipes for to align language models with human and AI preferences

<https://github.com/huggingface/alignment-handbook>

✨ Argilla: the open-source feedback platform for LLMs

<https://github.com/argilla-io/argilla>

How to Effectively use Docker!

How to Effectively use Docker!

15 tips and tricks for effectively using Docker 🤝🐧🐳

1/ 🔗 Multi-Stage Builds:

Optimize your Dockerfiles by using multi-stage builds. This reduces image size by including only necessary dependencies in the final image. #DockerTips #DevOps

2/ 🚀 Buildkit for Faster Builds:

Enable BuildKit to speed up your builds. Set the DOCKER_BUILDKIT=1 environment variable or use --progress=plain for cleaner build output. #BuildKit #DockerTricks

3/ 🌐 Network Mode: Host:

Improve container network performance by using host mode. Docker containers share the network namespace with the host, reducing networking overhead. Use with caution for security implications. #DockerNetworking #PerformanceBoost

4/ Seccomp Profiles:

Enhance container security with Seccomp profiles. Restrict system calls and actions within the container, limiting potential attack surfaces. #ContainerSecurity #DockerSecurity

5/ Docker Compose Overrides:

Simplify configuration management by using Docker Compose overrides. Separate development and production configurations for seamless environment-specific setups. #ComposeOverrides #DevOps

6/ Custom Bridge Networks:

Create custom bridge networks to isolate containers. This enhances security and allows better control over communication between containers. #DockerNetworking #ContainerIsolation

7/ IPv6 Support:

Embrace IPv6 by enabling it in Docker. Use the --ipv6 flag to make your containers accessible over IPv6 networks. Future-proof your applications! #IPv6 #DockerNetworking

8/ Docker Contexts:

Manage multiple Docker environments effortlessly with Docker Contexts. Switch between local and remote Docker hosts without changing context manually. #DockerContexts #DevOps

9/ Docker Build Cache:

Leverage the build cache intelligently. Use explicit COPY commands and order them strategically to maximize cache utilization and minimize build times. #BuildOptimization #DockerCache

10/ Docker Stats:

Monitor container resource usage with "docker stats." Understand CPU, memory, and network metrics to optimize performance and identify potential bottlenecks. #ContainerMonitoring #DockerStats

11/ Restart Policies:

Fine-tune container restart behavior using restart policies. Set options like "unless-stopped" or define custom backoff strategies to handle failures gracefully. #RestartPolicies #DevOps

12/ Dockerfile Healthchecks:

Improve container health monitoring by adding HEALTHCHECK instructions in your Dockerfile. Define custom healthcheck commands to ensure robust application health. #DockerHealthchecks #DevOps

13/ Container Labels:

Organize and categorize containers efficiently using labels. Attach metadata to containers, making it easier to manage, monitor, and filter them. #ContainerLabels #Metadata

14/ Docker Content Trust:

Enable Docker Content Trust to sign and verify image content. Ensure the integrity and authenticity of your images, enhancing security in your containerized environment. #DCT #ContainerSecurity

15/ Dive for Image Analysis:

Use "dive" to analyze Docker images layer by layer. Understand image size and layer contents, helping you optimize and reduce image bloat. #DockerImageAnalysis #Optimization

Feel free to explore these advanced Docker tips for a more efficient and secure containerized workflow! 🚀💡 #Docker #DevOps #Containerization

Data Science, Machine Learning you can find multiple data distributions

In Data Science, Machine Learning you can find multiple data distributions...

But where are they typically found? 🤔

1 Normal Distribution:

Often found in natural and social phenomena where many factors contribute to an outcome. Examples include heights of adults in a population, test scores, measurement errors, and blood pressure readings.

Normal Distribution (Bell Curve):

Characterized by a symmetric bell-shaped curve, where most of the data points cluster around the mean, with fewer and fewer appearing as you move away from the mean.

2 Uniform Distribution:

This appears when every outcome in a range is equally likely. Examples include rolling a fair die (each number has an equal chance of appearing) and selecting a random number within a fixed range.

Each value within a certain range has an equal probability of occurring. The distribution is flat, with no peaks

3 Binomial Distribution:

Used when you're dealing with a fixed number of trials or experiments, each of which has only two possible outcomes (success or failure), like flipping a coin a set number of times, or the number of defective items in a batch.

Describes the number of successes in a fixed number of trials, with each trial having the same probability of success. It is characterized by a peak at the most probable number of successes.

4 Poisson Distribution:

Common in scenarios where you're counting the number of times an event happens over a specific interval of time or space. Examples include the number of phone calls received by a call centre in an hour or the probability of taxi frequency.

Used for count-based data, like the number of events happening in a fixed interval of time or space. It is characterized by a peak at lower values, with the frequency of values decreasing as they increase.

5 Exponential Distribution:

Describes the time between events in a process where events occur continuously and independently at a constant average rate.

6 Skewed Distribution (Left or Right):

In a skewed distribution, the tail of the distribution is longer on one side. In a right-skewed distribution, the tail is longer on the right, while in a left-skewed distribution, it is longer on the left.

7 Bimodal/Multimodal Distribution:

A distribution with two or more peaks. These peaks may vary in height and spread.

8 Log-Normal Distribution:

This distribution is applicable when the logarithm of the variable is normally distributed. The distribution is skewed to the right.

These distributions are fundamental in statistics and data analysis, as they provide insights into the nature of the data and inform appropriate analytical strategies.

Each distribution offers insights into the underlying processes of the data and is useful for different kinds of statistical analysis and prediction.

12 Tools for Writers

1. Grammarly

To make your writing
Error free & Effective

7. Answerthepublic

To generate content ideas

2. Hemmingway Editor

To get a Readability score

8. BuzzSumo

For content research

3. Google Doc

To write and store on Drive

9. Notion

To makes notes

4. Yoast SEO

To optimize your content
for search engine

10. QuillBot

AI based paraphrasing tool

5. Daily Page

To build a writing habit

11. Copyscape

Plagiarism checker

6. CoSchedule

To write better headlines

12. Ginger Software

Context based corrections



12 Tools for Writers

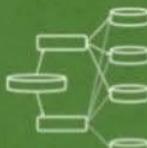
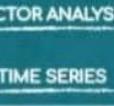
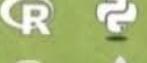
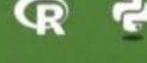
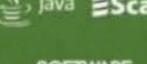
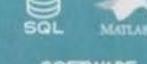
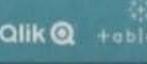
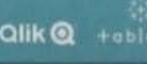
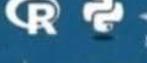
DATA STORYTELLING: A VISUAL GUIDE TO CHART SELECTION



Created by Gina Acosta
Join my newsletter joinhorizon.ai and get more free resources delivered right to your inbox!

Chart	Types of Chart	When to use it	Example of Use Cases
	Bar Chart	Compare quantities across Categories	Compare sales of different products.
	Line Chart	Show Trends over time	Display the growth of website traffic over a year.
	Pie Chart	Highlight proportions and percentages	Illustrate the breakdown of expenses in a budget.
	Scatter Plot	Represent relationships between variables	Identify correlations between marketing spend and ROI.
	Histogram	Visualize the distribution of data	Show the age distribution of survey respondents.
	Radar Chart	Compare multiple categories across dimensions	Evaluate the performance of a product in various areas.
	Map	Visualize geospatial data	Display regional sales performance on a map.
	Heatmap	Visualize data density and patterns especially in large data set	Identifying hotspots of customer activity in a shopping mall.
	Bubble Chart	Represent three-dimensional data	Compare revenue, cost, and profit in three dimensions.
	Donut Chart	Emphasize specific parts within a whole	Show the distribution of marketing expenditures.

[Data Storytelling: A Visual Guide to Chart Selection](#)

Data Science				
	Data	Data Science	Machine Learning	
WHEN it is applied	At the beginning of your analysis	After the data has been gathered & organized	After BI reports have been created and discussed	
PAST	NOW	NOW	FUTURE	
WHY you need it	<p>data-driven decisions require well-organized and relevant raw data stored in a digital format</p>	<p>use data to create reports and dashboards to gain business insights</p>	<p>assess potential future scenarios by using advanced statistical methods</p>	<p>Predictive Analytics</p>
WHAT techniques are involved	<p>DATA COLLECTION</p> <p>PREPROCESSING</p> <ul style="list-style-type: none"> • class labeling (categorical vs numerical) • data cleansing • dealing with missing values <p>CASE SPECIFIC</p> <ul style="list-style-type: none"> • e.g. balancing & shuffling datasets  	<p>DATA COLLECTION</p> <p>PREPROCESSING</p> <ul style="list-style-type: none"> • class labeling (number, text, digital images, digital video data, digital audio data) • data cleansing • dealing with missing values <p>CASE SPECIFIC</p> <ul style="list-style-type: none"> • text data mining, confidentiality-preserving data mining techniques 	<p>ANALYZE THE DATA</p> <p>EXTRACT INFO AND PRESENT IT IN THE FORM OF:</p> <ul style="list-style-type: none"> • metrics • KPIs • reports • dashboards   	<p>REGRESSION</p>  <p>LOGISTIC REGRESSION</p>  <p>CLUSTERING</p>  <p>FACTOR ANALYSIS</p>  <p>TIME SERIES</p>  <p>ML</p> <p>SUPERVISED LEARNING</p> <ul style="list-style-type: none"> • SVMs • NNs • deep learning • random forests • bayesian networks <p>UNSUPERVISED LEARNING</p> <ul style="list-style-type: none"> • k-means • deep learning <p>REINFORCEMENT LEARNING</p> <p>similar to supervised learning, but instead of minimizing the loss, one maximizes reward</p>
WHERE	<p>BASIC CUSTOMER DATA</p> <p>HISTORICAL STOCK PRICE DATA</p>	<p>SOCIAL MEDIA</p> <p>FINANCIAL TRADING DATA</p>	<p>PRICE OPTIMIZATION</p> <p>INVENTORY MANAGEMENT</p>	<p>USER EXPERIENCE (UX)</p> <p>SALES FORECASTING</p> <p>FRAUD DETECTION</p> <p>CLIENT RETENTION</p>
HOW using what tools	<p>PROGRAMMING LANGUAGES</p>  <p>SOFTWARE</p>   <p>IBM SPSS</p> 	<p>PROGRAMMING LANGUAGES</p>  <p>SOFTWARE</p>   <p>hadoop</p> 	<p>PROGRAMMING LANGUAGES</p>  <p>SOFTWARE</p>   <p>Power BI</p>  <p>Qlik Q</p>  <p>tableau</p> 	<p>PROGRAMMING LANGUAGES</p>  <p>SOFTWARE</p>   <p>IBM SPSS</p>  <p>EViews</p>  <p>STATA</p>  <p>Microsoft Azure</p>  <p>rapidminer</p> 
WHO	<p>DATA ARCHITECT</p> <p>DATA ENGINEER</p> <p>DATABASE ADMINISTRATOR</p>	<p>BIG DATA ARCHITECT</p> <p>BIG DATA ENGINEER</p>	<p>BI ANALYST</p> <p>BI CONSULTANT</p> <p>BI DEVELOPER</p>	<p>DATA SCIENTIST</p> <p>DATA ANALYST</p> <p>MACHINE LEARNING ENGINEER</p>
ARE YOU AWARE	<p>200/200 lines of data is not necessarily big data. It is not just volume that defines a data set as big - variety, variability, velocity, veracity, and other characteristics are determinative as well.</p> <p>Qualitative analysis tools such as SWOT are not used for quantitative analysis. Hence, they are not part of business intelligence.</p>			
	<p>Software like Excel, SPSS, and Stata, can be successfully used by data science teams in many companies.</p> <p>In deep learning, there is still a debate on WHY the algorithms used outperform all conventional methods.</p>			

When to use What Data Science Tools

How many open-source LLMs do you know?

Mistral-7B

outperforms Llama2 (Sep 2023)

@Alex Wang

Falcon 180B

Largest open-access model

Japanese StableLM Alpha 7B

Japanese language model

Bichuan 2

Chinese language model, equivalent to Meta's Llama

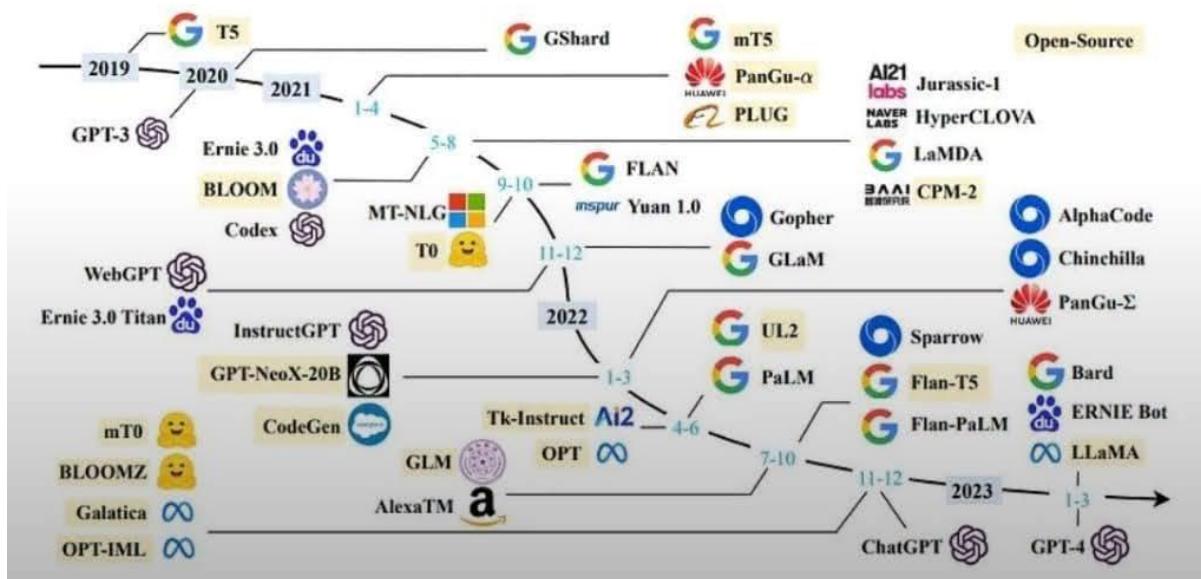
LLaMa2

comes in 3 parameter sizes - 7, 30, and 70 bn

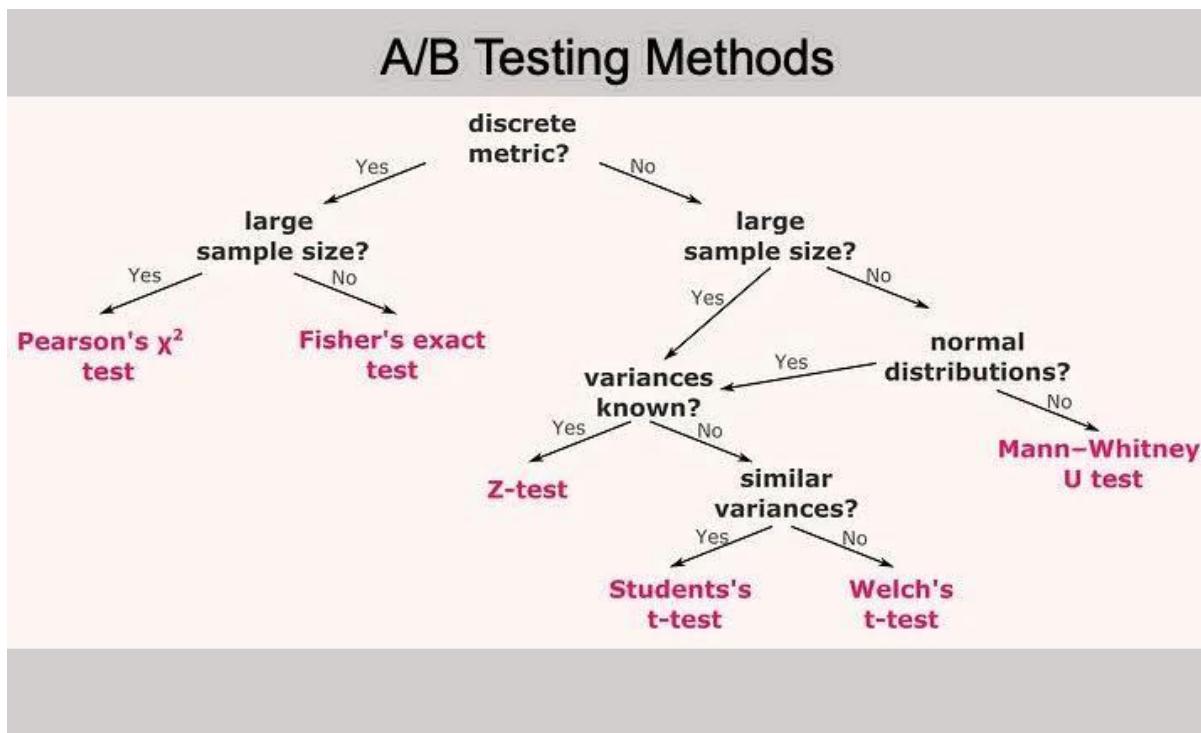
Vicuna-13B

open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT

(Mar 2023)

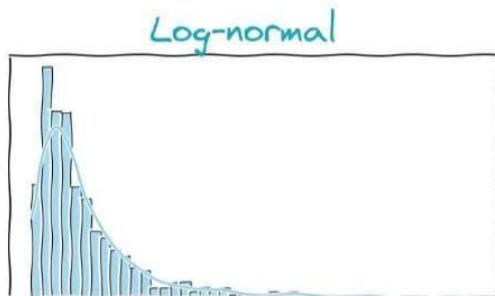
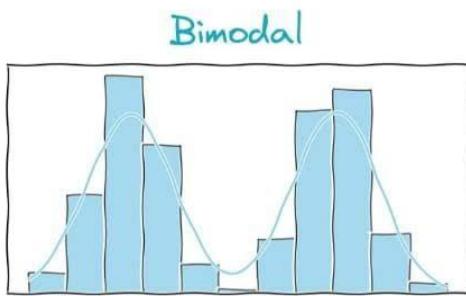
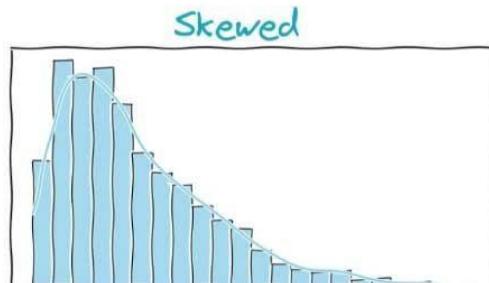
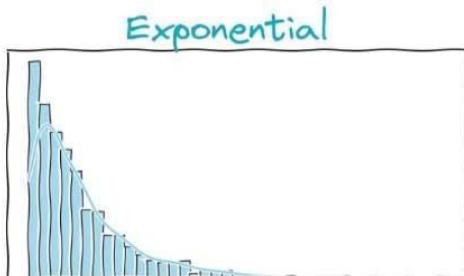
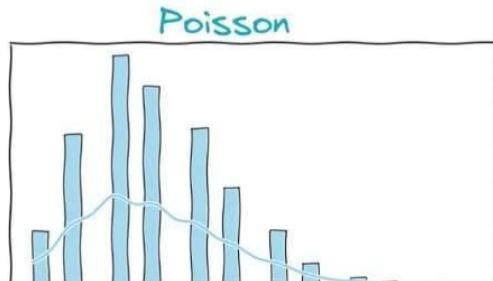
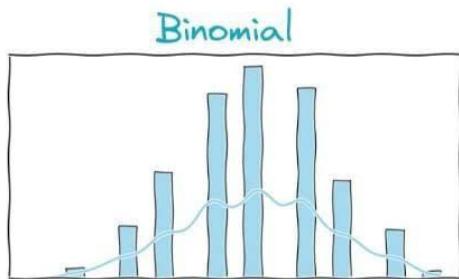
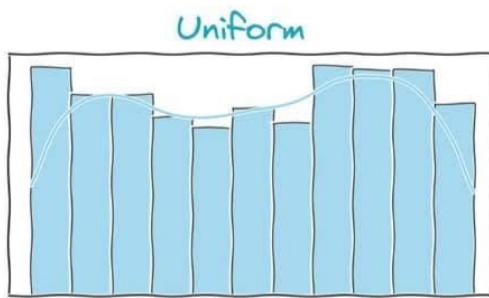
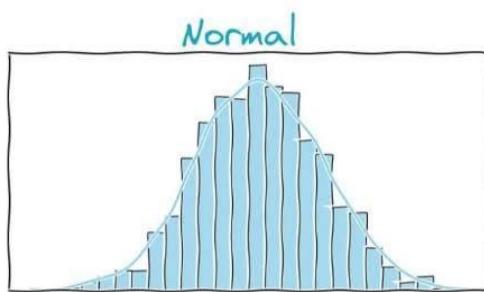


How Many Open-Source LLMs do you Know?



A/B Testing Methods

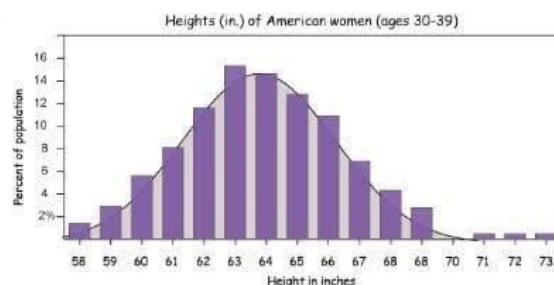
Data distributions



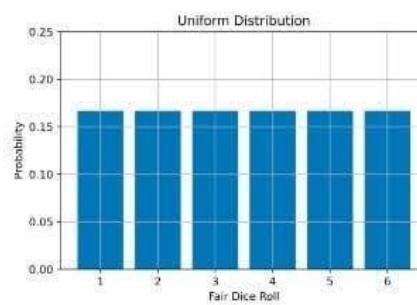
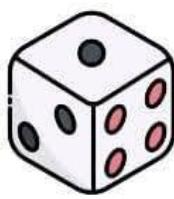
Data Distribution

Where can you find each Data Distribution?

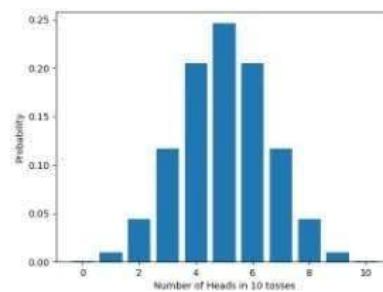
1 Normal distribution



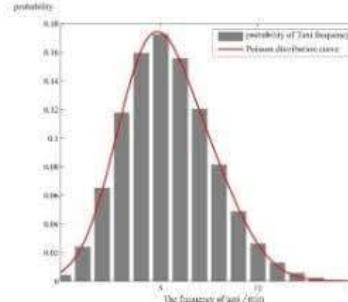
2 Uniform distribution



3 Binomial distribution



4 Poisson distribution



Where can you find each Data Distribution?

API Security Checklist

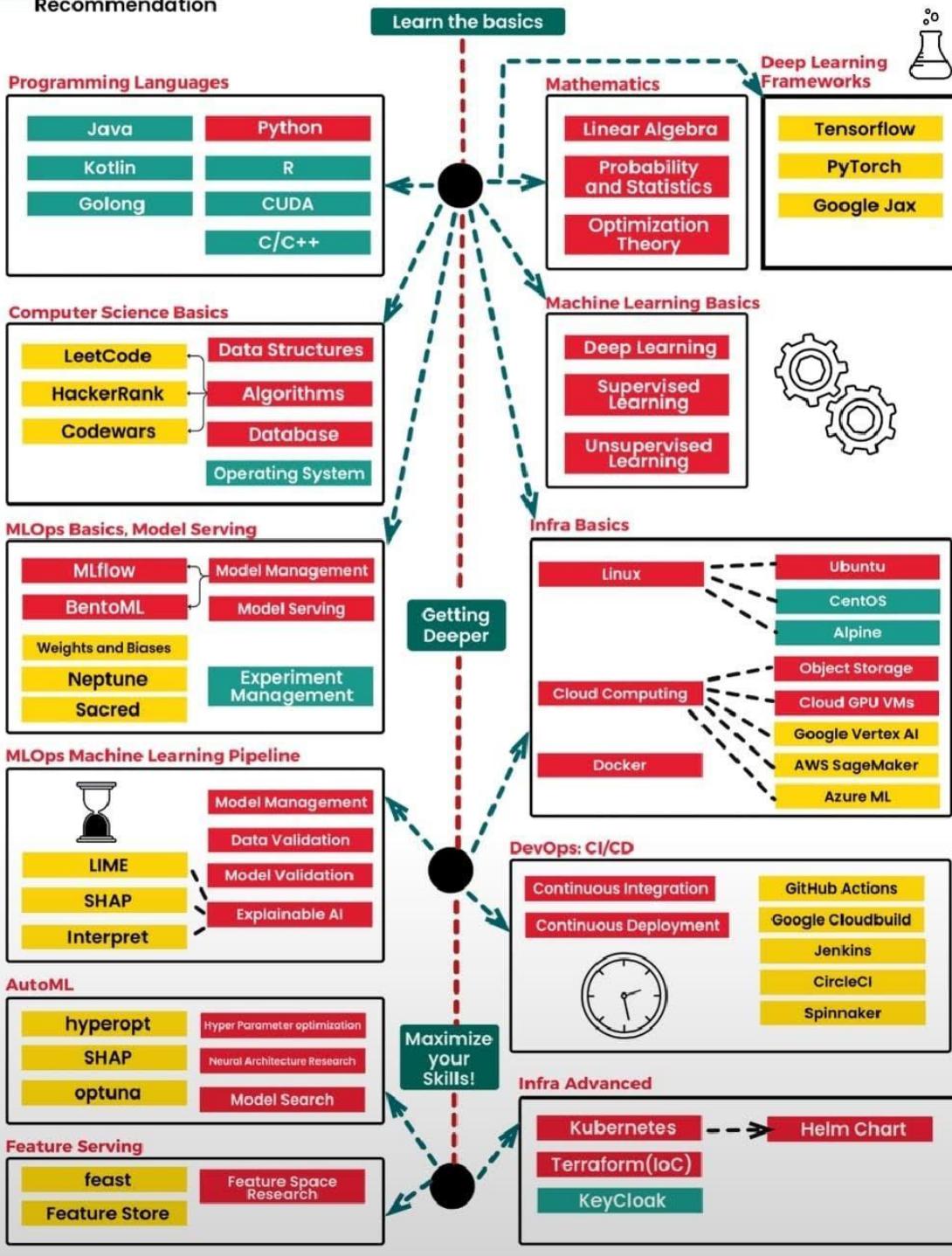
Checklist of the most important security countermeasures when designing, testing, and releasing your API.
<https://github.com/shieldfy/API-Security-Checklist>

Noisy Model Learning

1: Your foundation models are not clean since the (black-box) pre-training data are noisy! Check latest paper *Understanding and Mitigating the Label Noise in Pre-training on Downstream Tasks*. Researchers propose a new research topic: Noisy Model Learning! arxiv.org/abs/2309.17002

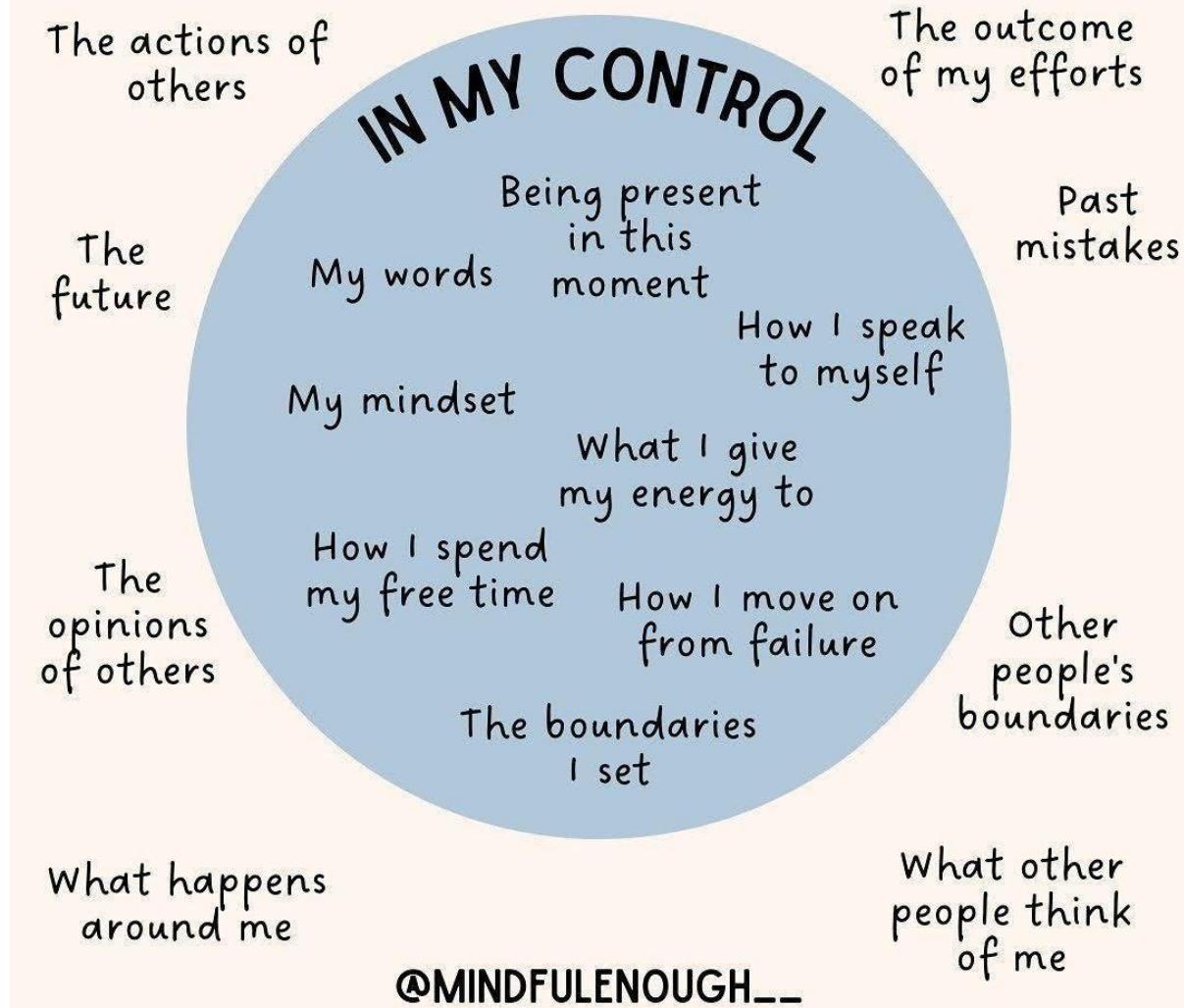
ML Engineer

- Possibilities
- Pick any
- Personal Recommendation



Machine Learning Engineer

OUT OF MY CONTROL



Circle of Controls

Llamaindex + ptx-api

Llamaindex + ptx-api (#perplexity_ai)

Perplexity has an API offering a one-stop shop for you to access open-source LLMs, and it's really fast ⚡

Custom-built on top of NVIDIA's TensorRT-LLM on A100's - up to 3-5x faster than standard inference solutions.

Models to choose from:

- Llama2

- mistral
- Code llama
- Openhermes
- Replit-code
- Pplx

How to use perplexity API, watch YT

<https://youtu.be/PHEZ6AHR57w?si=nX5NYUk5KbSjiOOD>



Read full cookbook, Jupyter Notebook showing how to plug this into your RAG system

https://github.com/run-llama/llama_index/blob/main/docs/examples/llm/perplexity.ipynb

Video lectures, Stanford CME 302 Numerical Linear Algebra autumn 2007

Video lectures, Stanford CME 302 Numerical Linear Algebra autumn 2007, by Gene Golub

<https://www.stat.uchicago.edu/~lekheng/courses/302/>

You need to be careful with Time Bias

⚠ You need to be careful with Time Bias!

Learn what it is and why it happens.

Time bias is a type of bias that can occur in time series analysis and forecasting.

This happens when the historical data used to build a time series model or forecast does not accurately represent the current or future conditions.

Time bias can arise for several reasons:

① Changes in trends or patterns

② Data contains outliers or anomalous events

① The historical data may not be representative of current trends or patterns due to changes in underlying conditions such as technology, consumer preferences, or market dynamics.

② Time bias can occur if the historical data contains outliers or anomalous events that are unlikely to occur again in the future.

To mitigate time bias in time series analysis and forecasting, it is important to carefully examine and validate the historical data and model assumptions.

This can involve performing sensitivity analyses and testing different model specifications to ensure that the model is robust to changes in underlying conditions.

Incorporating external data sources and expert insights can also help to account for changes in underlying conditions and reduce the potential for time bias.

So the next time you're working with time series data, be sure to watch out for time bias and take steps to address it.

By doing so, you can help ensure that your models and forecasts are accurate and reliable! 🌟

Data Leakage is more dangerous than Overfitting.

Data Leakage is more dangerous than Overfitting.

Why?

There is a big difference between the two.

If you understand this you will realize.

Let's define both concepts and examine how they perform on different datasets:

Overfitting

It means that the model learns the training data too well. It also learns the noise and the outliers. These are only present in the training data, so with unseen data the performance will be poor.

Performance on Training Data: From the definition we see that the performance on training data is really high.

Performance on Test Data: Test data should work as unseen data. If we properly separate it, the overfitted model will perform poorly. What the model learned during the training is not applicable to the test set.

Performance on New Data: An overfit model will perform poorly on new, unseen data for the same reasons.

An overfitted model works well only with the training data, so we can spot a bad model early, at the test stage.

Data Leakage

Data leakage is when some information from outside of the training set spoils the model. The source of this 'infection' is usually from the test set. The model 'sees into the future' and learns information it should have no access to.

Performance on Training Data: Like an overfitted model, a model with data leakage will perform well after training.

Performance on Test Data: Here comes the difference! If the test data is leaked, then the model will perform well at this stage. This is because the model could learn some aspects of the test set.

The test set should work as an unseen dataset! But here the model had some sort of access to it.

Performance on New Data: In most cases, the performance on new data will drop drastically. The good performance was generated by the leaked info, which has no help here.

If you have a too-good-to-be-true model after the test, you probably have a leakage!

Data Leakage is way more dangerous since it can give you false high performance during both the training and testing phases.

First-ever course on Data-Centric AI

The first-ever course on Data-Centric AI!

Offered by MIT, this highly practical course focuses on the impactful aspects of real-world ML applications.

While most classes primarily focus on models, the majority of work in real-world AI applications revolves around data.

In this course, you will learn:

- Automatically finding Label Errors
- Data-Centric vs Model-Centric AI
- Data-centric Evaluation of ML Models
- Interpretability in Data-Centric ML
- Obtaining high-quality Data Annotations
- Data-Centric Human Priors and Interpretability

And the best part? It's completely FREE! 🔥

<https://dcai.csail.mit.edu/>

Youtube Playlist :

<https://youtube.com/playlist?list=PLnSYPjg2dHQKdig0vVbN-ZnEU0yNJ1mo5&si=O3-tChT7NMEqeC4t>

Lab assignments for Introduction to Data-Centric AI, MIT IAP 2023 

<https://github.com/dcai-course/dcai-lab>

Anomaly detection with pytimetk

Anomaly detection with pytimetk. Time series. Easier. Faster. Pytimetk.

Explore a new beginner guide to anomaly detection in Pytimetk.

The short beginner guide is a great way for beginners to get up to speed.

Links:

👉 Anomaly Detection Beginner Tutorial:

https://business-science.github.io/pytimetk/guides/06_anomalize.html

👉 Anomaly Detection for Website Traffic:

https://business-science.github.io/pytimetk/tutorials/04_anomaly_detection.html

👉 Pytimetk Website:<https://business-science.github.io/pytimetk/>

👉 Pytimetk on GitHub:

<https://github.com/business-science/pytimetk>

Boxplots vs Raincloud

Use this plot instead.

ggdist: Make a Raincloud Plot to Visualize Distribution in ggplot2

<https://github.com/tidyverse/ggplot2>

References:

<https://www.analyticsvidhya.com/blog/2022/03/a-comprehensive-guide-on-ggplot2-in-r/>

<https://github.com/tidyverse/ggplot2/blob/main/README.Rmd>

#rstats #DataScience

Cleaning your data before building your Time Series model is crucial.

Learn how to do it, step by step  

- 1 Handle missing values
- 2 Remove trend

 <https://medium.datadriveninvestor.com/effective-strategies-for-managing-missing-values-and-detrending-time-series-data-9b41300f9c25>

- 3 Remove seasonality
- 4 Check for stationarity and make it stationary if necessary
- 5 Normalize the data

<https://medium.datadriveninvestor.com/practical-approaches-for-seasonality-removal-and-data-normalization-in-time-series-analysis-934b0259e25e>

- 6 Remove outliers

 <https://medium.datadriveninvestor.com/uncovering-patterns-in-time-series-data-by-eliminating-outliers-f8fd0b35c43>

- 7 Smooth the data



<https://medium.datadriveninvestor.com/refine-time-series-data-by-eliminating-noise-through-data-smoothing-19824eb054e3>

You may change this order depending on your data. Or even remove some steps or repeat some of them several times on different data cleaning stages.

This has the sole purpose of showing you one possibility.

Data visualization and exploration

Data visualization and exploration are essential components of the data analysis process. Free E-Book:

<https://pyoflife.com/data-visualization-and-exploration-with-r-pdf/>

#DataScience #rstats #DataVisualization #statistics #DataScientists #DataAnalytics #mathematics
#computerscience #R #programming

Focus on What You Can Control

In the tech world, things change fast. Leaders often face unexpected challenges, be it a sudden change in client needs, new market trends, or internal team issues. It's easy to feel overwhelmed, but a simple idea can help a lot: focus on what you can control. This can improve our happiness by an order of magnitude.

Here are a few tips for focusing on what you can control:

- ◆ **Identify your circle of control.** What are the things that you have direct control over? This could include your thoughts, actions, reactions, attitude, and effort. It could also include your work environment, relationships, and finances.
- ◆ **Focus on your actions.** Don't waste time and energy worrying about things outside your control. Instead, focus on taking positive steps to improve your situation.
- ◆ **Set realistic goals.** Don't try to change everything at once. Set small, achievable goals that you can build on over time.
- ◆ **Celebrate your successes.** When you achieve a goal, take the time to celebrate your accomplishment. This will help you stay motivated and focused on your plans.
- ◆ **Be kind to yourself.** We often know to think wrong about ourselves and to be too much critique. But thinking and speaking friendly about ourselves can be a game-changer in overall well-being.
- ◆ **Learn to say NO (a lot),** especially for those activities that drain your energy. Choose wisely, and your productivity will skyrocket. Every no is yes to something else you like more.

As the new week starts tomorrow, remember to focus on things you can control. For things we cannot control, **we can decide how to react to them.**

Image credits: Jacqueline Rae Legard (Mindful Enough)

#technology #techworldwithmilan #careers #personaldevelopment #productivity

Multi-Modal Retrieval

If you're building RAG, you know about text embeddings, but what about image embeddings? 

It'll be good to learn now!

CLIP is the most popular image encoder model. Here's a short  outlining the different CLIP models.

Architecture 1: ResNet

A ResNet is one of the most popular vision NN architectures. It consists of a standard CNN architecture with "residual" layers such that every additional block is additive from the previous one.

* RN-50 = 50 layers

* RN-50x4 = scaled up 4x according to EfficientNet

Architecture 2: Vision Transformer (ViT)

Instead of processing image patches via convolutions, a vision transformer processes patches via a transformers module.

* A ViT-B/32 model —> patch size = 32, small model

* ViT-L/14 —> patch size = 14, large model

Check out the diagram below (credits: original CLIP blog) to see the compute/performance curve - see the top two purple lines for CLIP ResNet and ViT models. ViT models have the highest performance for a given unit of compute, and ResNet is right below.

All of these are now supported in #llama_index, Check out notebook below:

Notebook:

https://github.com/run-llama/llama_index/blob/main/docs/examples/multi_modal/multi_modal_retrieval.ipynb

Original CLIP blog: <https://openai.com/research/clip>

Top Important Computer Vision Papers for the week from 28-8-to-3-9

<https://pub.towardsai.net/top-important-computer-vision-papers-for-the-week-from-28-8-to-3-9-24e4c9847ee0>

Do You Need to Know All Design Pattern?

The answer is no. Even though we have 23 design patterns, around 10 are mostly used in everyday development. Knowing which patterns exist overall is good, but you need to know these very well.

Design patterns can be divided into three main types:

1. Creational Patterns

These design patterns deal with object creation mechanisms, trying to create objects in a manner suitable to the situation.

Important patterns in this group are:

- ◆ **Factory:** This pattern allows delegating the instantiation logic to factory classes. The Factory Method creates objects without exposing the instantiation logic to the client.
- ◆ **Singleton:** The Singleton pattern ensures that a class has only one instance and provides a global point of access to it. It's useful when exactly one object is needed to coordinate actions across the system.

2. Structural Patterns

These patterns deal with the composition of classes and objects that form larger structures.

Important patterns in this group are:

- ◆ **Adapter:** This pattern works as a bridge between two incompatible interfaces. It wraps an existing class with a new interface to become compatible with the client's interface.
- ◆ **Facade:** The Façade pattern provides a unified interface to a set of interfaces in a subsystem. Façade defines a higher-level interface that makes the subsystem easier to use.
- ◆ **Decorator:** This pattern dynamically adds/overrides behavior in an existing method of an object. This pattern provides a flexible alternative to subclassing for extending functionality.
- ◆ **Proxy:** The Proxy pattern provides a surrogate or placeholder for another object to control access to it. In its most general form, a proxy is a class functioning as an interface to something else.

3. Behavioral Patterns

These patterns are specifically concerned with communication between objects and how they interact and distribute work.

Important patterns in this group are:

- ◆ **Command:** The Command pattern encapsulates a request as an object, thus allowing users to parameterize clients with queues, requests, and operations.
- ◆ **Template Method:** This pattern defines the program skeleton of an algorithm in a method called template method, which defers some steps to subclasses.
- ◆ **Strategy:** The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it.
- ◆ **Observer:** This pattern defines a one-to-many dependency between objects so that all its dependents are notified and updated automatically when one object changes state.

#technology #softwareengineering #programming #designpatterns #developers

Embedding In OpenAI API

<https://machinelearning-basics.com/embedding-in-openai-api/>

8 API Architectural Styles

SOAP



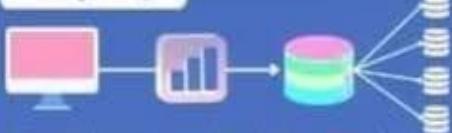
XML-based, strict, operates over HTTP/HTTPS, favored in enterprises

REST



Building scalable APIs, using standard HTTP methods

GraphQL



Query language for APIs, allowing clients to request specific data

gRPC



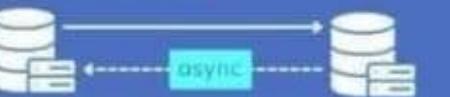
High-performance framework for remote procedure calls, using HTTP/2

WebSocket



Bidirectional, real-time communication protocol

Webhook



Event-driven, server-side mechanism that sends HTTP callbacks

MQTT



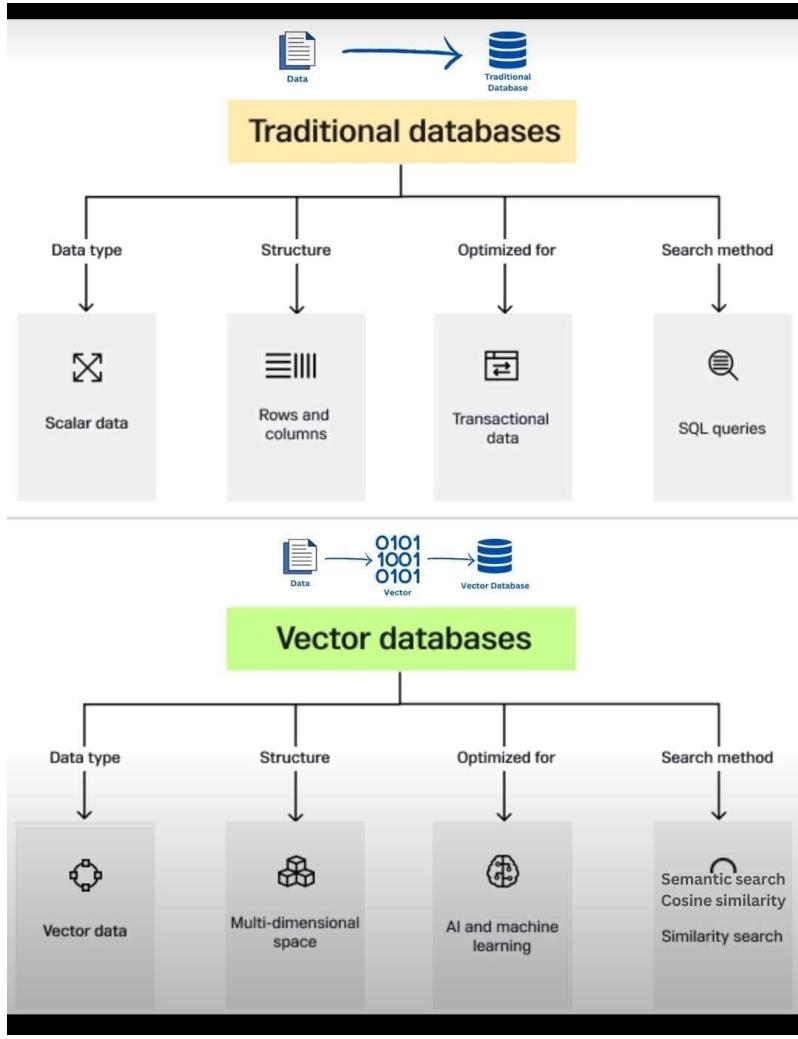
Lightweight messaging protocol for machine-to-machine or "IOT" connectivity

AMQP

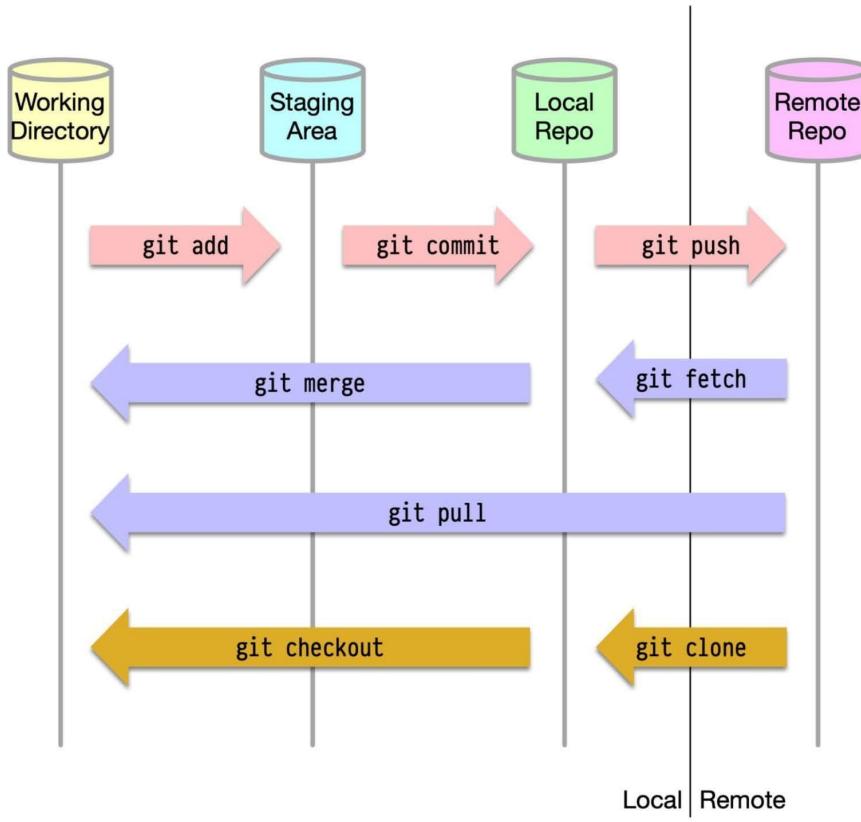


Open standard protocol for message-oriented middleware

API Architectural Styles



Databases



How Github Works?

10 AI Tools, 10 websites that will save you hundreds of hours

10 AI Tools, 10 websites that will save you hundreds of hours:

1. Typedream AI

Build your websites without writing any code from just one AI prompt.

 typedream.com

2. Monster API

Turn your Text Prompts into Stunning Images.

The best part is that you can use a single prompt for multiple styles.

 monsterapi.ai

3. Kickresume

Your success story begins with a resume

Create a beautiful resume quickly with the help of artificial intelligence

 kickresume.com

4. Gamma AI

A GPT-4 powered app that can create compelling presentation decks in minutes.

 gamma.app

5. Stockimg

Stockimg is an AI image generation tool that lets you instantly create high-quality, custom assets for your brand.

 stockimg.ai

6. Lovo

Realistic AI voices that captivate your audience

Surprise your audience with the most realistic and human AI voices in the world

 lovo.ai

7. ChatSimple

Set up AI-powered 24/7 customer service chatbots in just 5 minutes with no coding required.

 chatsimple.ai

8. Perplexity

ChatGPT on steroids 🎉

Get instant answers along with cited sources on any topic.

 perplexity.ai

9. Decktopus AI

Introducing the future of presentations.

Create amazing presentations in minutes!

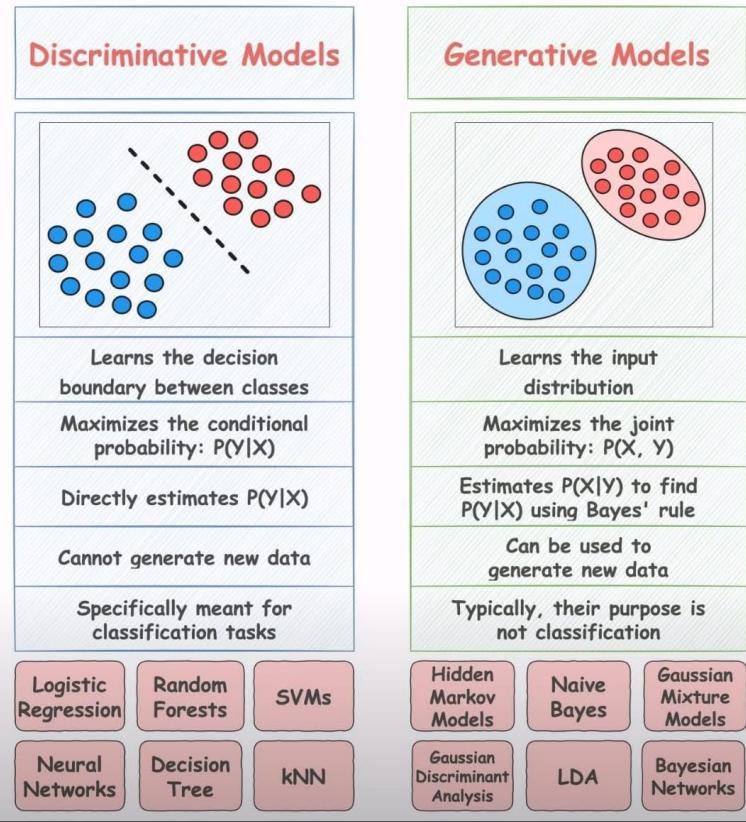
 decktopus.com

10. LensGo

Make all your wonders visual.

AI-based image and video generation using customized models.

 lensgo.ai



Discriminative vs Generative Models

Most Used Design Patterns Cheat Sheet

Creational Patterns

Used to construct objects

Structural Patterns

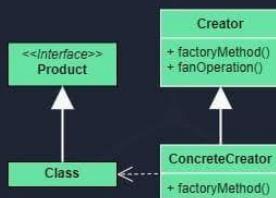
Used to form large object structures

Behavioral Patterns

Used to manage algorithms and relationships

Factory Method

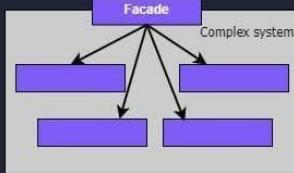
Use when you want to delegate object creation to subclasses.



Example: create GUI component

Facade

Use when you want to provide a simplified interface to a complex subsystem

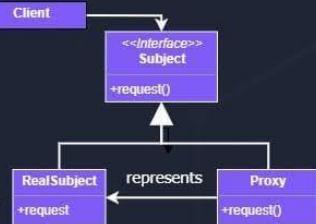


Example: Providing a simple interface to a complex subsystem

Proxy

Use for object access control

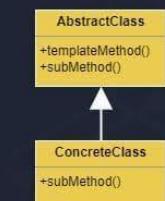
Example: Controlling access to sensitive resources



Template Method

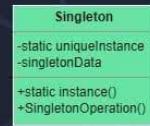
Use when you want to break down an algorithm into a series of steps

Example: Common behavior should be located in one class



Singleton

Use when you want to have one instances of a class.



Example: logging, db connections.

Adapter

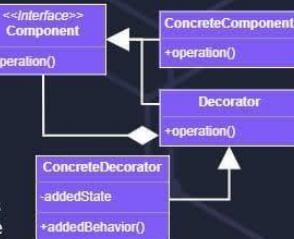
Use when you need to convert an interface to another interface



Example: make incompatible classes work together

Decorator

Use when you need to wrap objects to modify their behaviors

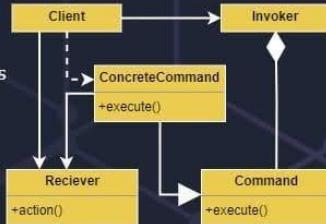


Example: make object behaviors dynamically modifiable

Command

Use for encapsulating requests with parameters

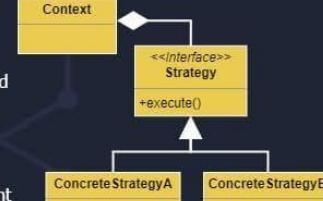
Example: Implementing operations



Strategy

Use for interchangeable algorithms that can be swapped at runtime

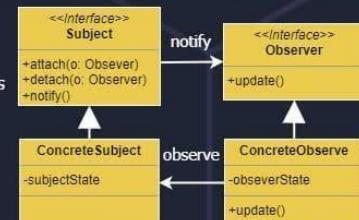
Example: Implement different sorting algorithms



Observer

Use for automatic updates of dependant objects

Example: Implement subscribers



Most Used Design Patterns Cheatsheet

Stay ahead of the curve with these 12 powerful GitHub repositories for learning data science, analytics, and engineering

This blog lists down-trending data science, analytics, and engineering GitHub repositories that can help you with learning data science to build your own portfolio:

https://datasciencedojo.com/blog/trending-github-repositories/?utm_campaign=DSD%20blogs%202023&utm_content=273096354&utm_medium=social&utm_source=twitter&hss_channel=tw-1318985240#

#datascience #dataanalytics #dataengineering

Introduction MMMU, a Massive Multi-discipline Multimodal Understanding

Introducing MMMU, a Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI.

<https://mmmu-benchmark.github.io>

<https://arxiv.org/abs/2311.16502>

💡 Highlights of the MMMU benchmark:

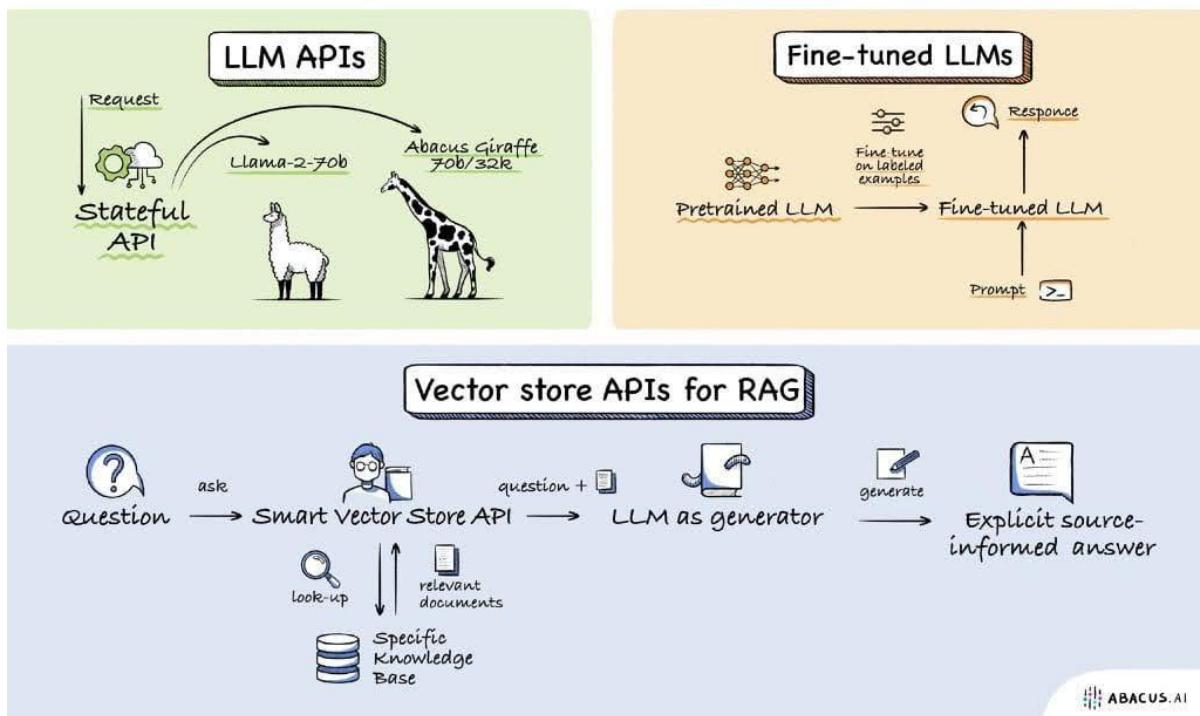
- > 11.5K meticulously collected multimodal questions from college exams, quizzes, and textbooks
- > Spanning Art & Design 🎨, Business 💼, Science 🔬, Health & Medicine 🩺, Humanities & Social Science 📚, Tech & Engineering 🚗 across 30 subjects and 183 subfields
- > 30 heterogeneous image types 🌏📈🎵, such as charts, diagrams, maps, tables, music sheets, and chemical structures
- > Focuses on advanced perception and reasoning with domain-specific knowledge 🧠
- 📊🔍 Results and Takeaways from evaluating 14 open-source models and #GPT4-Vision:
 - > 🤖💡 MMMU Benchmark post a great challenge to existing #LMMs: #GPT4V only hits 56% accuracy, showing a vast landscape for #LMMs advancement.
 - > 💪 Long way to go for open-source LMMs. Top open-source models like #BLIP2-FLAN-T5-XXL and #LLaVA-1.5 achieve around 34% accuracy.
 - > 📄📝 OCR and captions addition to #LMMs show little gain in MMMU, highlighting the need for deeper joint image-text interpretation.
 - > Models tend to perform better on photos and paintings 🖼 than on diagrams and tables 📈, where nuanced and fine-grained visual information persists.

> 🚗 Error analysis on 150 error cases of GPT-4V reveals that 35% of errors are perceptual, 29% stem from a lack of knowledge, and 26% are due to flaws in the reasoning process.

Introduction to Important AI Tools/Courses

1. 🤖📚 AI Essentials: <https://lnkd.in/dyEt4DGt>
2. 💬🧠 ChatGPT Mastery: <https://lnkd.in/eiRtk-6q>
3. 🎨🌟 Google AI Magic: <https://lnkd.in/eBQXfBe9>
4. 🐍🎓 Harvard AI Intro: <https://lnkd.in/eu4mZaAG>
5. 🛠️💡 Microsoft AI Basics: <https://lnkd.in/eYNWzXUX>
6. 🚀🔧 Prompt Engineering Pro: https://lnkd.in/eNi_YNSe
7. 🌐🌐 Google's Ethical AI: <https://lnkd.in/eTrwSU89>
8. 📊🎥 ML by Harvard: <https://lnkd.in/eX28syMJ>
9. 💡🔗 LangChain LLMs: <https://lnkd.in/evZVJbNy>
10. 📈💼 Bing Chat Boost: <https://lnkd.in/ejN-qrVy>
11. 🧠👁️ Visionary LLM Apps: <https://bit.ly/46ACTS8>
12. 🌟📚 Generative AI by MS: <https://lnkd.in/dqjnzcCD>
13. 🚗🤖 Amazon's AI Strategy: <https://lnkd.in/dFhmsvZC>
14. 🌎🤖 AI for All: <https://lnkd.in/eFx7zCz7>
15. 🌎🧠 AWS AI Foundations: <https://lnkd.in/dEjN9PRm>

LLM + LLMops APIs - 60x CHEAPER THAN GPT-4V



LLM+LLMops APIs

Generating or engineering features from Time Series data

Generating or engineering features from Time Series data when using an ML approach involves extracting meaningful information that can be used by algorithms to understand patterns, make predictions, or identify trends.

Here are some feature engineering techniques  

▶ Time-Based Features:

Extracting features like hour, day, week, month, year, or season can be very informative, especially if the time series shows periodicity or seasonality.

▶ Lag Features:

These are values at previous time steps. For instance, the value of a time series at time $t-1$, $t-2$, etc., can be used as a feature to predict the value at time t .

▶ Rolling Window Statistics:

Calculate statistics (mean, median, variance, etc.) over a rolling window. This helps in capturing trends and patterns over specified time intervals.

▶ Expanding Window Statistics:

Similar to rolling windows but the window size increases over time, providing cumulative statistics.

▶ Differencing Features:

This involves calculating the difference between two consecutive observations. This can help in removing trends and seasonality.

▶ Fourier Transforms:

To capture cyclical patterns, Fourier transforms can be used to extract periodic frequencies and amplitudes.

▶ Wavelet Transforms:

Useful for capturing both frequency and location information in your time series data, particularly for non-stationary time series.

▶ Autoregressive Features:

These features come from models that use a combination of past values to predict future values, like ARIMA models.

▶ Decomposition Components:

Decompose time series into trend, seasonal, and residual components. Each of these components can be used as separate features.

▶ Domain-Specific Features:

Depending on the specific domain (finance, weather, etc.), certain features might be more relevant, like holiday effects in sales data.

► Binary or Categorical Flags:

Create flags or categorical variables for specific events or conditions (e.g., weekends, holidays, events).

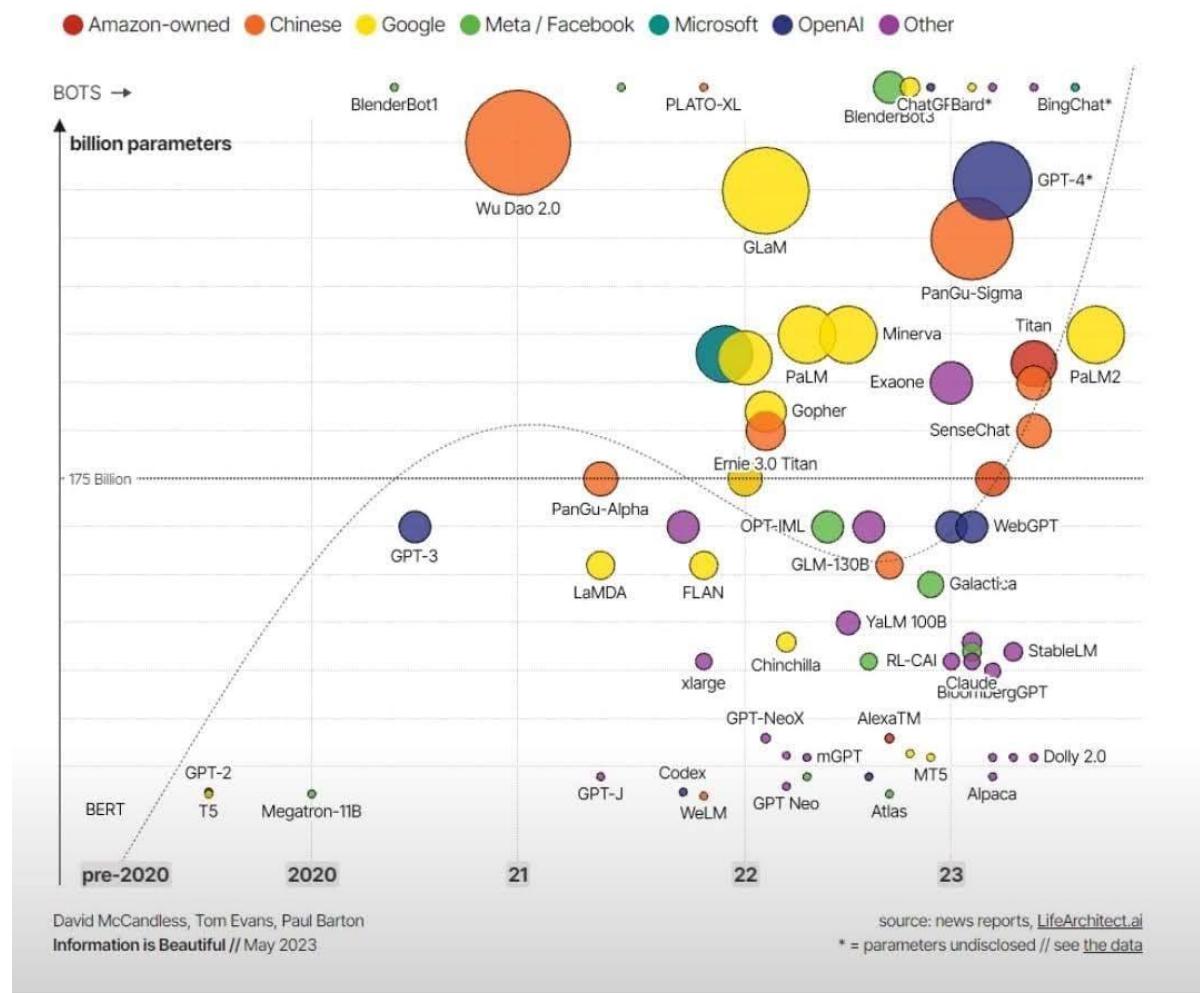
► Cross-Correlation with External Time Series:

Sometimes, the relationship between two time series can be a feature, especially if one series can be predictive of another.

Each of these techniques can bring out different aspects of the time series data, and the choice of features often depends on the specific problem and the nature of the data you're working with.

#FeatureEngineering #TimeSeries #AI #DataScience #MachineLearning

How Many LLMs Do You Know?



How Many LLMs do you know?

The latest ML research:

The latest ML research:

- UltraFastBERT
- Orca 2
- MultiLoRA
- System 2 Attention
- White-Box Transformers
- GAIA Benchmark
- Direct Preference for Denoising Diffusion Policy Optimization (D3PO)

UltraFastBERT: Revolutionizes BERT models by reducing neuron usage, achieving massive speedup in language model inference

<https://t.co/xXXH85HDDp>

Orca 2: Trains smaller language models for enhanced reasoning, challenging the capabilities of larger counterparts

<https://t.co/cL6X7llamr>

MultiLoRA: Improves LLMs for multi-task learning, optimizing performance with minimal additional parameters

<https://t.co/eIMxYrgTPm>

System 2 Attention – Is Something You Might Need Too (Meta): Aims to improve LLMs' response quality by refining attention mechanisms

<https://t.co/WXzUZaHuqa>

Hashing Vs Encryption vs Encoding

🔗 Hashing Vs Encryption vs Encoding: A Brief Comparison

📜 Hashing:

This is primarily used for data integrity verification and password storage.

It involves applying the Hash function to data, this generates a fixed-size structure of characters (hash value) that is unique to the input data.

🔒 Encryption:

This is used to protect the confidentiality of data during transmission or storage.

It uses algorithms and a key to transform plaintext into cipher text , making it unreadable without the corresponding decryption key.

🛡 Encoding:

Encoding is a method of converting data to a different format for safe and efficient data transmission. Encoding is typically reversible, meaning the original data can be obtained by reversing the process. Unlike Hashing and Encryption, Encoding is not designed to keep data secret.

Checklist for Data Scientists or Machine Learning or AI practitioners

Checklist for Data Scientists or Machine Learning or AI practitioners

Top 10 common errors juniors can make.

Check this list before training your model to make sure you don't make these mistakes!  

1 Data Leakage in Training and Evaluation:

A significant mistake is allowing data leakage from training to evaluation sets. This can give a false impression of model performance, especially in cases involving temporal elements.

2 Inadequate Handling of Temporal Data:

Junior data scientists often struggle with handling temporal data, leading to data leakage. Suggestions like participating in ML competitions (e.g., Kaggle) were given for better learning.

3 Preprocessing Data Before Splitting:

A common error is preprocessing data (like normalization) before splitting into training and test sets, which can bias the test data.

4 Incorrect Application of Categorical Encoding:

Using methods like target encoding on the entire dataset rather than just the training set can lead to leakage.

5 Overestimating Model Performance:

There's a tendency to overestimate a model's performance, especially when initial results seem exceptionally good, without thorough debugging and validation.

6 Lack of Domain Knowledge and Data Intuition:

Junior data scientists often focus too much on modeling without gaining a sufficient understanding of the data and its domain context.

7 Not Starting with a Simple Baseline:

There's a tendency to skip establishing a simple baseline model to understand its limitations before moving to complex models.

8 Overreliance on Complex Models:

New data scientists often prefer complex or 'fancy' algorithms without assessing their necessity or effectiveness for the specific data type or problem.

9 Failure to Recognize Business Impact:

Many junior data scientists fail to understand how their models can practically impact business decisions or actions.

10 Mistaking Features for Causes:

There is confusion between identifying important features in a model and interpreting them as the actual causes of outcomes.

#AIChecklist #DataScientistChecklist
#MLPractitionerChecklist #AI #MachineLearning #DataScience

Activation functions while working with neural nets.

- Sigmoid
- Tanh
- ReLu & Leaky ReLu
- Gelu

Ever wondered why they are so important ?

You can think of a layer in neural net as a function & multiple layers makes the network a composite function.

Now, a composite function consisting of individual linear functions is also linear.

Check this ↗

We have a simple neural net that does binary classification.

Scenario 1:

- Linear decision boundary
- Linear Activation function

Observe how the neural net is able to quickly learn & loss converges to zero.

Scenario 2:

- Non Linear decision boundary
- Linear Activation function

Observe how the neural net struggles to learn & the loss consistently remains high!

With linear activations it's unable to create a non-linear decision boundary.

Watch this ↗

Scenario 3:

- Non Linear decision boundary
- Non-linear Activation function (Sigmoid)

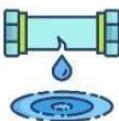
Observe how the neural net performs well this time.

With a non-linear activation function we give the network ability to create a non-linear decision boundary.

Watch this ↗

Playground :

<http://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=4,2&seed=0.92253&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>

- Data Leakage in Training and Evaluation? 
- Adequate Handling of Temporal Data? 
- Preprocessing Data Post-Splitting? 
- Correct Categorical Encoding Application? 
- Overestimation of Model Performance? 
- Sufficient Domain Knowledge and Data Intuition? 
- Starting with a Simple Baseline Model? 
- Necessity and Effectiveness of Complex Models? 
- Business Impact Consideration? 
- Mistaking Features for Causes? 

Critical Considerations in Model Development and Evaluation

Open-Source LLMs vs. ChatGPT

Open-Source LLMs vs. ChatGPT:

1. General Capabilities: Llama-2-chat-70B variant exhibits enhanced capabilities in general conversational tasks, surpassing the performance of GPT-3.5-turbo; UltraLlama matches GPT-3.5-turbo's performance in its proposed benchmark.
2. Agent Capabilities (using tools, self-debugging, following natural language feedback, exploring environment): Lemur-70B-chat surpasses the performance of GPT-3.5-turbo when exploring the environment or following natural language feedback on coding tasks. AgentLlama-70B achieves comparable performance to GPT-3.5-turbo on unseen agent tasks. Gorilla outperforms GPT-4 on writing API calls.
3. Logical Reasoning Capabilities: fine-tuned models (e.g., WizardCoder, WizardMath) and pre-training on higher quality data models (e.g., Lemur-70B-chat, Phi-1, Phi-1.5) show stronger performance than GPT-3.5-turbo.

4. Modeling Long-Context Capabilities: Llama-2-long-chat-70B outperforms GPT-3.5-turbo-16k on ZeroSCROLLS.

5. Application-specific Capabilities:

- query-focused summarization (fine-tuning on training data is better)
- open-ended QA (InstructRetro shows improvement over GPT3)
- medical (MentalLlama-chat-13 and Radiology-Llama-2 outperform ChatGPT)
- generate structured responses (Struc-Bench outperforms ChatGPT)
- generate critiques (Shepherd is almost on-par with ChatGPT)

6. Trust-worthy AI:

- hallucination: during finetuning - improving data quality during fine-tuning; during inference - specific decoding strategies, external knowledge augmentation (Chain-of-Knowledge, LLM-AUGMENTER, Knowledge Solver, CRITIC, Prametric Knowlege Guiding), and multi-agent dialogue.
- safety: GPT-3.5-turbo and GPT-4 models remain at the top for safety evaluations. This is largely attributed to Reinforcement Learning with Human Feedback (RLHF). RL from AI Feedback (RLAIF) could help reduce costs for RLHF.

 <https://arxiv.org/abs/2311.16989>

What is Table Transformer (TATR)?

Table Transformer (TATR) is a deep learning model for extracting tables from unstructured documents (PDFs and images). This is also the official repository for the PubTables-1M dataset and GriTS evaluation metric.

<https://github.com/microsoft/table-transformer>

The Table Transformer is equivalent to DETR, a Transformer-based object detection model. Note that the authors decided to use the "normalize before" setting of DETR, which means that layernorm is applied before self- and cross-attention.

<https://huggingface.co/microsoft/table-transformer-detection>

Open Source LLMs, Fine-Tunes and RAG Based Vector Store APIs - The Abacus.AI Blog

<https://blog.abacus.ai/blog/2023/10/19/open-source-langs-fine-tunes-and-rag-based-vector-store-apis/>

Trip down the GPU lane with Machine Learning | PPT

<https://www.slideshare.net/RenaldasZioma/trip-down-the-gpu-lane-with-machine-learning-83311744>

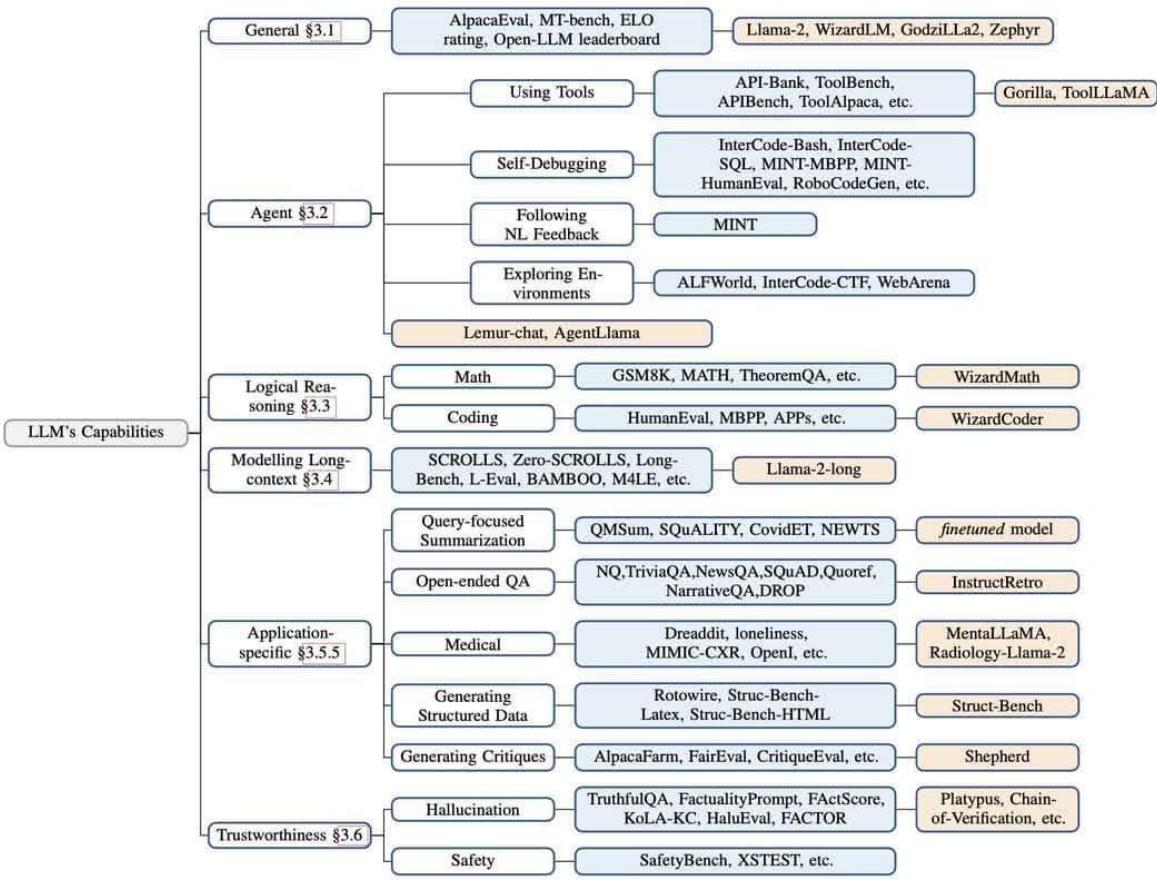


Figure 2: Typology of LLM’s capabilities and best performing open-LLMs. White boxes denote domains, blue boxes represent specific datasets and orange boxes denote open-sourced LLMs.

LLM Capabilities

The 40-hour LLM application roadmap: Learn to build your own LLM applications from scratch

<https://datasciencedojo.com/blog/build-custom-llm-application/#>

What's next in large language model (LLM) research? Here's what's coming down the ML pike

<https://venturebeat.com/ai/whats-next-in-large-language-model-llm-research-heres-whats-coming-down-the-ml-pike/>

<https://github.com/Developer-Y/cs-video-courses>

Feature Engineering

Feature Selection

Feature Transformation

Feature Creation
(Encoding, Binning)

Feature Extraction
(Automated in Deep Learning)

Feature Engineering

Accelerating Generative AI with PyTorch II: GPT, Fast | PyTorch

https://pytorch.org/blog/accelerating-generative-ai-2/?utm_content=273712247&utm_medium=social&utm_source=facebook&hss_channel=fbp-1620822758218702

Google Cloud Skills Boost

<https://www.cloudskillsboost.google/parts/118>

Table detection and recognition with Table Transformer (TATR).

Parsing tables in PDFs is a super important RAG use case. Explore using the recent Table Transformer model (#deepmock) combined with GPT-4V gives you superpowers 💪

The Table Transformer model extracts tables from PDFs using object detection 

Check full notebook guide  on the following: 1) CLIP to retrieve relevant pages, 2) Table Transforms to extract table images, and 3) GPT-4V to help synthesize an answer.

Compare to three other multi-modal table understanding baselines:

- Retrieving entire page via CLIP and feeding to GPT-4V
- Use GPT-4V to extract text from each page, index/retrieve based on text
- Do OCR on extracted table images, use that as context

Notebook:

https://github.com/run-llama/llama_index/blob/main/docs/examples/multi_modal/multi_modal_pdf_tables.ipynb

Docs:

https://docs.llamaindex.ai/en/latest/use_cases/multimodal.html#multi-modal-rag-on-pdf-s-with-tables-using-microsoft-table-transformer

Demo: table detection & recognition with Table Transformer (TATR).

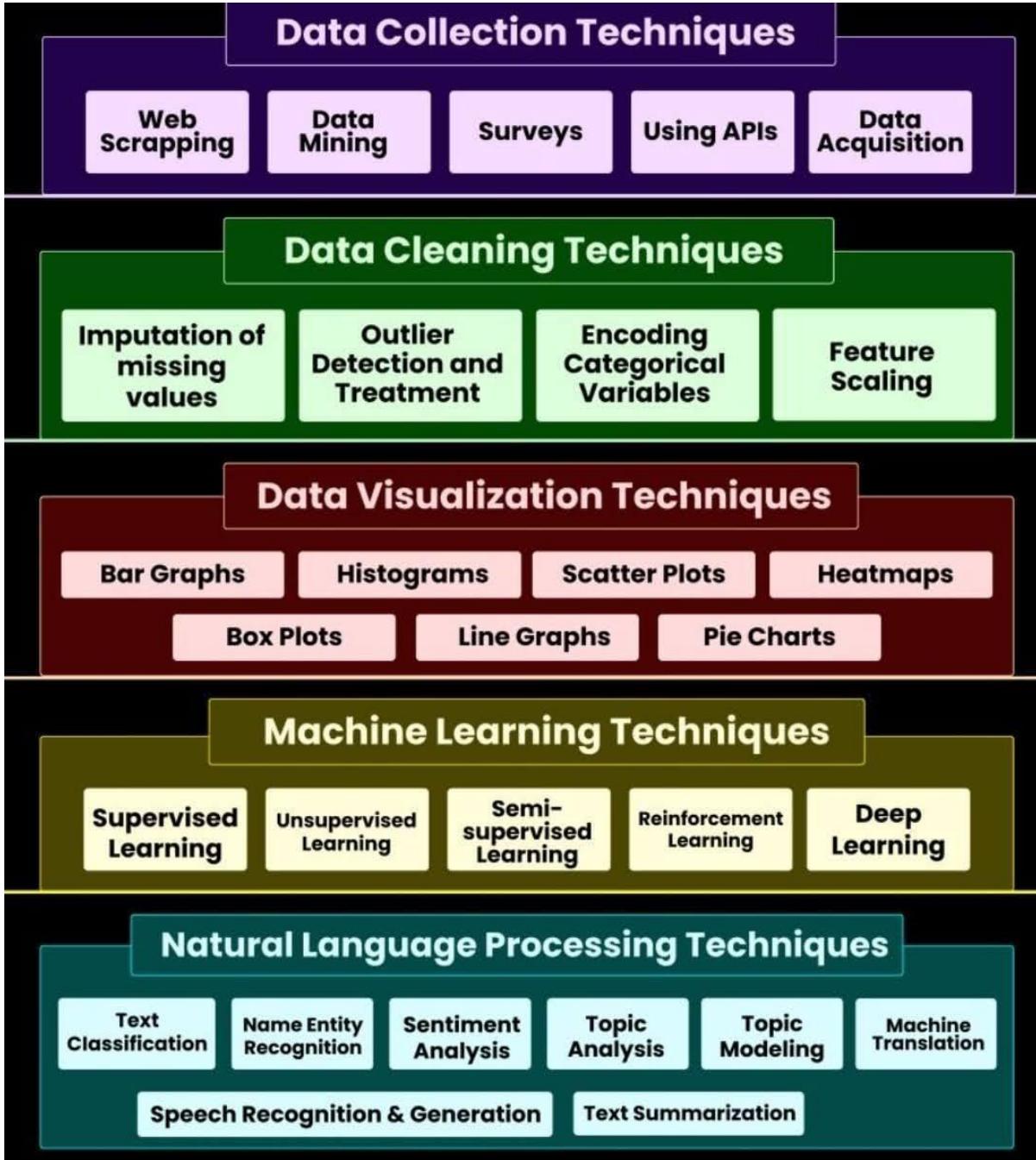
<https://huggingface.co/spaces/niehsr/tatr-demo>

Glossary of artificial intelligence - Wikipedia

https://en.wikipedia.org/wiki/Glossary_of_artificial_intelligence

Machine learning - Wikipedia

https://en.wikipedia.org/wiki/Machine_learning



Data Collection Techniques

Document Embedding

Context length is one of the biggest problems with LLMs such as ChatGPT.

There is a limitation on the number of words in your prompt because the models can only accept a certain number of tokens.

The solution? Embeddings.

Word embedding is a technique used to represent documents with a dense vector representation.

The vocabulary in these documents is mapped to real number vectors. Semantically similar words are mapped close to each other in the vector space.

For instance, you want to ask a question to one of videos, which are over two hours long.

The first step is to transcribe the video. The transcription is longer than the input context that a GPT model can accept.

The solution is to break up the transcription into shorter sentences and create word embeddings for each.

Next, you create the embedding for the input question. Then you compare the embedding of the question to the embeddings of the transcription and return the top, say 3 most similar embeddings.

Now instead of passing the entire transcribed text to the model, you will pass the context as those similar embeddings. With that, you can talk to a PDF, a transcribed video, etc., without passing the entire video or PDF to the model, which is impossible.

How to Represent Words as Numbers

Before creating word embeddings, you must convert the words to some numerical representation. For example, consider the sentence, "The cat sat on the mat".

Each word can be represented in a matrix with 0 indicating the absence of the word and 1 its presence.

The above approach is inefficient because it leads to a vector with many zeros, a sparse matrix. The alternative is to represent each word with a unique integer. In "The cat sat on the mat" you can define the words as:

- The 1
- cat 2
- sat 3
- on 4
- the 1
- mat 5

Therefore the sentence will be numerically represented as [1, 2, 3, 4, 1, 5], which is a dense vector.

A word embedding represents the words in a text corpus with floating point values while considering the relationship between the different words.

These relationships are learned when training the embeddings. The size of the embedding vector can be assigned manually.

The Embedding layer is used for learning word embeddings in TensorFlow.