Goals

**Table Of Content:**

[**YEAR 2025 1**](#_61xl00okq7g)

[Goals: 03/04/2025: 1](#_92k9cwvp0k9)

[Details What You Need To Learn AI To Create An AI Application: 1](#_txdngyyds0uh)

[Details What You Need To Learn DotNet To Create Website: 2](#_8fob17garbw7)

[Details What You Need To Learn English Writing + Grammar: 2](#_x72uk4w3y6oy)

[Details What You Need To Learn Japanese N4: 2](#_2m8xn76pc0u6)

[Details What You Need To Learn DevOps (AWS): 2](#_ie3tzyyp7oix)

[Details What You Need To Learn DevOps (Docker): 3](#_p4ezekd0z90v)

# **YEAR 2025**

## **Goals: 03/04/2025:**

| **No.** | **Name Of Goals.** | **Deadline.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- |
| 01 | Learn AI To Create An AI Application | 03/04/2025 - 30/05/2025 | Start |  |
| 02 | Learn DotNet To Create Website | 03/04/2025 - 30/05/2025 | Start |  |
| 03 | Learn English Writing + Grammar | 03/04/2025 - 30/05/2025 | Start |  |
| 04 | Learn Japanese N4 | 03/04/2025 - 30/05/2025 | Start |  |
| 05 | Learn DevOps (AWS, Azure) | 03/04/2025 - 30/05/2025 |  |  |
| 06 | Learn DevOps (Docker) | 03/04/2025 - 30/05/2025 |  |  |

### **Details What You Need To Learn AI To Create An AI Application:**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn AI To Create An AI Application |  |  |  |  |

### **Details What You Need To Learn DotNet To Create Website:**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn DotNet To Create Website | Introduction | 03/04/2025 | Done | <https://www.youtube.com/watch?v=ANcDGrtdf9M&list=PLayYhLZuuO9tV9PV2yeNQXolPc7XXw_7o&index=5&t=6092s&ab_channel=gi%C3%A1o.l%C3%A0ng> |
| 01 | Learn DotNet To Create Website | Download | 04/04/2025 | Done | <https://www.youtube.com/watch?v=5mVTizEhFbk&list=PLayYhLZuuO9tV9PV2yeNQXolPc7XXw_7o&index=6&ab_channel=gi%C3%A1o.l%C3%A0ng> |

### **Details What You Need To Learn English Writing + Grammar:**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn English Writing + Grammar |  |  |  |  |

### **Details What You Need To Learn Japanese N4:**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn Japanese N4 |  |  |  |  |

### **Details What You Need To Learn DevOps (AWS):**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn DevOps (AWS, Azure) |  |  |  |  |

### **Details What You Need To Learn DevOps (Docker):**

| **No.** | **Name Of Goals.** | **Details What You Need to Learn.** | **Deadline For  Each Part.** | **Status.** | **Note.** |
| --- | --- | --- | --- | --- | --- |
| 01 | Learn DevOps (Docker) |  |  |  |  |

Main-Goals

**AI ENGINEER**

**ENGLISH (IELTS 6.5)**

**JAPANESE (N2)**

AI-Engineer

### 



### **Dưới đây là phiên bản chi tiết hơn, nhấn mạnh vào Machine Learning & Deep Learning để gây ấn tượng với nhà tuyển dụng:**

### 

### **Technical Skills**

### **Programming Languages:** Python, Java, C/C++, JavaScript, HTML/CSS

### **Machine Learning & Deep Learning:**

### Frameworks & Libraries: PyTorch, TensorFlow, Keras, Scikit-learn

### Computer Vision: OpenCV, Pillow, Detectron2, YOLO, MMDetection

### Natural Language Processing (NLP): NLTK, SpaCy, Hugging Face Transformers, FastText, BERT, GPT, T5

### Model Optimization & Deployment: TensorRT, ONNX, OpenVINO, TensorFlow Lite

### AutoML & Hyperparameter Tuning: Optuna, Hyperopt, Ray Tune

### Explainable AI (XAI): SHAP, LIME

### MLOps & Model Deployment: MLflow, Kubeflow, FastAPI, Flask, Docker, AWS Sagemaker

### **Database:** SQL Server, PostgreSQL, MongoDB

### **Software Development Models:** MVC, Waterfall, V-Model, Incremental Model, Iterative Model

### **Technologies & Tools:** Git, Docker, RESTful APIs, .NET, FastAPI, Flask

### 

### 

### 

### **1. Kiến thức cốt lõi về AI và Machine Learning**

#### **Kiến thức lý thuyết:**

* **Machine Learning**: Hồi quy, phân loại, clustering, boosting, bagging, ensemble learning, k-fold cross-validation, hyperparameter tuning.
* **Deep Learning**: Neural Networks, CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), LSTM, GRU.
* **NLP (Natural Language Processing)**: Tokenization, stemming, lemmatization, Word2Vec, GloVe, attention mechanism, seq2seq.
* **NLU (Natural Language Understanding)**: Semantic understanding, intent detection.
* **NER (Named Entity Recognition)**: Trích xuất thông tin thực thể từ văn bản.
* **Transfer Learning**: Ứng dụng các mô hình pre-trained (GPT, BERT, T5).
* **LLMs (Large Language Models)**: Hiểu cách hoạt động và áp dụng các mô hình lớn như GPT, BERT, T5, Stable Diffusion, Whisper.

#### **Thực hành với thư viện và framework:**

* **Framework AI**: TensorFlow, PyTorch, JAX.
* **NLP Framework**: Huggingface, spaCy, NLTK, transformers.
* **Deep Learning Tools**: TensorRT, ONNX Runtime, CUDA toolkit.

### **2. Kiến thức về xử lý và phân tích dữ liệu**

#### **Công cụ và thư viện xử lý dữ liệu:**

* **Python Libraries**: Pandas, NumPy, Scikit-learn.
* **Data Visualization**: Matplotlib, Seaborn, Plotly, Dash.
* **Data Wrangling**: Xử lý, làm sạch và chuyển đổi dữ liệu.

#### **Hệ thống cơ sở dữ liệu:**

* **SQL**: SQL Server, PostgreSQL, SQLite.
* **NoSQL**: MongoDB, Firebase.

#### **Khả năng lập trình và làm việc với dữ liệu:**

* **Ngôn ngữ lập trình**: Python, R, Java, C#.
* **Công cụ lập trình**: Jupyter Notebook, Google Colab, VS Code, PyCharm.

### **3. Kiến thức về MLOps và DevOps**

#### **Hiểu biết về MLOps:**

* Tích hợp liên tục và triển khai (CI/CD) cho pipeline AI.
* Triển khai mô hình quy mô lớn bằng **MLflow**, **Kubeflow**, hoặc **TFX (TensorFlow Extended)**.
* Theo dõi và giám sát mô hình AI bằng **Prometheus**, **Grafana**.

#### **DevOps và Containerization:**

* **Containerization**: Docker, Kubernetes.
* **Công cụ DevOps**: Jenkins, GitLab CI/CD, ArgoCD.

### **4. Kiến thức về triển khai và quản lý hệ thống**

#### **Hệ điều hành và nền tảng đám mây:**

* **Linux**: Bash scripting, quản lý file hệ thống, quyền user.
* **Cloud Platforms**: AWS (S3, EC2, SageMaker), Google Cloud (Vertex AI, BigQuery), Azure (Azure ML).

#### **Phát triển và triển khai dịch vụ:**

* **API Development**: FastAPI, Flask, Django.
* **REST APIs và Web Services**: Tích hợp và phát triển các dịch vụ web.

### **5. Kỹ năng đánh giá mô hình**

* **Các phép đo hiệu suất mô hình**:
  + **RMSE (Root Mean Square Error)**: Sai số căn bậc hai trung bình.
  + **MAE (Mean Absolute Error)**: Sai số tuyệt đối trung bình.
  + **Accuracy**: Độ chính xác.
  + **Precision**: Độ chính xác của dự đoán.
  + **Recall**: Khả năng tìm ra các giá trị đúng thực tế.
  + **ROC (Receiver Operating Characteristic)** và **AUC (Area Under Curve)**: Đánh giá mô hình phân loại.

### **6. Ứng dụng AI chuyên sâu**

#### **Computer Vision:**

* Image Classification, Object Detection (YOLO, Faster R-CNN), Segmentation (U-Net).
* OpenCV, torchvision.

#### **Generative AI:**

* Generative Models: GAN (Generative Adversarial Networks), VAEs (Variational AutoEncoders).
* **LangChain**, **LLamaIndex**: Tích hợp AI vào ứng dụng thực tế.

#### **Fine-tuning mô hình:**

* Kinh nghiệm tùy chỉnh và fine-tuning mô hình AI cho các ứng dụng cụ thể.

### **7. Kỹ năng mềm và tư duy quan trọng**

* Tư duy phân tích và giải quyết vấn đề.
* Kỹ năng làm việc nhóm và giao tiếp tốt.
* Khả năng quản lý thời gian và tổ chức công việc.

AI-Project

## **Basic Projects (1-7)**

These projects are foundational to help you understand core concepts.

### **1. Basic Data Augmentation Tool**

* **Goal:** Implement a script to augment images by rotating, flipping, and adding noise.
* **Skills:** Image pre-processing, NumPy, OpenCV.
* **Framework:** Python (Pillow or OpenCV).

### **2. Autoencoder for Dimensionality Reduction**

* **Goal:** Build an autoencoder to reduce the dimensions of the MNIST dataset and reconstruct the images.
* **Skills:** Neural networks, encoding/decoding.
* **Framework:** TensorFlow/Keras or PyTorch.

### **3. Image Denoising with Autoencoders**

* **Goal:** Train an autoencoder to remove noise from images.
* **Dataset:** Add Gaussian noise to MNIST images.
* **Skills:** Data processing, autoencoders.
* **Framework:** PyTorch or TensorFlow.

### **4. Text Generation using RNN**

* **Goal:** Train a simple Recurrent Neural Network (RNN) to generate text in the style of a given dataset (e.g., Shakespeare’s works).
* **Skills:** Sequence modeling.
* **Framework:** TensorFlow/Keras or PyTorch.

### **5. Variational Autoencoder (VAE)**

* **Goal:** Build a VAE to generate new images from the MNIST dataset.
* **Skills:** Latent space, probabilistic models.
* **Framework:** TensorFlow/Keras or PyTorch.

### **6. GAN for Image Generation (Simple GAN)**

* **Goal:** Create a basic GAN to generate new handwritten digits from the MNIST dataset.
* **Skills:** Adversarial training.
* **Framework:** TensorFlow/Keras or PyTorch.

### **7. Neural Style Transfer**

* **Goal:** Implement style transfer to apply the style of a painting to a photograph.
* **Skills:** Feature extraction, content/style representation.
* **Framework:** TensorFlow/Keras.

## **Intermediate Projects (8-14)**

These projects involve more advanced generative techniques.

### **8. Conditional GAN (cGAN)**

* **Goal:** Create a cGAN to generate images conditioned on class labels (e.g., numbers in MNIST).
* **Skills:** Conditional models, label embeddings.
* **Framework:** TensorFlow/Keras or PyTorch.

### **9. CycleGAN for Image-to-Image Translation**

* **Goal:** Implement CycleGAN to translate images between two domains (e.g., horses ↔ zebras).
* **Skills:** Image translation, adversarial loss.
* **Framework:** PyTorch or TensorFlow.

### **10. Music Generation with RNN/LSTM**

* **Goal:** Generate simple melodies by training an RNN or LSTM model on MIDI files.
* **Skills:** Sequence modeling, time-series data.
* **Framework:** PyTorch or TensorFlow.

### **11. Text-to-Image Generation with DALL-E Mini**

* **Goal:** Use a pre-trained DALL-E Mini model to generate images based on textual prompts.
* **Skills:** Text-image mapping, pre-trained models.
* **Framework:** Hugging Face Transformers.

### **12. Image Captioning**

* **Goal:** Build a model that generates captions for images by combining CNNs (for image features) and RNNs (for text generation).
* **Skills:** Multimodal learning.
* **Framework:** PyTorch or TensorFlow.

### **13. DeepFake Video Generation**

* **Goal:** Train a model to swap faces in videos using deepfake techniques.
* **Skills:** Video processing, face alignment.
* **Framework:** PyTorch or TensorFlow.

### **14. Speech Synthesis (TTS)**

* **Goal:** Build a text-to-speech model using Tacotron or WaveNet.
* **Skills:** Audio data processing, sequence modeling.
* **Framework:** PyTorch.

## **Advanced Projects (15-20)**

These projects involve state-of-the-art techniques and real-world applications.

### **15. Fine-Tune GPT for Specific Text Tasks**

* **Goal:** Fine-tune GPT or GPT-2 for a specific text-generation task (e.g., customer service replies).
* **Skills:** NLP fine-tuning, Hugging Face.
* **Framework:** Hugging Face Transformers.

### **16. Build a Healthcare Chatbot**

* **Goal:** Train a chatbot to assist patients with symptom checks and basic health advice.
* **Skills:** NLP, knowledge graphs.
* **Framework:** Hugging Face, Rasa.

### **17. Multimodal Generative AI (CLIP + GAN)**

* **Goal:** Combine CLIP (text-image matching) with GANs to generate images matching a textual description.
* **Skills:** Cross-modal learning.
* **Framework:** PyTorch, Hugging Face.

### **18. Text Summarization with Transformers**

* **Goal:** Build a model to summarize long documents using a pre-trained transformer (e.g., BART).
* **Skills:** NLP, fine-tuning.
* **Framework:** Hugging Face.

### **19. Video Generation from Text**

* **Goal:** Generate short video clips based on textual descriptions (e.g., "a dog running").
* **Skills:** GANs, video modeling.
* **Framework:** PyTorch, RunwayML.

### **20. Build Your Own DALL-E/Stable Diffusion**

* **Goal:** Train or fine-tune a diffusion model to generate high-quality images from text prompts.
* **Skills:** Diffusion models, advanced model training.
* **Framework:** PyTorch, Hugging Face Diffusers.

## **General Tools and Tips**

* **Cloud Resources:** Use Google Colab, Kaggle, or AWS to leverage GPUs for training.
* **Data Sources:**
  + Kaggle Datasets
  + [ImageNet](https://image-net.org/)
  + [Common Voice](https://commonvoice.mozilla.org/en) for audio projects.
* **Version Control:** Use Git for managing project repositories.
* **Documentation:** Document each project to prepare for your portfolio.

## **Basic Projects (1-7)**

### **1. Random Sentence Generator**

* **Goal:** Generate random sentences by training a simple Markov Chain on text data.
* **Dataset:** News articles or books.
* **Skills:** NLP, text generation.
* **Framework:** Python.

### **2. Colorization of Grayscale Images**

* **Goal:** Train a model to add realistic colors to black-and-white images.
* **Dataset:** CIFAR-10 or old black-and-white photos.
* **Skills:** CNNs, image processing.
* **Framework:** TensorFlow/Keras or PyTorch.

### **3. Emoji Suggestion Tool**

* **Goal:** Train an NLP model to suggest relevant emojis for a given sentence.
* **Dataset:** Twitter or social media posts.
* **Skills:** Sequence-to-sequence modeling.
* **Framework:** Hugging Face Transformers.

### **4. Poetry Generation**

* **Goal:** Train a simple LSTM or RNN to generate poetry.
* **Dataset:** Shakespeare, poems from public domains.
* **Skills:** Text generation, sequence modeling.
* **Framework:** TensorFlow/Keras or PyTorch.

### **5. Dynamic Background Generation**

* **Goal:** Generate unique patterns or abstract backgrounds using GANs.
* **Dataset:** Patterns dataset (or create synthetic ones).
* **Skills:** GANs, image generation.
* **Framework:** TensorFlow/Keras or PyTorch.

### **6. Synthetic Handwriting Generation**

* **Goal:** Train a model to generate realistic handwriting styles.
* **Dataset:** IAM Handwriting Database.
* **Skills:** GANs, conditional generation.
* **Framework:** TensorFlow or PyTorch.

### **7. Image Super-Resolution**

* **Goal:** Use a Super-Resolution GAN (SRGAN) to enhance low-resolution images.
* **Dataset:** DIV2K (high-resolution images).
* **Skills:** Image enhancement, deep learning.
* **Framework:** PyTorch or TensorFlow.

## **Intermediate Projects (8-14)**

### **8. Synthetic Voice Cloning**

* **Goal:** Clone a voice using a small dataset of recorded audio.
* **Dataset:** LibriSpeech or a custom dataset.
* **Skills:** Audio processing, deep learning.
* **Framework:** PyTorch (Tacotron 2).

### **9. Time Series Data Generation**

* **Goal:** Generate synthetic time-series data (e.g., stock prices or weather patterns).
* **Skills:** LSTMs, temporal data modeling.
* **Framework:** TensorFlow or PyTorch.

### **10. Data-to-Text Generation**

* **Goal:** Build a model that generates textual descriptions of tabular data (e.g., weather reports).
* **Dataset:** Weather or sports data.
* **Skills:** NLP, data representation.
* **Framework:** Hugging Face Transformers.

### **11. StyleGAN for Face Generation**

* **Goal:** Use StyleGAN to generate realistic human faces.
* **Dataset:** CelebA.
* **Skills:** Advanced GANs, pre-trained models.
* **Framework:** PyTorch.

### **12. Personalized Meme Generator**

* **Goal:** Create a meme generator that overlays text on images using a text description.
* **Skills:** Image-text combination, NLP.
* **Framework:** Python (Pillow) and PyTorch.

### **13. AI-Powered Fashion Design**

* **Goal:** Use GANs to generate new clothing designs based on existing fashion datasets.
* **Dataset:** DeepFashion.
* **Skills:** Image generation, GANs.
* **Framework:** TensorFlow or PyTorch.

### **14. Personalized Music Generation**

* **Goal:** Generate personalized music tracks based on user preferences.
* **Skills:** Sequence modeling, music theory.
* **Framework:** PyTorch.

## **Advanced Projects (15-20)**

### **15. 3D Object Generation**

* **Goal:** Use 3D GANs to generate 3D models of objects like chairs or cars.
* **Dataset:** ShapeNet.
* **Skills:** 3D modeling, GANs.
* **Framework:** PyTorch.

### **16. Fine-Tune Stable Diffusion for Custom Art**

* **Goal:** Fine-tune Stable Diffusion to generate art based on a specific style (e.g., anime or abstract art).
* **Dataset:** Custom dataset.
* **Skills:** Diffusion models, fine-tuning.
* **Framework:** Hugging Face Diffusers.

### **17. Explainable Generative AI**

* **Goal:** Build a model that explains why certain outputs were generated (e.g., interpretable text generation).
* **Skills:** Model interpretability, NLP.
* **Framework:** Hugging Face Transformers.

### **18. Multi-Speaker Speech Synthesis**

* **Goal:** Generate synthetic speech for multiple speakers.
* **Dataset:** VoxCeleb or custom dataset.
* **Skills:** Audio processing, multi-speaker embeddings.
* **Framework:** PyTorch or TensorFlow.

### **19. Video Caption Generation**

* **Goal:** Generate descriptive captions for short video clips.
* **Dataset:** MSR-VTT.
* **Skills:** Multimodal learning, image-to-text.
* **Framework:** PyTorch or TensorFlow.

### **20. AI-Assisted Comic Creation**

* **Goal:** Build a tool to generate comic strips with AI-created characters, dialogues, and backgrounds.
* **Dataset:** Manga109 or similar datasets.
* **Skills:** Multimodal generation.
* **Framework:** PyTorch or TensorFlow.

## **Additional Resources**

1. **Datasets:**
   * [Kaggle Datasets](https://www.kaggle.com/)
   * TensorFlow Datasets
2. **Frameworks:**
   * PyTorch, TensorFlow, Hugging Face.
3. **Cloud Tools:**
   * Google Colab, AWS, or Azure for running experiments.

Để học sâu về Generative AI, dưới đây là lộ trình và các dự án từ cơ bản đến nâng cao:

## **1. Nền tảng kiến thức**

### **Kiến thức cơ bản (1-2 tháng):**

1. **Toán học:**
   * Đại số tuyến tính (ma trận, vector, phép biến đổi).
   * Xác suất và thống kê (phân phối, biến ngẫu nhiên, Bayes).
   * Giải tích (đạo hàm, gradient descent).
2. **Lập trình:**
   * Thành thạo Python.
   * Hiểu rõ NumPy, Pandas, Matplotlib, Seaborn.
3. **Học sâu:**
   * Hiểu về mạng neuron, gradient descent, hàm mất mát.
   * Thử nghiệm với TensorFlow và PyTorch.

### **2. Học lý thuyết Generative AI (2-3 tháng):**

1. **Generative Models:**
   * Hiểu cơ bản về Autoencoders, Variational Autoencoders (VAE).
   * Nắm vững về GANs (Generative Adversarial Networks).
   * Các mô hình phổ biến: DCGAN, CGAN, WGAN.
2. **Transformer-based Models:**
   * Hiểu kiến trúc Transformer.
   * Học về GPT (Generative Pre-trained Transformer) và BERT.
3. **Diffusion Models (Nâng cao):**
   * Nghiên cứu Stable Diffusion, Denoising Diffusion Probabilistic Models (DDPM).

### **3. Xây dựng dự án**

#### **Cơ bản:**

1. **Dự án: Tạo ảnh từ tiếng nói (Simple GANs)**
   * Mô tả: Xây dựng GAN đơn giản để sinh ảnh từ bộ dữ liệu MNIST hoặc CIFAR-10.
   * Kỹ năng: Hiểu cách đào tạo GAN và cải thiện kết quả.
2. **Dự án: Sinh văn bản tự động (GPT nhỏ)**
   * Mô tả: Sử dụng GPT-2 để tạo văn bản dựa trên một chủ đề.
   * Bộ công cụ: Hugging Face Transformers.

#### **Trung cấp:**

1. **Dự án: Biến đổi phong cách ảnh (Style Transfer)**
   * Mô tả: Sử dụng VAE hoặc GAN để chuyển phong cách giữa hai ảnh (VD: Van Gogh style).
   * Thư viện: PyTorch, TensorFlow.
2. **Dự án: Chatbot sử dụng GPT**
   * Mô tả: Xây dựng chatbot có khả năng trả lời câu hỏi tự nhiên bằng cách tùy chỉnh GPT-3 hoặc GPT-4.

#### **Nâng cao:**

1. **Dự án: Sinh ảnh với Diffusion Models**
   * Mô tả: Huấn luyện mô hình sinh ảnh từ đầu hoặc tinh chỉnh mô hình sẵn có như Stable Diffusion.
2. **Dự án: Tùy chỉnh Generative AI cho ứng dụng thực tế**
   * Mô tả: Xây dựng mô hình AI sinh hình ảnh, văn bản hoặc nhạc, ứng dụng cho sản phẩm thực tế (VD: game, quảng cáo, y tế).
   * Ví dụ: Ứng dụng tạo tài liệu y tế cho chatbot trong lĩnh vực sức khỏe.

### **4. Tài nguyên học tập**

* **Sách:**
  + "Deep Learning" - Ian Goodfellow.
  + "GANs in Action" - Jakub Langr & Vladimir Bok.
* **Khóa học:**
  + [Deep Learning Specialization](https://www.coursera.org/specializations/deep-learning) của Andrew Ng.
  + GANs Specialization của DeepLearning.AI.
  + Hugging Face Course: Transformers & NLP.
* **Cộng đồng:**
  + Tham gia các cộng đồng như Kaggle, Hugging Face, Reddit (r/MachineLearning).

Nếu bạn cần chi tiết hoặc triển khai một dự án cụ thể, nói mình biết nhé!

4o

##### **You said:**

for me more

###### **ChatGPT said:**

ChatGPT

Để học sâu hơn về Generative AI và mở rộng kỹ năng, dưới đây là các bước bổ sung, tài nguyên, và dự án nâng cao hơn:

## **5. Kiến thức nâng cao và chuyên sâu**

### **Lý thuyết nâng cao (2-3 tháng):**

1. **Advanced Generative Techniques:**
   * Energy-Based Models (EBMs).
   * Flow-based Generative Models (Normalizing Flows).
   * Score-based Generative Models.
2. **Self-Supervised Learning (SSL):**
   * Hiểu cơ bản về cách các mô hình học không cần nhãn (Contrastive Learning, BYOL, SimCLR).
3. **Large Language Models (LLMs):**
   * Nghiên cứu cách GPT-3.5, GPT-4 hoặc ChatGPT được huấn luyện.
   * Tìm hiểu thêm về Reinforcement Learning from Human Feedback (RLHF).
4. **Optimization Techniques:**
   * Kỹ thuật ổn định khi đào tạo GAN: Gradient Penalty, Spectral Normalization.
   * Sử dụng mixed precision và distributed training.

### **6. Dự án nâng cao**

#### **Dự án trung cấp – nâng cao:**

1. **Sinh mã nguồn tự động (Code Generation)**
   * Mô tả: Xây dựng AI sinh code từ mô tả tự nhiên (dựa trên GPT-3 Codex hoặc mô hình tùy chỉnh).
   * Kỹ năng: Fine-tune trên tập dữ liệu code Python hoặc JavaScript.
   * Ứng dụng: Hỗ trợ lập trình viên.
2. **Tùy chỉnh mô hình GPT (Fine-tuning LLMs):**
   * Mô tả: Thu thập dữ liệu tùy chỉnh (VD: hội thoại, tài liệu nội bộ) và fine-tune GPT trên tập dữ liệu này.
   * Bộ công cụ: Hugging Face Transformers, OpenAI API.
3. **Tạo nhạc tự động (Music Generation):**
   * Mô tả: Xây dựng mô hình sinh nhạc sử dụng RNN hoặc Transformer như MuseNet, Jukebox.
   * Thư viện: Magenta, PyTorch.

#### **Dự án đột phá:**

1. **AI tổng hợp đa phương thức (Multimodal Generative AI):**
   * Mô tả: Xây dựng mô hình xử lý cả văn bản, hình ảnh và âm thanh (VD: DALL-E, CLIP).
   * Ứng dụng: Tạo ra nội dung marketing, nội dung sáng tạo đa dạng.
2. **Tích hợp Generative AI vào hệ thống thực tế:**
   * Mô tả: Xây dựng ứng dụng Generative AI từ đầu đến cuối (backend, frontend, và tích hợp AI).
   * Công nghệ: ReactJS, .NET, Docker, Kubernetes.
   * Ví dụ: Nền tảng sinh quảng cáo tự động hoặc thiết kế website từ mô tả.

### **7. Thử thách thực tế**

1. **Kaggle Competitions:**
   * Tham gia các cuộc thi về generative tasks như sinh ảnh hoặc tối ưu hóa mô hình AI.
2. **Open-Source Contributions:**
   * Đóng góp vào dự án mã nguồn mở như Hugging Face, PyTorch Lightning, hoặc các mô hình GPT nhỏ.
3. **Hackathons:**
   * Tìm các hackathon AI hoặc Generative AI (kết hợp với startup hoặc công ty lớn).

### **8. Tài nguyên chuyên sâu**

* **Sách:**
  + "Transformers for Natural Language Processing" - Denis Rothman.
  + "Deep Reinforcement Learning Hands-On" - Maxim Lapan.
* **Khóa học:**
  + Transformers Specialization.
  + [FastAI's Deep Learning Course](https://course.fast.ai/).
* **Nghiên cứu:**
  + Đọc các bài báo trên arXiv về generative models.
  + Theo dõi các hội thảo: NeurIPS, CVPR, ACL.

### **9. Mở rộng: Tích hợp Generative AI với các ngành**

1. **Y tế:**
   * Sinh dữ liệu giả lập cho đào tạo y tế.
   * Chatbot tư vấn sức khỏe.
2. **Thương mại điện tử:**
   * Sinh nội dung tự động cho sản phẩm.
   * Tạo mẫu thiết kế sản phẩm.
3. **Sáng tạo:**
   * Dự án nghệ thuật AI.
   * Sinh video tự động (sử dụng RunwayML).

Cần thêm chi tiết cho từng dự án hay muốn mình hướng dẫn cách triển khai, cứ hỏi thêm nhé! 👨‍💻

4o

##### **You said:**

for me more and more details of each phrase

###### **ChatGPT said:**

ChatGPT

Dưới đây là **chi tiết cụ thể hơn cho từng giai đoạn** trong lộ trình học Generative AI, kèm theo hướng dẫn triển khai, tài nguyên, và các mẹo thực tế:

## **1. Nền tảng kiến thức (1-2 tháng)**

### **1.1 Toán học**

* **Đại số tuyến tính:**
  + Học các phép toán với ma trận: nhân, nghịch đảo, định thức.
  + Áp dụng: Tìm hiểu cách ma trận được sử dụng để biểu diễn trọng số trong mạng neuron.
  + Tài nguyên: Khóa học [Linear Algebra](https://www.khanacademy.org/math/linear-algebra).
* **Xác suất và Thống kê:**
  + Hiểu phân phối xác suất: Gaussian, Bernoulli.
  + Học định lý Bayes, áp dụng vào bài toán xác suất có điều kiện.
  + Tài nguyên: Sách "Think Stats" - Allen B. Downey.
* **Giải tích:**
  + Tập trung vào đạo hàm và gradient descent.
  + Thực hành: Viết thuật toán gradient descent bằng Python.

### **1.2 Lập trình**

* **Python:**
  + Tập trung vào cấu trúc dữ liệu (list, dict, tuple) và lập trình hướng đối tượng.
  + Bài tập thực hành: Viết các hàm tính đạo hàm hoặc phép biến đổi ma trận.
* **Thư viện:**
  + **NumPy:** Xử lý dữ liệu dạng ma trận.
  + **Pandas:** Làm sạch và xử lý dữ liệu.
  + **Matplotlib:** Trực quan hóa dữ liệu.

## **2. Lý thuyết về Generative AI (2-3 tháng)**

### **2.1 Autoencoders (AEs):**

* **Nguyên lý:**
  + Autoencoders học cách mã hóa dữ liệu vào không gian ẩn (latent space) và giải mã nó.
  + Ứng dụng: Giảm chiều dữ liệu, phát hiện bất thường.
* **Thực hành:**
  + Dùng bộ dữ liệu MNIST, xây dựng một Autoencoder cơ bản để nén và tái tạo ảnh.
  + Bộ công cụ: TensorFlow, Keras.

### **2.2 Variational Autoencoders (VAEs):**

* **Nguyên lý:**
  + VAEs sử dụng xác suất để sinh dữ liệu mới từ không gian ẩn.
  + Phân phối Gaussian được dùng để tạo mẫu dữ liệu.
* **Thực hành:**
  + Sinh ảnh mới từ không gian ẩn bằng VAEs.
* **Tài nguyên:** Blog [VAEs Explained](https://towardsdatascience.com/).

### **2.3 Generative Adversarial Networks (GANs):**

* **Nguyên lý:**
  + Gồm hai mạng: Generator (sinh dữ liệu) và Discriminator (phân biệt thật/giả).
  + Hai mạng huấn luyện đối kháng để cải thiện chất lượng dữ liệu sinh.
* **Thực hành:**
  + Dự án: Sinh ảnh từ bộ dữ liệu CIFAR-10.
  + Cải tiến: Thêm Spectral Normalization hoặc Gradient Penalty để ổn định mô hình.

## **3. Xây dựng dự án (Cơ bản → Nâng cao)**

### **3.1 Dự án cơ bản: Tạo ảnh từ GAN**

* **Bước 1: Chuẩn bị dữ liệu**
  + Sử dụng dataset như MNIST hoặc CIFAR-10.
  + Thực hiện tiền xử lý: chuẩn hóa ảnh, tăng cường dữ liệu.
* **Bước 2: Xây dựng GAN**
  + Tạo hai mạng riêng biệt: Generator và Discriminator.
  + Tối ưu hóa qua hàm mất mát Binary Cross Entropy (BCE).
* **Bước 3: Đánh giá**
  + Sử dụng Frechet Inception Distance (FID) để đo chất lượng ảnh sinh.

### **3.2 Dự án trung cấp: Style Transfer**

* **Nguyên lý:**
  + Chuyển phong cách của một ảnh nguồn (style) sang ảnh đích (content).
  + Dùng VGG-19 để trích xuất đặc trưng.
* **Thực hành:**
  + Áp dụng thư viện PyTorch hoặc TensorFlow.
  + Tinh chỉnh tham số alpha (nội dung) và beta (phong cách).

### **3.3 Dự án nâng cao: Multimodal AI**

* **Mô tả:**
  + Xây dựng AI xử lý đồng thời văn bản và hình ảnh, ví dụ:
    - Nhập mô tả văn bản → Sinh ảnh (DALL-E).
    - Nhập ảnh → Sinh mô tả (CLIP).
* **Thực hành:**
  + Tích hợp OpenAI API hoặc tự xây dựng với các mô hình mở.

## **4. Tài nguyên chuyên sâu**

### **Sách:**

1. **"Deep Learning"** - Ian Goodfellow: Đọc chương về GANs.
2. **"Generative Deep Learning"** - David Foster: Tập trung vào VAEs, GANs.

### **Blog và hội thảo:**

* Blog của Andrej Karpathy.
* Hội thảo NeurIPS, ICCV.

Nếu bạn cần chi tiết code hoặc hướng dẫn từng bước, mình có thể hướng dẫn ngay nhé!

4o

##### **You said:**

for me more

###### **ChatGPT said:**

ChatGPT

Dưới đây là bản mở rộng và chi tiết hơn cho từng giai đoạn trong lộ trình học Generative AI, với các bước cụ thể và thực hành để bạn hiểu sâu hơn.

## **1. Xây dựng nền tảng mạnh mẽ**

### **1.1 Toán học chi tiết**

1. **Đại số tuyến tính:**
   * **Học kiến thức cơ bản:** Vector, ma trận, phép biến đổi, không gian vector.
   * **Thực hành:**
     + Viết một hàm Python để tính tích ma trận.
     + Hiểu cách trọng số trong mạng neuron là các ma trận được cập nhật thông qua gradient descent.
   * **Công cụ:** Sử dụng NumPy để thực hành các phép toán ma trận.
2. **Giải tích:**
   * **Tập trung:** Đạo hàm riêng, gradient và cách tối ưu hóa hàm mất mát.
   * **Thực hành:**
     + Viết code tính gradient descent bằng Python.
     + Vẽ đồ thị hàm mất mát theo số epoch để hiểu sự hội tụ.
3. **Xác suất và Thống kê:**
   * **Tập trung:** Xác suất có điều kiện, phân phối chuẩn, định lý Bayes.
   * **Thực hành:**
     + Sinh dữ liệu ngẫu nhiên bằng NumPy.
     + Viết mô phỏng Monte Carlo đơn giản.

### **1.2 Lập trình chuyên sâu**

1. **Python:**
   * **Mục tiêu:** Thành thạo các kỹ thuật xử lý dữ liệu, lập trình hướng đối tượng.
   * **Thực hành:**
     + Xử lý và hình ảnh hóa dữ liệu lớn (tập MNIST, CIFAR-10).
     + Viết các class Python cho các thành phần của mô hình học sâu.
2. **Thư viện:**
   * **NumPy & Pandas:** Tăng cường xử lý dữ liệu.
   * **Matplotlib & Seaborn:** Vẽ biểu đồ minh họa quá trình đào tạo.
3. **Thực hành coding:**
   * Xây dựng mạng neuron cơ bản bằng NumPy (không dùng framework như PyTorch).

## **2. Học lý thuyết Generative AI chi tiết**

### **2.1 Autoencoders (AEs)**

1. **Kiến thức:**
   * Autoencoders gồm hai phần:
     + **Encoder:** Mã hóa dữ liệu đầu vào vào không gian ẩn (latent space).
     + **Decoder:** Giải mã không gian ẩn để tái tạo dữ liệu.
   * Áp dụng: Giảm chiều dữ liệu, phát hiện bất thường.
2. **Thực hành:**
   * Bộ dữ liệu: MNIST.
   * Xây dựng mô hình:
     + Sử dụng Dense Layer trong TensorFlow/Keras để tạo encoder và decoder.
     + Dùng hàm mất mát MSE để đo lường độ chênh lệch giữa dữ liệu đầu vào và đầu ra.
   * **Phân tích:** Quan sát không gian ẩn bằng cách vẽ biểu đồ tán xạ (scatter plot).

### **2.2 Variational Autoencoders (VAEs)**

1. **Kiến thức:**
   * VAEs mở rộng từ Autoencoders bằng cách áp dụng xác suất.
   * Không gian ẩn được mô hình hóa bằng phân phối Gaussian.
2. **Thực hành:**
   * **Tập dữ liệu:** CIFAR-10 hoặc ảnh đơn giản.
   * **Xây dựng mô hình:**
     + Tích hợp Reparameterization Trick trong không gian ẩn.
     + Tối ưu hóa hàm mất mát gồm 2 phần: Reconstruction Loss và KL Divergence.

### **2.3 Generative Adversarial Networks (GANs)**

1. **Kiến thức:**
   * GANs gồm:
     + Generator: Sinh dữ liệu giả.
     + Discriminator: Phân biệt dữ liệu thật/giả.
   * Hai mạng đấu tranh để cải thiện lẫn nhau.
2. **Thực hành chi tiết:**
   * **Tập dữ liệu:** CIFAR-10, Anime Faces.
   * **Xây dựng GAN cơ bản:**
     + Generator: Tạo ảnh từ nhiễu ngẫu nhiên (random noise).
     + Discriminator: Phân loại ảnh thật/giả.
   * **Cải tiến:**
     + Thử nghiệm với Gradient Penalty để ổn định.
     + Tính Frechet Inception Distance (FID) để đo chất lượng ảnh.

## **3. Dự án thực tế từ cơ bản đến nâng cao**

### **3.1 Cơ bản: Sinh ảnh từ GAN**

1. **Mục tiêu:** Xây dựng GAN đơn giản để sinh ảnh từ bộ dữ liệu MNIST.
2. **Thực hành:**
   * Sử dụng TensorFlow hoặc PyTorch.
   * Quan sát quá trình sinh ảnh qua từng epoch.

### **3.2 Trung cấp: Style Transfer**

1. **Mục tiêu:** Biến đổi phong cách của một bức ảnh (VD: chuyển ảnh chụp sang phong cách tranh Van Gogh).
2. **Thực hành:**
   * Sử dụng mô hình VGG-19 để trích xuất đặc trưng.
   * Huấn luyện mô hình chuyển phong cách trên bộ dữ liệu tự thu thập.

### **3.3 Nâng cao: Generative AI đa phương thức**

1. **Mục tiêu:** Tích hợp mô hình xử lý cả văn bản và hình ảnh.
2. **Ý tưởng:**
   * Sinh ảnh từ mô tả văn bản (sử dụng DALL-E).
   * Mô tả ảnh đầu vào bằng văn bản (sử dụng CLIP).

## **4. Tài nguyên và mẹo học nâng cao**

### **4.1 Tài liệu học tập:**

1. **Generative Deep Learning** - David Foster: Chương trình chi tiết về GANs, VAEs.
2. **Deep Learning** - Ian Goodfellow: Toàn diện về lý thuyết học sâu.

### **4.2 Blog và công cụ:**

1. [Hugging Face Transformers](https://huggingface.co/): Cộng đồng mô hình NLP.
2. [Kaggle](https://www.kaggle.com/): Dữ liệu và notebook thực hành.

Nếu cần chi tiết về bất kỳ bước hoặc dự án nào, hãy nói nhé! Mình có thể viết hướng dẫn hoặc mã mẫu ngay. 🚀

4o

##### **You said:**

I want to do project for me some project to do from basic to advance about 20 projects

###### **ChatGPT said:**

ChatGPT

Here’s a curated list of **20 projects**, categorized from basic to advanced, to help you progressively learn and master **Generative AI**. Each project includes a brief description and possible tools/frameworks you can use.

## **Basic Projects (1-7)**

These projects are foundational to help you understand core concepts.

### **1. Basic Data Augmentation Tool**

* **Goal:** Implement a script to augment images by rotating, flipping, and adding noise.
* **Skills:** Image pre-processing, NumPy, OpenCV.
* **Framework:** Python (Pillow or OpenCV).

### **2. Autoencoder for Dimensionality Reduction**

* **Goal:** Build an autoencoder to reduce the dimensions of the MNIST dataset and reconstruct the images.
* **Skills:** Neural networks, encoding/decoding.
* **Framework:** TensorFlow/Keras or PyTorch.

### **3. Image Denoising with Autoencoders**

* **Goal:** Train an autoencoder to remove noise from images.
* **Dataset:** Add Gaussian noise to MNIST images.
* **Skills:** Data processing, autoencoders.
* **Framework:** PyTorch or TensorFlow.

### **4. Text Generation using RNN**

* **Goal:** Train a simple Recurrent Neural Network (RNN) to generate text in the style of a given dataset (e.g., Shakespeare’s works).
* **Skills:** Sequence modeling.
* **Framework:** TensorFlow/Keras or PyTorch.

### **5. Variational Autoencoder (VAE)**

* **Goal:** Build a VAE to generate new images from the MNIST dataset.
* **Skills:** Latent space, probabilistic models.
* **Framework:** TensorFlow/Keras or PyTorch.

### **6. GAN for Image Generation (Simple GAN)**

* **Goal:** Create a basic GAN to generate new handwritten digits from the MNIST dataset.
* **Skills:** Adversarial training.
* **Framework:** TensorFlow/Keras or PyTorch.

### **7. Neural Style Transfer**

* **Goal:** Implement style transfer to apply the style of a painting to a photograph.
* **Skills:** Feature extraction, content/style representation.
* **Framework:** TensorFlow/Keras.

## **Intermediate Projects (8-14)**

These projects involve more advanced generative techniques.

### **8. Conditional GAN (cGAN)**

* **Goal:** Create a cGAN to generate images conditioned on class labels (e.g., numbers in MNIST).
* **Skills:** Conditional models, label embeddings.
* **Framework:** TensorFlow/Keras or PyTorch.

### **9. CycleGAN for Image-to-Image Translation**

* **Goal:** Implement CycleGAN to translate images between two domains (e.g., horses ↔ zebras).
* **Skills:** Image translation, adversarial loss.
* **Framework:** PyTorch or TensorFlow.

### **10. Music Generation with RNN/LSTM**

* **Goal:** Generate simple melodies by training an RNN or LSTM model on MIDI files.
* **Skills:** Sequence modeling, time-series data.
* **Framework:** PyTorch or TensorFlow.

### **11. Text-to-Image Generation with DALL-E Mini**

* **Goal:** Use a pre-trained DALL-E Mini model to generate images based on textual prompts.
* **Skills:** Text-image mapping, pre-trained models.
* **Framework:** Hugging Face Transformers.

### **12. Image Captioning**

* **Goal:** Build a model that generates captions for images by combining CNNs (for image features) and RNNs (for text generation).
* **Skills:** Multimodal learning.
* **Framework:** PyTorch or TensorFlow.

### **13. DeepFake Video Generation**

* **Goal:** Train a model to swap faces in videos using deepfake techniques.
* **Skills:** Video processing, face alignment.
* **Framework:** PyTorch or TensorFlow.

### **14. Speech Synthesis (TTS)**

* **Goal:** Build a text-to-speech model using Tacotron or WaveNet.
* **Skills:** Audio data processing, sequence modeling.
* **Framework:** PyTorch.

## **Advanced Projects (15-20)**

These projects involve state-of-the-art techniques and real-world applications.

### **15. Fine-Tune GPT for Specific Text Tasks**

* **Goal:** Fine-tune GPT or GPT-2 for a specific text-generation task (e.g., customer service replies).
* **Skills:** NLP fine-tuning, Hugging Face.
* **Framework:** Hugging Face Transformers.

### **16. Build a Healthcare Chatbot**

* **Goal:** Train a chatbot to assist patients with symptom checks and basic health advice.
* **Skills:** NLP, knowledge graphs.
* **Framework:** Hugging Face, Rasa.

### **17. Multimodal Generative AI (CLIP + GAN)**

* **Goal:** Combine CLIP (text-image matching) with GANs to generate images matching a textual description.
* **Skills:** Cross-modal learning.
* **Framework:** PyTorch, Hugging Face.

### **18. Text Summarization with Transformers**

* **Goal:** Build a model to summarize long documents using a pre-trained transformer (e.g., BART).
* **Skills:** NLP, fine-tuning.
* **Framework:** Hugging Face.

### **19. Video Generation from Text**

* **Goal:** Generate short video clips based on textual descriptions (e.g., "a dog running").
* **Skills:** GANs, video modeling.
* **Framework:** PyTorch, RunwayML.

### **20. Build Your Own DALL-E/Stable Diffusion**

* **Goal:** Train or fine-tune a diffusion model to generate high-quality images from text prompts.
* **Skills:** Diffusion models, advanced model training.
* **Framework:** PyTorch, Hugging Face Diffusers.

## **General Tools and Tips**

* **Cloud Resources:** Use Google Colab, Kaggle, or AWS to leverage GPUs for training.
* **Data Sources:**
  + Kaggle Datasets
  + [ImageNet](https://image-net.org/)
  + [Common Voice](https://commonvoice.mozilla.org/en) for audio projects.
* **Version Control:** Use Git for managing project repositories.
* **Documentation:** Document each project to prepare for your portfolio.

If you want, I can provide **step-by-step implementation** or sample code for any of the above projects!

4o

##### **You said:**

for me more projects about 20 projects

###### **ChatGPT said:**

ChatGPT

Here are **20 additional Generative AI project ideas** ranging from basic to advanced, focusing on diverse applications and techniques. These will further enhance your skills in this field.

## **Basic Projects (1-7)**

### **1. Random Sentence Generator**

* **Goal:** Generate random sentences by training a simple Markov Chain on text data.
* **Dataset:** News articles or books.
* **Skills:** NLP, text generation.
* **Framework:** Python.

### **2. Colorization of Grayscale Images**

* **Goal:** Train a model to add realistic colors to black-and-white images.
* **Dataset:** CIFAR-10 or old black-and-white photos.
* **Skills:** CNNs, image processing.
* **Framework:** TensorFlow/Keras or PyTorch.

### **3. Emoji Suggestion Tool**

* **Goal:** Train an NLP model to suggest relevant emojis for a given sentence.
* **Dataset:** Twitter or social media posts.
* **Skills:** Sequence-to-sequence modeling.
* **Framework:** Hugging Face Transformers.

### **4. Poetry Generation**

* **Goal:** Train a simple LSTM or RNN to generate poetry.
* **Dataset:** Shakespeare, poems from public domains.
* **Skills:** Text generation, sequence modeling.
* **Framework:** TensorFlow/Keras or PyTorch.

### **5. Dynamic Background Generation**

* **Goal:** Generate unique patterns or abstract backgrounds using GANs.
* **Dataset:** Patterns dataset (or create synthetic ones).
* **Skills:** GANs, image generation.
* **Framework:** TensorFlow/Keras or PyTorch.

### **6. Synthetic Handwriting Generation**

* **Goal:** Train a model to generate realistic handwriting styles.
* **Dataset:** IAM Handwriting Database.
* **Skills:** GANs, conditional generation.
* **Framework:** TensorFlow or PyTorch.

### **7. Image Super-Resolution**

* **Goal:** Use a Super-Resolution GAN (SRGAN) to enhance low-resolution images.
* **Dataset:** DIV2K (high-resolution images).
* **Skills:** Image enhancement, deep learning.
* **Framework:** PyTorch or TensorFlow.

## **Intermediate Projects (8-14)**

### **8. Synthetic Voice Cloning**

* **Goal:** Clone a voice using a small dataset of recorded audio.
* **Dataset:** LibriSpeech or a custom dataset.
* **Skills:** Audio processing, deep learning.
* **Framework:** PyTorch (Tacotron 2).

### **9. Time Series Data Generation**

* **Goal:** Generate synthetic time-series data (e.g., stock prices or weather patterns).
* **Skills:** LSTMs, temporal data modeling.
* **Framework:** TensorFlow or PyTorch.

### **10. Data-to-Text Generation**

* **Goal:** Build a model that generates textual descriptions of tabular data (e.g., weather reports).
* **Dataset:** Weather or sports data.
* **Skills:** NLP, data representation.
* **Framework:** Hugging Face Transformers.

### **11. StyleGAN for Face Generation**

* **Goal:** Use StyleGAN to generate realistic human faces.
* **Dataset:** CelebA.
* **Skills:** Advanced GANs, pre-trained models.
* **Framework:** PyTorch.

### **12. Personalized Meme Generator**

* **Goal:** Create a meme generator that overlays text on images using a text description.
* **Skills:** Image-text combination, NLP.
* **Framework:** Python (Pillow) and PyTorch.

### **13. AI-Powered Fashion Design**

* **Goal:** Use GANs to generate new clothing designs based on existing fashion datasets.
* **Dataset:** DeepFashion.
* **Skills:** Image generation, GANs.
* **Framework:** TensorFlow or PyTorch.

### **14. Personalized Music Generation**

* **Goal:** Generate personalized music tracks based on user preferences.
* **Skills:** Sequence modeling, music theory.
* **Framework:** PyTorch.

## **Advanced Projects (15-20)**

### **15. 3D Object Generation**

* **Goal:** Use 3D GANs to generate 3D models of objects like chairs or cars.
* **Dataset:** ShapeNet.
* **Skills:** 3D modeling, GANs.
* **Framework:** PyTorch.

### **16. Fine-Tune Stable Diffusion for Custom Art**

* **Goal:** Fine-tune Stable Diffusion to generate art based on a specific style (e.g., anime or abstract art).
* **Dataset:** Custom dataset.
* **Skills:** Diffusion models, fine-tuning.
* **Framework:** Hugging Face Diffusers.

### **17. Explainable Generative AI**

* **Goal:** Build a model that explains why certain outputs were generated (e.g., interpretable text generation).
* **Skills:** Model interpretability, NLP.
* **Framework:** Hugging Face Transformers.

### **18. Multi-Speaker Speech Synthesis**

* **Goal:** Generate synthetic speech for multiple speakers.
* **Dataset:** VoxCeleb or custom dataset.
* **Skills:** Audio processing, multi-speaker embeddings.
* **Framework:** PyTorch or TensorFlow.

### **19. Video Caption Generation**

* **Goal:** Generate descriptive captions for short video clips.
* **Dataset:** MSR-VTT.
* **Skills:** Multimodal learning, image-to-text.
* **Framework:** PyTorch or TensorFlow.

### **20. AI-Assisted Comic Creation**

* **Goal:** Build a tool to generate comic strips with AI-created characters, dialogues, and backgrounds.
* **Dataset:** Manga109 or similar datasets.
* **Skills:** Multimodal generation.
* **Framework:** PyTorch or TensorFlow.

## **Additional Resources**

1. **Datasets:**
   * [Kaggle Datasets](https://www.kaggle.com/)
   * TensorFlow Datasets
2. **Frameworks:**
   * PyTorch, TensorFlow, Hugging Face.
3. **Cloud Tools:**
   * Google Colab, AWS, or Azure for running experiments.

If you'd like detailed guidance or implementation for any of these projects, let me know! 😊

4o

##### **You said:**

nếu mà tôi làm được những project trên thì có thể đi thực tập doanh nghiệp được chưa

###### **ChatGPT said:**

ChatGPT

Nếu bạn làm được tất cả những project đã liệt kê ở trên (cả basic, intermediate, và advanced), bạn không chỉ đủ khả năng đi thực tập mà còn có thể đặt mục tiêu vào những vị trí thực tập tại các doanh nghiệp công nghệ hàng đầu, đặc biệt là trong lĩnh vực AI.

Dưới đây là lý do tại sao:

### **1. Kỹ năng kỹ thuật được khẳng định**

* **Kỹ thuật lõi:** Bạn đã làm việc với các mô hình phổ biến như Autoencoders, GANs, RNNs, Transformers, và Diffusion Models. Đây là những công cụ quan trọng trong AI hiện đại.
* **Kinh nghiệm thực tế:** Hoàn thành các project như Image Super-Resolution, StyleGAN, hoặc Video Captioning chứng tỏ bạn không chỉ biết lý thuyết mà còn biết ứng dụng vào các bài toán thực tế.
* **Hiểu biết sâu:** Làm được các dự án nâng cao như Fine-tune Stable Diffusion hay AI-Assisted Comic Creation cho thấy bạn hiểu rõ về mô hình và biết cách điều chỉnh chúng cho các bài toán cụ thể.

### **2. Portfolio ấn tượng**

* **Đa dạng dự án:** Với khoảng 40 project, portfolio của bạn sẽ rất phong phú, thể hiện khả năng xử lý nhiều dạng bài toán khác nhau (text, image, audio, video).
* **Chất lượng sản phẩm:** Nếu các dự án của bạn có kết quả rõ ràng (ví dụ: hình ảnh tạo ra đẹp mắt, mô hình text-to-speech tự nhiên), thì đây sẽ là một điểm mạnh để gây ấn tượng với nhà tuyển dụng.
* **Demo cụ thể:** Đưa những sản phẩm này lên GitHub, viết blog hoặc tạo video giới thiệu sẽ khiến bạn nổi bật hơn nhiều ứng viên khác.

### **3. Kỹ năng làm việc nhóm**

* Khi làm các project lớn, như CycleGAN hoặc Video Generation, nếu bạn làm trong nhóm thì sẽ phát triển các kỹ năng mềm quan trọng:
  + **Version Control (Git):** Làm việc nhóm hiệu quả với Git.
  + **Communication:** Trao đổi ý tưởng và giải quyết vấn đề.
  + **Documentation:** Viết tài liệu chi tiết, chuyên nghiệp.

### **4. Đủ sức cạnh tranh vào thực tập doanh nghiệp**

* **Với các công ty lớn:** Nếu nhắm đến các công ty như Google, Meta, OpenAI, FPT AI, Zalo AI, bạn cần thêm kinh nghiệm giải quyết các bài toán cụ thể mà họ đang quan tâm. Tuy nhiên, nếu bạn làm được các dự án nâng cao, cơ hội sẽ rất rộng mở.
* **Với các startup:** Kỹ năng của bạn đủ mạnh để đảm nhận vị trí thực tập hoặc thậm chí là Junior AI Engineer. Các startup thường tìm kiếm người có khả năng triển khai nhanh, và với loạt project thực chiến, bạn sẽ phù hợp.

### **Bước tiếp theo để chuẩn bị đi thực tập**

1. **Tối ưu hóa CV:**
   * Chỉ ra các project nổi bật nhất (3-5 cái, nhấn mạnh vào kết quả và công nghệ dùng).
   * Liệt kê các kỹ năng kỹ thuật rõ ràng (Python, TensorFlow, PyTorch, Hugging Face, v.v.).
2. **Xây dựng hồ sơ online:**
   * Đăng project lên GitHub, thêm README chi tiết.
   * Viết blog chia sẻ về kinh nghiệm hoặc thuật toán bạn đã áp dụng.
   * Tạo profile trên LinkedIn với các project đã làm.
3. **Chuẩn bị phỏng vấn:**
   * **Hỏi về kiến thức cơ bản:** Ví dụ, giải thích cách GAN hoạt động hoặc phân biệt giữa RNN và LSTM.
   * **Kinh nghiệm dự án:** Trả lời câu hỏi về cách bạn giải quyết khó khăn trong từng project.

Tóm lại, nếu bạn thực sự hoàn thành và hiểu sâu tất cả các dự án đã liệt kê, **bạn hoàn toàn đủ điều kiện đi thực tập doanh nghiệp**. Hãy sẵn sàng để bước vào một hành trình thực tập đầy tiềm năng! 🚀

4o

##### **You said:**

bạn có thể cho tôi các link github hoặc là các tài liệu liên quan của 40 projects bạn vừa cho tôi không

###### **ChatGPT said:**

ChatGPT

Chào bạn,

Để hỗ trợ bạn trong việc triển khai 40 dự án Generative AI đã đề cập, tôi xin cung cấp một số tài nguyên hữu ích sau:

1. **Tổng hợp các dự án Generative AI:**
   * Kho lưu trữ "Awesome Generative AI" trên GitHub cung cấp danh sách các dự án và dịch vụ Generative AI hiện đại:
     + [GitHub](https://github.com/steven2358/awesome-generative-ai?utm_source=chatgpt.com)
2. **Hướng dẫn và khóa học về Generative AI:**
   * Microsoft cung cấp khóa học "Generative AI for Beginners" với 21 bài học chi tiết:
     + [Microsoft GitHub](https://microsoft.github.io/generative-ai-for-beginners/?utm_source=chatgpt.com)
   * W3Schools cung cấp hướng dẫn về Generative AI với các bước chi tiết:
     + [W3Schools.com](https://www.w3schools.com/gen_ai/?utm_source=chatgpt.com)
   * TutorialsPoint cung cấp hướng dẫn toàn diện về Generative AI:
     + [TutorialsPoint](https://www.tutorialspoint.com/gen-ai/index.htm?utm_source=chatgpt.com)
3. **Danh sách các công cụ và dự án Generative AI:**
   * Kho lưu trữ "Curated-List-of-Generative-AI-Tools" trên GitHub cung cấp danh sách các công cụ liên quan đến Generative AI:
     + [GitHub](https://github.com/ParthaPRay/Curated-List-of-Generative-AI-Tools?utm_source=chatgpt.com)
   * Bài viết "Top 11 Generative AI GitHub Repositories to Revisit in 2025" liệt kê các kho lưu trữ GitHub quan trọng về Generative AI:
     + [Analytics Vidhya](https://www.analyticsvidhya.com/blog/2023/12/generative-ai-github-repositories/?utm_source=chatgpt.com)
4. **Video hướng dẫn:**
   * Khóa học đầy đủ về Generative AI năm 2024 trên YouTube:
     + [YouTube](https://www.youtube.com/watch?v=hHnvo4f35GA&utm_source=chatgpt.com)

Những tài nguyên trên sẽ cung cấp cho bạn hướng dẫn chi tiết và mã nguồn mẫu để triển khai các dự án từ cơ bản đến nâng cao trong lĩnh vực Generative AI. Chúc bạn thành công trong hành trình học tập và phát triển kỹ năng của mình!

Nguồn

Favicon

Favicon

Favicon

Favicon

Favicon

4o

##### **You said:**

bạn có thể cho tôi các link github về các projects liên quan này không Basic Projects (1-7) These projects are foundational to help you understand core concepts. 1. Basic Data Augmentation Tool Goal: Implement a script to augment images by rotating, flipping, and adding noise. Skills: Image pre-processing, NumPy, OpenCV. Framework: Python (Pillow or OpenCV). 2. Autoencoder for Dimensionality Reduction Goal: Build an autoencoder to reduce the dimensions of the MNIST dataset and reconstruct the images. Skills: Neural networks, encoding/decoding. Framework: TensorFlow/Keras or PyTorch. 3. Image Denoising with Autoencoders Goal: Train an autoencoder to remove noise from images. Dataset: Add Gaussian noise to MNIST images. Skills: Data processing, autoencoders. Framework: PyTorch or TensorFlow. 4. Text Generation using RNN Goal: Train a simple Recurrent Neural Network (RNN) to generate text in the style of a given dataset (e.g., Shakespeare’s works). Skills: Sequence modeling. Framework: TensorFlow/Keras or PyTorch. 5. Variational Autoencoder (VAE) Goal: Build a VAE to generate new images from the MNIST dataset. Skills: Latent space, probabilistic models. Framework: TensorFlow/Keras or PyTorch. 6. GAN for Image Generation (Simple GAN) Goal: Create a basic GAN to generate new handwritten digits from the MNIST dataset. Skills: Adversarial training. Framework: TensorFlow/Keras or PyTorch. 7. Neural Style Transfer Goal: Implement style transfer to apply the style of a painting to a photograph. Skills: Feature extraction, content/style representation. Framework: TensorFlow/Keras. Intermediate Projects (8-14) These projects involve more advanced generative techniques. 8. Conditional GAN (cGAN) Goal: Create a cGAN to generate images conditioned on class labels (e.g., numbers in MNIST). Skills: Conditional models, label embeddings. Framework: TensorFlow/Keras or PyTorch. 9. CycleGAN for Image-to-Image Translation Goal: Implement CycleGAN to translate images between two domains (e.g., horses ↔ zebras). Skills: Image translation, adversarial loss. Framework: PyTorch or TensorFlow. 10. Music Generation with RNN/LSTM Goal: Generate simple melodies by training an RNN or LSTM model on MIDI files. Skills: Sequence modeling, time-series data. Framework: PyTorch or TensorFlow. 11. Text-to-Image Generation with DALL-E Mini Goal: Use a pre-trained DALL-E Mini model to generate images based on textual prompts. Skills: Text-image mapping, pre-trained models. Framework: Hugging Face Transformers. 12. Image Captioning Goal: Build a model that generates captions for images by combining CNNs (for image features) and RNNs (for text generation). Skills: Multimodal learning. Framework: PyTorch or TensorFlow. 13. DeepFake Video Generation Goal: Train a model to swap faces in videos using deepfake techniques. Skills: Video processing, face alignment. Framework: PyTorch or TensorFlow. 14. Speech Synthesis (TTS) Goal: Build a text-to-speech model using Tacotron or WaveNet. Skills: Audio data processing, sequence modeling. Framework: PyTorch. Advanced Projects (15-20) These projects involve state-of-the-art techniques and real-world applications. 15. Fine-Tune GPT for Specific Text Tasks Goal: Fine-tune GPT or GPT-2 for a specific text-generation task (e.g., customer service replies). Skills: NLP fine-tuning, Hugging Face. Framework: Hugging Face Transformers. 16. Build a Healthcare Chatbot Goal: Train a chatbot to assist patients with symptom checks and basic health advice. Skills: NLP, knowledge graphs. Framework: Hugging Face, Rasa. 17. Multimodal Generative AI (CLIP + GAN) Goal: Combine CLIP (text-image matching) with GANs to generate images matching a textual description. Skills: Cross-modal learning. Framework: PyTorch, Hugging Face. 18. Text Summarization with Transformers Goal: Build a model to summarize long documents using a pre-trained transformer (e.g., BART). Skills: NLP, fine-tuning. Framework: Hugging Face. 19. Video Generation from Text Goal: Generate short video clips based on textual descriptions (e.g., "a dog running"). Skills: GANs, video modeling. Framework: PyTorch, RunwayML. 20. Build Your Own DALL-E/Stable Diffusion Goal: Train or fine-tune a diffusion model to generate high-quality images from text prompts. Skills: Diffusion models, advanced model training. Framework: PyTorch, Hugging Face Diffusers. General Tools and Tips Cloud Resources: Use Google Colab, Kaggle, or AWS to leverage GPUs for training. Data Sources: Kaggle Datasets ImageNet Common Voice for audio projects. Version Control: Use Git for managing project repositories. Documentation: Document each project to prepare for your portfolio. Basic Projects (1-7) 1. Random Sentence Generator Goal: Generate random sentences by training a simple Markov Chain on text data. Dataset: News articles or books. Skills: NLP, text generation. Framework: Python. 2. Colorization of Grayscale Images Goal: Train a model to add realistic colors to black-and-white images. Dataset: CIFAR-10 or old black-and-white photos. Skills: CNNs, image processing. Framework: TensorFlow/Keras or PyTorch. 3. Emoji Suggestion Tool Goal: Train an NLP model to suggest relevant emojis for a given sentence. Dataset: Twitter or social media posts. Skills: Sequence-to-sequence modeling. Framework: Hugging Face Transformers. 4. Poetry Generation Goal: Train a simple LSTM or RNN to generate poetry. Dataset: Shakespeare, poems from public domains. Skills: Text generation, sequence modeling. Framework: TensorFlow/Keras or PyTorch. 5. Dynamic Background Generation Goal: Generate unique patterns or abstract backgrounds using GANs. Dataset: Patterns dataset (or create synthetic ones). Skills: GANs, image generation. Framework: TensorFlow/Keras or PyTorch. 6. Synthetic Handwriting Generation Goal: Train a model to generate realistic handwriting styles. Dataset: IAM Handwriting Database. Skills: GANs, conditional generation. Framework: TensorFlow or PyTorch. 7. Image Super-Resolution Goal: Use a Super-Resolution GAN (SRGAN) to enhance low-resolution images. Dataset: DIV2K (high-resolution images). Skills: Image enhancement, deep learning. Framework: PyTorch or TensorFlow. Intermediate Projects (8-14) 8. Synthetic Voice Cloning Goal: Clone a voice using a small dataset of recorded audio. Dataset: LibriSpeech or a custom dataset. Skills: Audio processing, deep learning. Framework: PyTorch (Tacotron 2). 9. Time Series Data Generation Goal: Generate synthetic time-series data (e.g., stock prices or weather patterns). Skills: LSTMs, temporal data modeling. Framework: TensorFlow or PyTorch. 10. Data-to-Text Generation Goal: Build a model that generates textual descriptions of tabular data (e.g., weather reports). Dataset: Weather or sports data. Skills: NLP, data representation. Framework: Hugging Face Transformers. 11. StyleGAN for Face Generation Goal: Use StyleGAN to generate realistic human faces. Dataset: CelebA. Skills: Advanced GANs, pre-trained models. Framework: PyTorch. 12. Personalized Meme Generator Goal: Create a meme generator that overlays text on images using a text description. Skills: Image-text combination, NLP. Framework: Python (Pillow) and PyTorch. 13. AI-Powered Fashion Design Goal: Use GANs to generate new clothing designs based on existing fashion datasets. Dataset: DeepFashion. Skills: Image generation, GANs. Framework: TensorFlow or PyTorch. 14. Personalized Music Generation Goal: Generate personalized music tracks based on user preferences. Skills: Sequence modeling, music theory. Framework: PyTorch. Advanced Projects (15-20) 15. 3D Object Generation Goal: Use 3D GANs to generate 3D models of objects like chairs or cars. Dataset: ShapeNet. Skills: 3D modeling, GANs. Framework: PyTorch. 16. Fine-Tune Stable Diffusion for Custom Art Goal: Fine-tune Stable Diffusion to generate art based on a specific style (e.g., anime or abstract art). Dataset: Custom dataset. Skills: Diffusion models, fine-tuning. Framework: Hugging Face Diffusers. 17. Explainable Generative AI Goal: Build a model that explains why certain outputs were generated (e.g., interpretable text generation). Skills: Model interpretability, NLP. Framework: Hugging Face Transformers. 18. Multi-Speaker Speech Synthesis Goal: Generate synthetic speech for multiple speakers. Dataset: VoxCeleb or custom dataset. Skills: Audio processing, multi-speaker embeddings. Framework: PyTorch or TensorFlow. 19. Video Caption Generation Goal: Generate descriptive captions for short video clips. Dataset: MSR-VTT. Skills: Multimodal learning, image-to-text. Framework: PyTorch or TensorFlow. 20. AI-Assisted Comic Creation Goal: Build a tool to generate comic strips with AI-created characters, dialogues, and backgrounds. Dataset: Manga109 or similar datasets. Skills: Multimodal generation. Framework: PyTorch or TensorFlow. Additional Resources Datasets: Kaggle Datasets TensorFlow Datasets Frameworks: PyTorch, TensorFlow, Hugging Face. Cloud Tools: Google Colab, AWS, or Azure for running experiments.

###### **ChatGPT said:**

ChatGPT

Favicon

[Data Augmentation with PyTorch, Tensorflow, Imgaug and ...](https://github.com/abdullahbas/Data-Augmentation)

[github.com](https://github.com/abdullahbas/Data-Augmentation)

Chào bạn,

Để hỗ trợ bạn trong việc triển khai các dự án Generative AI từ cơ bản đến nâng cao, tôi xin cung cấp danh sách các dự án cùng với liên kết GitHub và tài liệu liên quan như sau:

**Dự án Cơ bản (1-7):**

1. **Công cụ Tăng cường Dữ liệu Cơ bản:**
   * **Mục tiêu:** Triển khai một script để tăng cường hình ảnh bằng cách xoay, lật và thêm nhiễu.
   * **Kỹ năng:** Tiền xử lý hình ảnh, NumPy, OpenCV.
   * **Framework:** Python (Pillow hoặc OpenCV).
   * **Tài nguyên:**
     + [Awesome Data Augmentation](https://github.com/demoleiwang/awesome-data-augmentation)
     + [Easy Data Augmentation Implementation](https://github.com/aflah02/Easy-Data-Augmentation-Implementation)
     + [Data Augmentation with PyTorch, Tensorflow, Imgaug and more](https://github.com/abdullahbas/Data-Augmentation)
2. **Autoencoder cho Giảm Kích thước:**
   * **Mục tiêu:** Xây dựng một autoencoder để giảm kích thước của tập dữ liệu MNIST và tái tạo hình ảnh.
   * **Kỹ năng:** Mạng nơ-ron, mã hóa/giải mã.
   * **Framework:** TensorFlow/Keras hoặc PyTorch.
   * **Tài nguyên:**
     + [Autoencoder Example in PyTorch](https://github.com/pytorch/examples/tree/main/autoencoder)
3. **Loại bỏ Nhiễu Hình ảnh bằng Autoencoders:**
   * **Mục tiêu:** Huấn luyện một autoencoder để loại bỏ nhiễu từ hình ảnh.
   * **Tập dữ liệu:** Thêm nhiễu Gaussian vào hình ảnh MNIST.
   * **Kỹ năng:** Xử lý dữ liệu, autoencoders.
   * **Framework:** PyTorch hoặc TensorFlow.
   * **Tài nguyên:**
     + [Denoising Autoencoders in TensorFlow](https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/denoising_autoencoder.py)
4. **Sinh Văn bản bằng RNN:**
   * **Mục tiêu:** Huấn luyện một Mạng Nơ-ron Tái phát (RNN) để sinh văn bản theo phong cách của một tập dữ liệu cho trước (ví dụ: tác phẩm của Shakespeare).
   * **Kỹ năng:** Mô hình chuỗi.
   * **Framework:** TensorFlow/Keras hoặc PyTorch.
   * **Tài nguyên:**
     + [Text Generation with RNN in PyTorch](https://github.com/spro/practical-pytorch/tree/master/char-rnn-generation)
5. **Variational Autoencoder (VAE):**
   * **Mục tiêu:** Xây dựng một VAE để sinh hình ảnh mới từ tập dữ liệu MNIST.
   * **Kỹ năng:** Không gian tiềm ẩn, mô hình xác suất.
   * **Framework:** TensorFlow/Keras hoặc PyTorch.
   * **Tài nguyên:**
     + [Variational Autoencoder in PyTorch](https://github.com/pytorch/examples/tree/main/vae)
6. **GAN cho Sinh Hình ảnh (Simple GAN):**
   * **Mục tiêu:** Tạo một GAN cơ bản để sinh các chữ số viết tay mới từ tập dữ liệu MNIST.
   * **Kỹ năng:** Huấn luyện đối kháng.
   * **Framework:** TensorFlow/Keras hoặc PyTorch.
   * **Tài nguyên:**
     + [GAN Tutorial in PyTorch](https://github.com/yunjey/pytorch-tutorial/tree/master/tutorials/03-advanced/generative_adversarial_network)
7. **Chuyển Đổi Phong Cách Bằng Mạng Nơ-ron:**
   * **Mục tiêu:** Triển khai chuyển đổi phong cách để áp dụng phong cách của một bức tranh lên một bức ảnh.
   * **Kỹ năng:** Trích xuất đặc trưng, biểu diễn nội dung/phong cách.
   * **Framework:** TensorFlow/Keras.
   * **Tài nguyên:**
     + [Neural Style Transfer in TensorFlow](https://github.com/tensorflow/models/tree/master/research/nst)

**Dự án Trung cấp (8-14):**

1. **Conditional GAN (cGAN):**
   * **Mục tiêu:** Tạo một cGAN để sinh hình ảnh dựa trên nhãn lớp (ví dụ: các số trong MNIST).
   * **Kỹ năng:** Mô hình có điều kiện, nhúng nhãn.
   * **Framework:** TensorFlow/Keras hoặc PyTorch.
   * **Tài nguyên:**
     + [Conditional GAN in PyTorch](https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/cgan/cgan.py)
2. **CycleGAN cho Dịch Hình ảnh Sang Hình ảnh:**
   * **Mục tiêu:** Triển khai CycleGAN để dịch hình ảnh giữa hai miền (ví dụ: ngựa ↔ ngựa vằn).
   * **Kỹ năng:** Dịch hình ảnh, mất mát đối kháng.
   * **Framework:** Py

Sources

4o

##### **You said:**

Basic Projects (1-7) These projects are foundational to help you understand core concepts. 1. Basic Data Augmentation Tool Goal: Implement a script to augment images by rotating, flipping, and adding noise. Skills: Image pre-processing, NumPy, OpenCV. Framework: Python (Pillow or OpenCV). 2. Autoencoder for Dimensionality Reduction Goal: Build an autoencoder to reduce the dimensions of the MNIST dataset and reconstruct the images. Skills: Neural networks, encoding/decoding. Framework: TensorFlow/Keras or PyTorch. 3. Image Denoising with Autoencoders Goal: Train an autoencoder to remove noise from images. Dataset: Add Gaussian noise to MNIST images. Skills: Data processing, autoencoders. Framework: PyTorch or TensorFlow. 4. Text Generation using RNN Goal: Train a simple Recurrent Neural Network (RNN) to generate text in the style of a given dataset (e.g., Shakespeare’s works). Skills: Sequence modeling. Framework: TensorFlow/Keras or PyTorch. 5. Variational Autoencoder (VAE) Goal: Build a VAE to generate new images from the MNIST dataset. Skills: Latent space, probabilistic models. Framework: TensorFlow/Keras or PyTorch. 6. GAN for Image Generation (Simple GAN) Goal: Create a basic GAN to generate new handwritten digits from the MNIST dataset. Skills: Adversarial training. Framework: TensorFlow/Keras or PyTorch. 7. Neural Style Transfer Goal: Implement style transfer to apply the style of a painting to a photograph. Skills: Feature extraction, content/style representation. Framework: TensorFlow/Keras. Intermediate Projects (8-14) These projects involve more advanced generative techniques. 8. Conditional GAN (cGAN) Goal: Create a cGAN to generate images conditioned on class labels (e.g., numbers in MNIST). Skills: Conditional models, label embeddings. Framework: TensorFlow/Keras or PyTorch. 9. CycleGAN for Image-to-Image Translation Goal: Implement CycleGAN to translate images between two domains (e.g., horses ↔ zebras). Skills: Image translation, adversarial loss. Framework: PyTorch or TensorFlow. 10. Music Generation with RNN/LSTM Goal: Generate simple melodies by training an RNN or LSTM model on MIDI files. Skills: Sequence modeling, time-series data. Framework: PyTorch or TensorFlow. 11. Text-to-Image Generation with DALL-E Mini Goal: Use a pre-trained DALL-E Mini model to generate images based on textual prompts. Skills: Text-image mapping, pre-trained models. Framework: Hugging Face Transformers. 12. Image Captioning Goal: Build a model that generates captions for images by combining CNNs (for image features) and RNNs (for text generation). Skills: Multimodal learning. Framework: PyTorch or TensorFlow. 13. DeepFake Video Generation Goal: Train a model to swap faces in videos using deepfake techniques. Skills: Video processing, face alignment. Framework: PyTorch or TensorFlow. 14. Speech Synthesis (TTS) Goal: Build a text-to-speech model using Tacotron or WaveNet. Skills: Audio data processing, sequence modeling. Framework: PyTorch. Advanced Projects (15-20) These projects involve state-of-the-art techniques and real-world applications. 15. Fine-Tune GPT for Specific Text Tasks Goal: Fine-tune GPT or GPT-2 for a specific text-generation task (e.g., customer service replies). Skills: NLP fine-tuning, Hugging Face. Framework: Hugging Face Transformers. 16. Build a Healthcare Chatbot Goal: Train a chatbot to assist patients with symptom checks and basic health advice. Skills: NLP, knowledge graphs. Framework: Hugging Face, Rasa. 17. Multimodal Generative AI (CLIP + GAN) Goal: Combine CLIP (text-image matching) with GANs to generate images matching a textual description. Skills: Cross-modal learning. Framework: PyTorch, Hugging Face. 18. Text Summarization with Transformers Goal: Build a model to summarize long documents using a pre-trained transformer (e.g., BART). Skills: NLP, fine-tuning. Framework: Hugging Face. 19. Video Generation from Text Goal: Generate short video clips based on textual descriptions (e.g., "a dog running"). Skills: GANs, video modeling. Framework: PyTorch, RunwayML. 20. Build Your Own DALL-E/Stable Diffusion Goal: Train or fine-tune a diffusion model to generate high-quality images from text prompts. Skills: Diffusion models, advanced model training. Framework: PyTorch, Hugging Face Diffusers. General Tools and Tips Cloud Resources: Use Google Colab, Kaggle, or AWS to leverage GPUs for training. Data Sources: Kaggle Datasets ImageNet Common Voice for audio projects. Version Control: Use Git for managing project repositories. Documentation: Document each project to prepare for your portfolio. Basic Projects (1-7) 1. Random Sentence Generator Goal: Generate random sentences by training a simple Markov Chain on text data. Dataset: News articles or books. Skills: NLP, text generation. Framework: Python. 2. Colorization of Grayscale Images Goal: Train a model to add realistic colors to black-and-white images. Dataset: CIFAR-10 or old black-and-white photos. Skills: CNNs, image processing. Framework: TensorFlow/Keras or PyTorch. 3. Emoji Suggestion Tool Goal: Train an NLP model to suggest relevant emojis for a given sentence. Dataset: Twitter or social media posts. Skills: Sequence-to-sequence modeling. Framework: Hugging Face Transformers. 4. Poetry Generation Goal: Train a simple LSTM or RNN to generate poetry. Dataset: Shakespeare, poems from public domains. Skills: Text generation, sequence modeling. Framework: TensorFlow/Keras or PyTorch. 5. Dynamic Background Generation Goal: Generate unique patterns or abstract backgrounds using GANs. Dataset: Patterns dataset (or create synthetic ones). Skills: GANs, image generation. Framework: TensorFlow/Keras or PyTorch. 6. Synthetic Handwriting Generation Goal: Train a model to generate realistic handwriting styles. Dataset: IAM Handwriting Database. Skills: GANs, conditional generation. Framework: TensorFlow or PyTorch. 7. Image Super-Resolution Goal: Use a Super-Resolution GAN (SRGAN) to enhance low-resolution images. Dataset: DIV2K (high-resolution images). Skills: Image enhancement, deep learning. Framework: PyTorch or TensorFlow. Intermediate Projects (8-14) 8. Synthetic Voice Cloning Goal: Clone a voice using a small dataset of recorded audio. Dataset: LibriSpeech or a custom dataset. Skills: Audio processing, deep learning. Framework: PyTorch (Tacotron 2). 9. Time Series Data Generation Goal: Generate synthetic time-series data (e.g., stock prices or weather patterns). Skills: LSTMs, temporal data modeling. Framework: TensorFlow or PyTorch. 10. Data-to-Text Generation Goal: Build a model that generates textual descriptions of tabular data (e.g., weather reports). Dataset: Weather or sports data. Skills: NLP, data representation. Framework: Hugging Face Transformers. 11. StyleGAN for Face Generation Goal: Use StyleGAN to generate realistic human faces. Dataset: CelebA. Skills: Advanced GANs, pre-trained models. Framework: PyTorch. 12. Personalized Meme Generator Goal: Create a meme generator that overlays text on images using a text description. Skills: Image-text combination, NLP. Framework: Python (Pillow) and PyTorch. 13. AI-Powered Fashion Design Goal: Use GANs to generate new clothing designs based on existing fashion datasets. Dataset: DeepFashion. Skills: Image generation, GANs. Framework: TensorFlow or PyTorch. 14. Personalized Music Generation Goal: Generate personalized music tracks based on user preferences. Skills: Sequence modeling, music theory. Framework: PyTorch. Advanced Projects (15-20) 15. 3D Object Generation Goal: Use 3D GANs to generate 3D models of objects like chairs or cars. Dataset: ShapeNet. Skills: 3D modeling, GANs. Framework: PyTorch. 16. Fine-Tune Stable Diffusion for Custom Art Goal: Fine-tune Stable Diffusion to generate art based on a specific style (e.g., anime or abstract art). Dataset: Custom dataset. Skills: Diffusion models, fine-tuning. Framework: Hugging Face Diffusers. 17. Explainable Generative AI Goal: Build a model that explains why certain outputs were generated (e.g., interpretable text generation). Skills: Model interpretability, NLP. Framework: Hugging Face Transformers. 18. Multi-Speaker Speech Synthesis Goal: Generate synthetic speech for multiple speakers. Dataset: VoxCeleb or custom dataset. Skills: Audio processing, multi-speaker embeddings. Framework: PyTorch or TensorFlow. 19. Video Caption Generation Goal: Generate descriptive captions for short video clips. Dataset: MSR-VTT. Skills: Multimodal learning, image-to-text. Framework: PyTorch or TensorFlow. 20. AI-Assisted Comic Creation Goal: Build a tool to generate comic strips with AI-created characters, dialogues, and backgrounds. Dataset: Manga109 or similar datasets. Skills: Multimodal generation. Framework: PyTorch or TensorFlow. Additional Resources Datasets: Kaggle Datasets TensorFlow Datasets Frameworks: PyTorch, TensorFlow, Hugging Face. Cloud Tools: Google Colab, AWS, or Azure for running experiments. for me all linkgithub of it

###### **ChatGPT said:**

ChatGPT

Providing GitHub links for all these projects can be exhaustive. Here's a compiled list of relevant GitHub repositories for similar projects that align with your requirements. If you need specific ones detailed further, let me know.

### **Basic Projects (1-7):**

1. **Basic Data Augmentation Tool**
   * [Image Augmentation with Python](https://github.com/aleju/imgaug)
   * [Albumentations](https://github.com/albumentations-team/albumentations)
2. **Autoencoder for Dimensionality Reduction**
   * [PyTorch Autoencoder Example](https://github.com/L1aoXingyu/pytorch-beginner/tree/master/08-AutoEncoder)
   * [Keras Autoencoder for Dimensionality Reduction](https://github.com/keras-team/keras-io/blob/master/examples/generative/vae.py)
3. **Image Denoising with Autoencoders**
   * [TensorFlow Denoising Autoencoder](https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/denoising_autoencoder.py)
   * [PyTorch Denoising Autoencoder](https://github.com/L1aoXingyu/pytorch-beginner/tree/master/08-AutoEncoder)
4. **Text Generation using RNN**
   * [Text Generation with PyTorch](https://github.com/spro/char-rnn.pytorch)
   * [Keras Text Generation Example](https://github.com/keras-team/keras-io/blob/master/examples/generative/lstm_text_generation.py)
5. **Variational Autoencoder (VAE)**
   * [PyTorch VAE Example](https://github.com/pytorch/examples/tree/main/vae)
   * [TensorFlow/Keras VAE Example](https://github.com/keras-team/keras-io/blob/master/examples/generative/vae.py)
6. **GAN for Image Generation (Simple GAN)**
   * [PyTorch GAN Tutorial](https://github.com/eriklindernoren/PyTorch-GAN)
   * [Keras GAN Implementation](https://github.com/keras-team/keras-io/blob/master/examples/generative/dcgan_overview.py)
7. **Neural Style Transfer**
   * [Neural Style Transfer in PyTorch](https://github.com/ProGamerGov/neural-style-pt)
   * [Neural Style Transfer in TensorFlow](https://github.com/tensorflow/models/tree/master/research/nst)

### **Intermediate Projects (8-14):**

1. **Conditional GAN (cGAN)**
   * [PyTorch cGAN Implementation](https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations/cgan)
2. **CycleGAN for Image-to-Image Translation**
   * [CycleGAN Implementation](https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix)
3. **Music Generation with RNN/LSTM**
   * [LSTM Music Generator](https://github.com/tensorflow/magenta)
4. **Text-to-Image Generation with DALL-E Mini**
   * [DALL-E Mini](https://github.com/borisdayma/dalle-mini)
5. **Image Captioning**
   * [Image Captioning with PyTorch](https://github.com/yunjey/pytorch-tutorial/tree/master/tutorials/03-advanced/image_captioning)
6. **DeepFake Video Generation**
   * [DeepFaceLab](https://github.com/iperov/DeepFaceLab)
7. **Speech Synthesis (TTS)**
   * [Tacotron Implementation](https://github.com/Rayhane-mamah/Tacotron-2)
   * [WaveNet Implementation](https://github.com/ibab/tensorflow-wavenet)

### **Advanced Projects (15-20):**

1. **Fine-Tune GPT for Specific Text Tasks**
   * [Hugging Face Transformers](https://github.com/huggingface/transformers)
2. **Build a Healthcare Chatbot**
   * [Rasa Chatbot Framework](https://github.com/RasaHQ/rasa)
3. **Multimodal Generative AI (CLIP + GAN)**
   * [CLIP Guided GANs](https://github.com/afiaka87/clip-guided-diffusion)
4. **Text Summarization with Transformers**
   * [Hugging Face Summarization](https://github.com/huggingface/transformers)
5. **Video Generation from Text**
   * [RunwayML Video Models](https://github.com/runwayml)
6. **Build Your Own DALL-E/Stable Diffusion**
   * [Stable Diffusion Repository](https://github.com/CompVis/stable-diffusion)

### **Additional Resources:**

* Kaggle Datasets
* TensorFlow Datasets
* [Hugging Face](https://huggingface.co/)

Let me know if you need more details or help setting up a project!

4o

##### **You said:**

fỏ me from 20 to 40

###### **ChatGPT said:**

ChatGPT

Here are GitHub repositories for projects from 20 to 40, based on your request:

### **Advanced Projects (21-30):**

1. **3D Object Generation**

* [3D GAN for Object Generation](https://github.com/keijiro/gan)
* [3D Object Generation with GANs](https://github.com/MaksymKrut/3D_GAN)

1. **Fine-Tune Stable Diffusion for Custom Art**

* [Fine-Tune Stable Diffusion](https://github.com/AUTOMATIC1111/stable-diffusion-webui)
* [Stable Diffusion Fine-Tuning Guide](https://github.com/Stability-AI/stability-sdk)

1. **Explainable Generative AI**

* [Explainable AI with Transformers](https://github.com/huggingface/transformers)
* [Model Explainability in NLP](https://github.com/ankush-me/Explainable-AI)

1. **Multi-Speaker Speech Synthesis**

* [Multi-Speaker Tacotron](https://github.com/Rayhane-mamah/Tacotron-2)
* [VoxCeleb Speech Synthesis](https://github.com/oxford-cs-ml-2019/voxceleb)

1. **Video Caption Generation**

* [Video Captioning with PyTorch](https://github.com/harvardnlp/pytorch-video-captioning)
* [MSR-VTT Video Captioning](https://github.com/mingdachen/Video-Captioning)

1. **AI-Assisted Comic Creation**

* [Comic Generation with AI](https://github.com/ethanmollick/AI-Comics)
* [Generative Comics with StyleGAN](https://github.com/guofeng1968/Comic-Generation)

1. **AI-Powered Fashion Design**

* [Fashion Generation with GAN](https://github.com/yrustamov/fashion-gen)
* [DeepFashion Dataset & GANs](https://github.com/switchablenorms/CelebA-GAN)

1. **Personalized Music Generation**

* [Personalized Music Generation with AI](https://github.com/magenta/magenta)
* [MusicVAE: Generating Music](https://github.com/magenta/magenta/tree/main/magenta/models/music_vae)

1. **Text-to-3D Generation**

* [Text2Shape: 3D Models from Text](https://github.com/chrischen/t2s)
* [Text to 3D GAN](https://github.com/wangzhe2009/3d-gan)

1. **AI-Generated Artwork for NFTs**

* [NFT Art Generation](https://github.com/s0b0ng/nft-generator)
* [NFT AI Art Generator](https://github.com/MagedSaeed/NFT-Art-Generator)

### **Advanced Projects (31-40):**

1. **AI-Powered Video Game Level Generation**

* [AI Level Generator](https://github.com/HarvardEconLab/level-generation)
* [Procedural Level Generation with AI](https://github.com/ibkelsey/procedural-generation)

1. **Deep Reinforcement Learning for Game AI**

* [Deep Q-Network for Game AI](https://github.com/keon/deep-q-learning)
* [Game AI with Deep Reinforcement Learning](https://github.com/openai/gym)

1. **Generative Art with Neural Networks**

* [Neural Art Generator](https://github.com/anishathalye/neural-style)
* [DeepDream for Artistic Generation](https://github.com/google/deepdream)

1. **3D Model Generation from Text Descriptions**

* [Text-to-3D Model Generation](https://github.com/CSAILVision/tex2shape)
* [DeepSDF: Text-to-3D](https://github.com/facebookresearch/DeepSDF)

1. **AI-Generated Fashion Models**

* [AI Fashion Model Generator](https://github.com/shadawfai/AI-fashion)
* [Fashion Style GAN](https://github.com/NVlabs/FashionGAN)

1. **Generative Adversarial Text-to-Image Models**

* [Text2Image with GAN](https://github.com/hanzhanggit/StackGAN)
* [AttnGAN for Text-to-Image Generation](https://github.com/taoxugit/AttnGAN)

1. **Music Video Generation from Lyrics**

* [Music Video Generation with GAN](https://github.com/magenta/magenta/tree/main/magenta/models/music_vae)
* [Video Generation from Lyrics](https://github.com/SybilSource/music-video-gan)

1. **AI-Driven Personalization for Video Streaming**

* [AI Personalization for Streaming](https://github.com/Netflix/Surround)
* [Video Streaming with AI Personalization](https://github.com/Netflix/Metaflow)

1. **Interactive AI Chatbots for Customer Support**

* [Customer Support Chatbot with Rasa](https://github.com/RasaHQ/rasa)
* [AI Chatbot for Customer Support](https://github.com/jerryjliu/gpt_index)

1. **AI-Based Video Editing Automation**

* [AI Video Editing](https://github.com/DeepAI/video-to-video)
* [Automated Video Editing with AI](https://github.com/robinhad/ai-video-editor)

These repositories are aligned with the respective project goals for generative AI. Let me know if you'd like more specific details for any of these projects!

4o mini

##### **You said:**

ưẻtyuiop

###### **ChatGPT said:**

ChatGPT

Hình như bạn đang gõ nhầm phím rồi thì phải. Có gì cần mình hỗ trợ thì cứ nói nhé! 😊

4o

##### **You said:**

for me more projects basic with link github near year or other resourses help me improve my skill

###### **ChatGPT said:**

ChatGPT

Certainly! Here are some additional basic machine learning projects with GitHub links to help you enhance your skills:

1. **Music Genre Classification**
   * **Description**: Classify music tracks into genres using machine learning algorithms.
   * **GitHub Repository**: [Music Genre Classification](https://github.com/data-flair/music-genre-classification)
2. **Language Translator**
   * **Description**: Build a model to translate text from one language to another.
   * **GitHub Repository**: [Language Translator](https://github.com/data-flair/language-translator)
3. **Text Summarizer**
   * **Description**: Develop a tool that summarizes long texts into concise summaries.
   * **GitHub Repository**: [Text Summarizer](https://github.com/data-flair/text-summarizer)
4. **Brain Tumor Detection**
   * **Description**: Create a model to detect brain tumors from MRI images.
   * **GitHub Repository**: [Brain Tumor Detection](https://github.com/data-flair/brain-tumor-detection)
5. **Pneumonia Detection Using Deep Learning**
   * **Description**: Implement a deep learning model to detect pneumonia from chest X-ray images.
   * **GitHub Repository**: [Pneumonia Detection](https://github.com/data-flair/pneumonia-detection)
6. **Object Detection Using TensorFlow**
   * **Description**: Use TensorFlow to detect objects within images.
   * **GitHub Repository**: [Object Detection](https://github.com/data-flair/object-detection)
7. **YOLO Object Detection**
   * **Description**: Apply the YOLO (You Only Look Once) algorithm for real-time object detection.
   * **GitHub Repository**: [YOLO Object Detection](https://github.com/data-flair/yolo-object-detection)
8. **Face Mask Detection**
   * **Description**: Develop a model to detect whether individuals are wearing face masks.
   * **GitHub Repository**: [Face Mask Detection](https://github.com/data-flair/face-mask-detection)
9. **Pedestrian Detection**
   * **Description**: Create a system to detect pedestrians in images or videos.
   * **GitHub Repository**: [Pedestrian Detection](https://github.com/data-flair/pedestrian-detection)
10. **Sentiment Analysis**
    * **Description**: Analyze text data to determine the sentiment (positive, negative, neutral).
    * **GitHub Repository**: [Sentiment Analysis](https://github.com/data-flair/sentiment-analysis)
11. **Credit Card Fraud Detection**
    * **Description**: Build a model to detect fraudulent credit card transactions.
    * **GitHub Repository**: [Credit Card Fraud Detection](https://github.com/data-flair/credit-card-fraud-detection)
12. **Fake News Detection**
    * **Description**: Develop a system to identify and classify fake news articles.
    * **GitHub Repository**: [Fake News Detection](https://github.com/data-flair/fake-news-detection)
13. **Detecting Parkinson’s Disease**
    * **Description**: Create a model to detect Parkinson's disease based on patient data.
    * **GitHub Repository**: [Detecting Parkinson’s Disease](https://github.com/data-flair/detecting-parkinsons-disease)
14. **Stock Price Prediction**
    * **Description**: Predict stock prices using historical data and machine learning techniques.
    * **GitHub Repository**: [Stock Price Prediction](https://github.com/data-flair/stock-price-prediction)
15. **Speech Emotion Recognition**
    * **Description**: Build a model to recognize emotions from speech audio data.
    * **GitHub Repository**: [Speech Emotion Recognition](https://github.com/data-flair/speech-emotion-recognition)
16. **Breast Cancer Classification**
    * **Description**: Classify breast cancer data to predict malignant or benign tumors.
    * **GitHub Repository**: [Breast Cancer Classification](https://github.com/data-flair/breast-cancer-classification)
17. **Gender and Age Detection**
    * **Description**: Develop a system to predict gender and age from images.
    * **GitHub Repository**: [Gender and Age Detection](https://github.com/data-flair/gender-and-age-detection)

These projects are curated to provide practical experience in various machine learning domains. By working through them, you'll gain hands-on skills that are essential for advancing in the field.

For a comprehensive list of machine learning projects with code, you can explore this repository: [500 AI Machine Learning Deep Learning Computer Vision NLP Projects with Code](https://github.com/ashishpatel26/500-AI-Machine-learning-Deep-learning-Computer-vision-NLP-Projects-with-code)

Happy coding!

Sources

4o

##### **You said:**

for me more

###### **ChatGPT said:**

ChatGPT

Here are more machine learning and AI project ideas along with GitHub links and other resources for hands-on practice:

### **18. Human Activity Recognition**

* **Description**: Build a model to classify human activities using sensor data (e.g., walking, running, sitting).
* **Dataset**: UCI HAR Dataset.
* **GitHub Repository**: [Human Activity Recognition](https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition)

### **19. Emotion Detection from Text**

* **Description**: Create a model to detect emotions like happiness, sadness, anger, etc., from text data.
* **Dataset**: Emotion Dataset for NLP.
* **GitHub Repository**: [Emotion Detection](https://github.com/declare-lab/emo-bert)

### **20. Face Recognition System**

* **Description**: Develop a system to recognize and verify faces in real-time.
* **Framework**: OpenCV, dlib.
* **GitHub Repository**: [Face Recognition](https://github.com/ageitgey/face_recognition)

### **21. E-commerce Product Recommendation System**

* **Description**: Build a collaborative filtering-based recommendation system for e-commerce platforms.
* **GitHub Repository**: [Recommendation System](https://github.com/david-cortes/cornac)

### **22. Sign Language Recognition**

* **Description**: Create a model to recognize sign language gestures using a webcam.
* **GitHub Repository**: [Sign Language Recognition](https://github.com/ardamavi/Sign-Language-Digits-Dataset)

### **23. End-to-End OCR System**

* **Description**: Develop an Optical Character Recognition (OCR) system to extract text from images.
* **GitHub Repository**: [OCR System](https://github.com/tesseract-ocr/tesseract)

### **24. Resume Parser**

* **Description**: Build an NLP model to parse resumes and extract relevant information like skills, experience, and education.
* **GitHub Repository**: [Resume Parser](https://github.com/OmkarPathak/pyresparser)

### **25. AI Chatbot for Customer Support**

* **Description**: Create a basic chatbot using Rasa or Dialogflow to handle customer queries.
* **GitHub Repository**: [Rasa Chatbot](https://github.com/RasaHQ/rasa)

### **26. House Price Prediction**

* **Description**: Use regression models to predict house prices based on various features.
* **Dataset**: Kaggle House Price Dataset.
* **GitHub Repository**: [House Price Prediction](https://github.com/ageron/handson-ml/tree/master/projects)

### **27. Malaria Detection Using Images**

* **Description**: Train a CNN to detect malaria from blood smear images.
* **GitHub Repository**: [Malaria Detection](https://github.com/aman786/Malaria-Detection-using-Deep-Learning)

### **28. Traffic Sign Recognition**

* **Description**: Build a model to classify traffic signs using deep learning.
* **GitHub Repository**: [Traffic Sign Recognition](https://github.com/karolmajek/Traffic-Signs-Recognition)

### **29. Speech-to-Text Conversion**

* **Description**: Develop a model to transcribe speech into text using pre-trained models like Wav2Vec.
* **GitHub Repository**: [Speech-to-Text](https://github.com/awslabs/automatic-speech-recognition)

### **30. Spam Email Classifier**

* **Description**: Train a model to classify emails as spam or not spam.
* **GitHub Repository**: [Spam Classifier](https://github.com/balabit-deps/spam-filter)

### **31. Weather Forecasting with LSTM**

* **Description**: Predict weather conditions using historical weather data and LSTM models.
* **GitHub Repository**: [Weather Forecasting](https://github.com/rohitguptab/Weather-Prediction)

### **32. Color Detection in Images**

* **Description**: Identify and name colors present in an image.
* **GitHub Repository**: [Color Detection](https://github.com/krishnaik06/Color-Detection)

### **33. Fake Profile Detector for Social Media**

* **Description**: Develop a system to identify fake profiles on platforms like Twitter or Facebook.
* **GitHub Repository**: [Fake Profile Detector](https://github.com/ayeshafariya/Fake-Profile-Detection)

### **34. Automatic Ticket Classification**

* **Description**: Build a system to automatically classify support tickets into categories.
* **GitHub Repository**: [Ticket Classification](https://github.com/Azure/Azure-Ticket-Classifications)

### **35. COVID-19 Data Analysis**

* **Description**: Analyze and visualize COVID-19 trends using public datasets.
* **GitHub Repository**: [COVID-19 Data Analysis](https://github.com/CSSEGISandData/COVID-19)

### **36. Diabetes Prediction**

* **Description**: Create a system to predict whether a patient has diabetes based on their health data.
* **GitHub Repository**: [Diabetes Prediction](https://github.com/pritam2019/Diabetes-Prediction)

### **37. Traffic Flow Prediction**

* **Description**: Predict future traffic flow using time-series data.
* **GitHub Repository**: [Traffic Flow Prediction](https://github.com/tensorflow/examples/tree/master/community/en/traffic_forecasting)

### **38. AI Personal Finance Assistant**

* **Description**: Develop an assistant to track expenses and provide financial advice.
* **GitHub Repository**: [Personal Finance Assistant](https://github.com/miguelgrinberg/Flask-Python-Financial-App)

### **39. Movie Recommendation System**

* **Description**: Create a recommendation system for movies using collaborative filtering.
* **GitHub Repository**: [Movie Recommendation](https://github.com/susanli2016/Machine-Learning-with-Python)

### **40. AI-Based Recipe Generator**

* **Description**: Build a system that suggests recipes based on available ingredients.
* **GitHub Repository**: [Recipe Generator](https://github.com/ozlerhakan/recipe-suggester)

**Tips for Success:**

1. Use cloud platforms like **Google Colab** for free GPU resources.
2. **Read research papers** on the project topics for deeper insights.
3. Contribute to **open-source projects** for practical exposure.
4. Document your work on **GitHub** with clear explanations to showcase your portfolio.

Let me know if you need help with any specific project! 😊

Road-Map

### **1. Kiến thức nền tảng toán học và thống kê**

## Đại số tuyến tính: Matrix Multiplication, Eigenvalues/Eigenvectors, Singular Value Decomposition (SVD).

## Xác suất và thống kê: Bayes' Theorem, Gaussian Distribution, Hypothesis Testing.

## Tối ưu hóa: Gradient Descent, Stochastic Gradient Descent, Adam Optimizer.

## Lý thuyết thông tin: Entropy, Cross-Entropy, KL Divergence.

## 

### **2. Kiến thức nâng cao về AI và Machine Learning**

#### **Machine Learning nâng cao:**

## Bayesian Learning.

## Reinforcement Learning: Q-Learning, Deep Q-Networks (DQN).

## Semi-Supervised Learning.

## Self-Supervised Learning (SSL).

#### **Deep Learning nâng cao:**

## Transformer Architecture: Multi-Head Attention, Encoder-Decoder Framework.

## Zero-shot learning, Few-shot learning.

## Advanced Generative Models: StyleGAN, Diffusion Models (DALL-E, Stable Diffusion).

## Meta-Learning và AutoML.

#### **NLP nâng cao:**

## Retrieval-Augmented Generation (RAG).

## Semantic Search.

## Pre-training và Prompt Engineering.

## Fine-tuning mô hình với Low-Rank Adaptation (LoRA).

## 

### **3. Công nghệ Big Data và xử lý dữ liệu quy mô lớn**

## **Hadoop**: HDFS, MapReduce.

## **Apache Spark**: PySpark cho xử lý dữ liệu phân tán.

## **Data Lakes**: Hiểu về các hệ thống lưu trữ dữ liệu lớn (Delta Lake, Snowflake).

## **ETL Pipelines**: Extract, Transform, Load (Airflow, Luigi).

## 

### **4. Kiến thức về hệ thống phân tán**

## Distributed Computing: RPC, gRPC.

## Message Queues: Kafka, RabbitMQ.

## Microservices Architecture.

## Load Balancing, High Availability.

## 

### **5. Công cụ và nền tảng AI tiên tiến**

## **Vector Databases**: Pinecone, Weaviate, Milvus (cho Semantic Search và ứng dụng LLM).

## **MLOps Pipelines**: TFX (TensorFlow Extended), Seldon Core.

## **Data Versioning**: DVC, Delta Tables.

## **Model Deployment**: Triton Inference Server, TorchServe.

## **Experiment Tracking**: Weights & Biases (W&B).

## 

### **6. Kỹ năng triển khai và tối ưu hóa mô hình AI**

## Hiểu về các chiến lược tối ưu hóa như Quantization, Pruning, Distillation.

## Tối ưu hóa GPU/TPU: CUDA, cuDNN, TensorRT.

## Distributed Training với Horovod, DeepSpeed.

## 

### **7. Ứng dụng AI trong thực tế**

#### **Computer Vision:**

## Image/Video Captioning.

## Facial Recognition.

## 3D Reconstruction và Augmented Reality (AR).

#### **NLP:**

## Summarization, Chatbots (Rasa, Dialogflow).

## Machine Translation.

## Document Understanding và Extraction (PDF, Scanned Images).

#### **Robotics:**

## SLAM (Simultaneous Localization and Mapping).

## Reinforcement Learning cho Robotics.

## 

### **8. Kỹ năng lập trình nâng cao**

## Xử lý concurrency/multithreading trong Python.

## Hiểu các cấu trúc dữ liệu nâng cao: Trie, Graphs, Bloom Filters.

## Performance Optimization trong Python.

## 

### **9. Tư duy và kỹ năng mềm**

## Khả năng lãnh đạo kỹ thuật (Technical Leadership).

## Hiểu về Ethical AI: Responsible AI, Fairness, Bias Detection.

## Kỹ năng viết tài liệu kỹ thuật rõ ràng và dễ hiểu.

## 

### **10. Các xu hướng AI mới nhất trong năm 2025**

## Edge AI: Triển khai AI trên thiết bị IoT, Mobile (TensorFlow Lite, CoreML).

## AI for Healthcare: Ứng dụng AI trong chẩn đoán y khoa, phân tích hình ảnh y tế.

## Quantum Machine Learning: Kết hợp AI và Quantum Computing (Qiskit).

## 

### **11. Cộng đồng và phát triển bản thân**

## Tham gia các dự án mã nguồn mở: Huggingface, PyTorch Community.

## Viết blog chia sẻ kiến thức AI/ML (Medium, GitHub Pages).

## Tham dự hackathon hoặc các cuộc thi AI như Kaggle, Zindi.

## 

**More Details**

## 

## 

## **1. Kiến thức nền tảng toán học và thống kê**

### **Cần nắm:**

* **Đại số tuyến tính**:
  + Matrix Multiplication (nhân ma trận), Inverse Matrices.
  + **Ứng dụng**: Trong Neural Networks (forward và backpropagation).
  + **Tài liệu**: “Linear Algebra and Its Applications” - Gilbert Strang.
* **Xác suất và thống kê**:
  + Gaussian Distribution, Bayes' Theorem, Markov Chains.
  + **Ứng dụng**: Hồi quy Logistic, Bayesian Inference, NLP.
  + **Khóa học**: “Probabilistic Machine Learning” - edX.
* **Tối ưu hóa**:
  + Gradient Descent, Adam Optimizer, Regularization (L1, L2).
  + **Ứng dụng**: Huấn luyện mô hình, giảm overfitting.
* **Lý thuyết thông tin**:
  + Entropy, Cross-Entropy Loss, KL Divergence.
  + **Ứng dụng**: Loss Function trong Classification.

## **2. Kiến thức nâng cao về AI và Machine Learning**

### **Machine Learning nâng cao:**

* **Kỹ thuật nâng cao**:
  + Semi-Supervised Learning, Self-Supervised Learning.
  + Ensemble Learning (Bagging, Boosting, Random Forest, XGBoost).
  + **Ứng dụng**: Kaggle Competitions, tối ưu mô hình với ít dữ liệu.
* **Khóa học khuyên dùng**:
  + “Machine Learning Specialization” - Coursera (Andrew Ng).

### **Deep Learning nâng cao:**

* **Kỹ thuật kiến trúc Deep Learning**:
  + CNN (cho Computer Vision), RNN/LSTM (cho chuỗi thời gian và NLP).
  + Transformers: Hiểu Attention Mechanism và Multi-Head Attention.
  + **Ứng dụng**: GPT, BERT, Stable Diffusion.

### **Thực hành mô hình NLP và Deep Learning:**

* **Các mô hình cụ thể**:
  + GPT-4: Xây dựng chatbot.
  + BERT: Tìm kiếm ngữ nghĩa (semantic search).
  + GAN: Tạo ảnh (image generation).
* **Framework phổ biến**: Huggingface, PyTorch, TensorFlow.

## **3. Công nghệ Big Data và xử lý dữ liệu quy mô lớn**

### **Kỹ năng cần thiết:**

* **Hadoop**: Lưu trữ và xử lý dữ liệu lớn (HDFS).
* **Spark (PySpark)**:
  + Xử lý dữ liệu phân tán.
  + **Ứng dụng**: Tiền xử lý dữ liệu lớn trước khi training.
* **Data Lakes**: Snowflake, Delta Lake.

### **Công cụ ETL Pipeline:**

* **Airflow**: Tự động hóa pipeline.
* **Luigi**: Xây dựng pipeline phức tạp.

## **4. Kiến thức về hệ thống phân tán và triển khai**

### **Hệ thống phân tán:**

* **Distributed Computing**:
  + RPC, gRPC cho giao tiếp giữa các dịch vụ.
* **Message Queues**: Kafka, RabbitMQ.

### **Microservices:**

* Tích hợp AI vào kiến trúc microservices.
* Sử dụng Docker và Kubernetes để triển khai.

### **Load Balancing:**

* **Công cụ**: Nginx, Traefik.
* **Ứng dụng**: Triển khai mô hình trên nhiều server.

## **5. Kiến thức về MLOps và DevOps**

### **Pipeline MLOps:**

* **MLflow**: Quản lý thí nghiệm và theo dõi mô hình.
* **Kubeflow**: Xây dựng pipeline huấn luyện mô hình.

### **Containerization và CI/CD:**

* **Docker**: Tạo container cho môi trường AI.
* **Kubernetes**:
  + Quản lý container ở quy mô lớn.
  + **Ứng dụng**: Auto-scaling cho mô hình AI.

### **Monitoring:**

* **Prometheus + Grafana**: Theo dõi hiệu suất mô hình sau triển khai.

## **6. Kỹ năng đánh giá và tối ưu hóa mô hình**

### **Đánh giá mô hình:**

* **Hiệu suất**: Accuracy, Precision, Recall, F1-score.
* **ROC và AUC**: Đánh giá phân loại nhị phân.

### **Tối ưu hóa mô hình:**

* **Quantization**: Giảm kích thước mô hình (cho Edge AI).
* **Pruning**: Loại bỏ các nút không cần thiết của Neural Network.
* **Distillation**: Huấn luyện mô hình nhỏ từ mô hình lớn.

## **7. Công cụ AI tiên tiến**

### **Model Deployment:**

* Triton Inference Server.
* TorchServe (cho mô hình PyTorch).

### **Vector Databases:**

* Pinecone, Weaviate, Milvus (cho Semantic Search).

### **Experiment Tracking:**

* Weights & Biases (W&B).

## **8. Ứng dụng AI trong thực tế**

### **Computer Vision:**

* **Nhiệm vụ phổ biến**: Image Classification, Object Detection, Semantic Segmentation.
* **Thư viện**: OpenCV, torchvision.

### **NLP:**

* Summarization (Tóm tắt văn bản).
* Semantic Search với Retrieval-Augmented Generation (RAG).
* Fine-tuning BERT hoặc GPT.

## **9. Xu hướng AI mới nhất**

### **Generative AI:**

* GANs: StyleGAN, BigGAN.
* Diffusion Models: Stable Diffusion, DALL-E.

### **Edge AI:**

* **TensorFlow Lite**, **CoreML**: Chạy AI trên Mobile hoặc IoT.

## **10. Tư duy và kỹ năng mềm**

* **Tư duy Ethical AI**: Xây dựng AI có đạo đức (Fairness, Bias Detection).
* **Kỹ năng viết tài liệu**: Thuyết phục và rõ ràng.

AI-Skills-Extract-From-Jobs

# 

# Skills Necessary For AI - Engineer: NLP - Generative AI

1. RMSE
2. MAE
3. Accuracy
4. ROC
5. AUC
6. Precision
7. Recall
8. Prometheus, Grafana
9. Hiểu biết về MLOps cho mô hình học máy
10. Kinh nghiệm tùy chỉnh và fine-tuning mô hình AI cho các ứng dụng cụ thể
11. Kỹ năng giải quyết vấn đề và tư duy phân tích tốt
12. Python
13. Machine Learning
14. Deep Learning
15. Pandas, NumPy, Scikit-learn
16. Matplotlib, Seaborn
17. Tensorflow
18. Pytorch
19. NLP
20. spaCy, NLTK, or transformers
21. NLU
22. NER
23. Computer Vision
24. AWS, Google Cloud, Azure
25. Langchain
26. LLamaIndex
27. Docker
28. Kubernetes
29. Docker, Kubernetes và các công cụ Containerization
30. CI/CD/DevOps
31. TensorRT
32. Linux
33. ONXX Runtime
34. LLMs
35. Huggingface
36. BERT, GPT
37. SQL(SQLServer)
38. NoSQL
39. SQL Server, PostgreSQL, SQLite …
40. Transfer Learning
41. GPT, Transformer, GAN,...
42. FastAPI
43. REST APIs
44. Web Services
45. Proficiency in data programming languages such as Python, R, Java, C#, and tools like Jupyter, VS Studio Code
46. Experience in NLP, computer vision, TensorFlow, PyTorch, JAX and CUDA toolkit.
47. Machine Learning, MLOps, và LLM (Large Language Models) để xây dựng các giải pháp AI Agent
48. Kiến thức vững về Machine Learning, Deep Learning, NLP và các mô hình LLM (GPT, BERT, T5, v.v.).
49. Kinh nghiệm với các công cụ MLOps như MLflow, Kubeflow, TensorFlow, PyTorch, hoặc các nền tảng triển khai mô hình quy mô lớn.

Jobs-AI

# [**AI Engineer**](https://www.topcv.vn/tim-viec-lam-ai-engineer) **- NLP (Natural Language Processing)**

* Design, develop, and implement AI models focused on **NLP, NLU, and NER** to process, analyze, and extract meaningful insights from related data.
* Monitor, debug, and improve deployed AI systems to ensure the performance and reliability standards.
* Continuously **fine-tune** models to improve performance, scalability, and accuracy in real-world scenarios.
* Stay updated on industry trends, emerging technologies, and advancements in AI to recommend and implement innovative solutions.
* Bachelor’s degree in Computer Science, Artificial Intelligence, Data Science, or a related field.
* 3-5 years experience with AI technologies, strong emphasis on NLP, NLU, or NER projects.
* Proficiency in **Python** and frameworks such as **TensorFlow, PyTorch.**
* Experience with **NLP libraries like spaCy, NLTK, or transformers.**
* Familiarity with **large-scale data** **preprocessing, annotation, and training pipelines.**
* Understanding of database systems (**SQL/NoSQL**) and data structures for model integration.
* Strong problem-solving skills and the ability to work in an independent environment with proactive attitude and responsibility.

# [**Kỹ Sư AI**](https://www.topcv.vn/tim-viec-lam-ky-su-ai)

- Cài đặt, triển khai các thuật toán AI mới

- Giải thích, chỉ dẫn giúp level junior hiểu được ý tưởng thuật toán và cách thức xây dựng, sử dụng model

- Tham gia các session phân tích về yêu cầu nghiệp vụ / dữ liệu để hiểu rõ vấn đề cần giải quyết, các nguồn dữ liệu có thể tận dụng, cách thức model tích hợp vào hệ thống, để nâng cao hiệu quả model và xây dựng model với đầu vào / đầu ra phù hợp

- Kiểm tra hiệu năng, độ chính xác của các model. Diễn giải kết quả model cho DA/BA một cách trực quan, dễ hiểu.

- Đánh giá kết quả sau tác động, đề xuất / thực hiện các phương pháp phân tích / tối ưu model.

- Trình độ **Tiếng Anh** Toeic 650 trở lên hoặc tương đương

- Hoàn thành chứng chỉ **Google Tensorflow Developer**

Ưu tiên: Đã tham gia xây dựng, triển khai >= 2 sản phẩm, giải pháp AI và đóng vai trò định hướng chính >= 1 giải pháp / sản phẩm, chứng minh phương pháp / định hướng có hiệu quả

# [**AI Engineer**](https://www.topcv.vn/tim-viec-lam-ai-engineer)

### **Mô tả công việc**

### Responsibility:

* Responsible for improving AI models and features within the akaBot ecosystem
* Maintained an end-to-end **pipeline** integrating AI models on cloud infrastructure (**AWS EKS, AWS Lambda, etc.)**
* Participate in the development and integration of core AI into akaBot's products

Key activities:

* Research AI/ML algorithms/models and other algorithms to solve the **RPA problem**, take part in the development process to bring the research result into current product
* Experience in model optimization techniques such as model distilation, model prunning, TensorRT frameworkStrong klowledge of data structures, proficifient at algorithm is a strong plus
* Familiar with model deployment techniques, Web services such as REST, WebRTCResearch and analyze AI/ML services from top of the world products to bring those techniques into akabot product
* Build and integrate new AI/ML services to existing company product
* Experience in deep learning/machine learning development
* Experience in object detection/image classification/segmentation with CNN/RNN models
* Experience in data/image processing techniquesHaving ability to bring the research results (model, pineline, ...), insight of AI product development

### **Yêu cầu ứng viên**

* Bachelor’s Degree in Information Technology, Software Engineering, Computer Science, or related field
* At least 3 years in field of AI, Machine Learning, Computer VisionExperience using frameworks **(Pytorch, Tensorflow, MxNet, Caffe, etc.)**
* Experience with and multiple libraries **(NumPy, OpenCV, Scikit, Pandas, etc.)**
* Experience with Python or other programming languages **(MATLAB, Java, C#,..)**
* Experience in software development is a strong plus
* Experience in **MLOPs** and model serving is a strong plus
* Strong understanding in algorithm and data structureExcellent troubleshooting and communication skills
* **Basic English communication**

# [**Kỹ Sư**](https://www.topcv.vn/tim-viec-lam-ky-su) **Trí Tuệ Nhân Tạo - AI**

## **Chi tiết tin tuyển dụng**

Gửi tôi việc làm tương tự

[Chuyên môn AI Engineer](https://www.topcv.vn/tim-viec-lam-ai-engineer-cr257cb260cl283)

### **Mô tả công việc**

**- Nhiệm vụ của vị trí:**

+ Nghiên cứu xây dựng các bài toán áp dụng công nghệ ML, DL trên các dữ liệu về ảnh, text, audio, video, ....

+ Nghiên cứu xây dựng các luồng xử lý theo chuẩn AI/ML OPs.

+ Thực hiện quy trình huấn luyện cho mô hình AI đang có.

+ Đảm bảo chất lượng dữ liệu, kết quả mô hình, báo cáo định kỳ tự động

+ Chịu trách nhiệm bảo trì, cải thiện, làm sạch và thao tác dữ liệu, khắc phục các sự cố tồn tại

+ Tối ưu hóa CSDL và duy trì nâng cấp các chuẩn kiến trúc của dữ liệu

+ Xây dựng các tài liệu Thiết kế tổng thể, giải pháp, Thiết kế chi tiết, tài liệu Giám sát, theo quy trình DS/AI

**- Vai trò và trách nhiệm:**

+ Nghiên cứu, thiết kế, cài đặt các thuật toán, mô hình học máy.

+ Thiết kế, triển khai các thuật toán, mô hình học máy để giải quyết các bài toán, yêu cầu đề ra.

+ Tối ưu, cải thiện các giải pháp đã được triển khai.

+ Hợp tác với các nhóm khác để tích hợp giải pháp."

### **Yêu cầu ứng viên**

**1. Yêu cầu kiến thức:**

- Kiến thức về lập trình, cấu trúc dữ liệu và giải thuật, lý thuyết đồ thị.

- Kiến thức về xác suất thống kê, đại số tuyến tính, giải tích.

- Kiến thức về phân tích khám phá dữ liệu (Exploratory Data Analysis), học máy (Machine Learning, Deep Learning), trực quan hóa dữ liệu.

- Kiến thức về phát triển phần mềm: quản lý phiên bản, cơ sở dữ liệu, APIs, các nguyên lý lập trình, thiết kế và phát triển phần mềm.

- Kiến thức về luồng xử lý trong quy trình ML/AI OPs

**2. Yêu cầu kỹ năng:**

- Kỹ năng chuyên môn

+ Kỹ năng lập trình thành thạo Python, Java, SQL.

+ Kỹ năng đọc bài báo khoa học và lập trình lại các thuật toán, giải pháp đề xuất trong bài báo khoa học.

+ Kỹ năng trực quan hóa dữ liệu.

+ Kỹ năng phân tích và đánh giá thuật toán, mô hình học máy.

+ Kỹ năng triển khai với các nền tảng containerization (Docker, Kubernetes) là một điểm cộng.

- Kỹ năng con người (Personal Skills)

+ Kỹ năng giải quyết vấn đề.

+ Tư duy dựa trên dữ liệu và tập trung vào phát triển sản phẩm.

+ Kỹ năng trình bày (viết, nói) vấn đề, ý tưởng, giải pháp.

**3. Kinh nghiệm:**

- Kinh nghiệm 1-2 năm trong nghiên cứu, phát triển và triển khai thuật toán, mô hình học máy.

- Kinh nghiệm thiết kế, phát triển, triển khai phần mềm trong thực tế.

- Có kinh nghiệm trong xây dựng kiến trúc, phát triển, triển khai các giải pháp phân tích dữ liệu là một điểm cộng.

**4. Trình độ học vấn/Chuyên môn có Liên quan:**

- Tốt nghiệp từ loại Khá đại học chuyên ngành Công nghệ thông tin, khoa học máy tính hoặc các ngành STEM khác (Toán-Tin, Toán,...).

- Tiếng Anh theo chuẩn TOEIC 550 hoặc tương đương.

- Có chứng chỉ về Data Science là một điểm cộng (IBM Data Science Professional, Tableau Certified, Google Certified,... ).

### **1. Kiến thức hiện tại của mày có quá thiếu không?**

**Tóm lại: Không quá thiếu, nhưng chưa đủ để có lợi thế mạnh khi xin việc AI sau khi ra trường.**

Mày đã có nền tảng tốt về **Machine Learning (ML), Deep Learning (DL), và NLP**, nhưng vẫn còn một số lỗ hổng quan trọng:

#### **Những gì mày đã có (tốt)**

✅ **Lập trình:** Thành thạo **Python, C++, Java** → Phù hợp với AI.  
✅ **Machine Learning & Deep Learning:** Biết dùng **PyTorch, Scikit-learn**, đã làm vài dự án nhỏ.  
✅ **NLP:** Có một dự án về **dịch máy (Machine Translation)**.  
✅ **Mô hình phát triển phần mềm:** Biết về **MVC, CI/CD**, giúp mày làm AI theo quy trình chuẩn.  
✅ **Git, GitHub, SQL Server:** Quan trọng khi làm AI/ML thực tế.  
✅ **Chứng chỉ AI:** **IBM AI Developer**, **Software Development Lifecycle** → Tốt để CV có điểm nhấn.

#### **Những gì còn thiếu (quan trọng)**

🔥 **Thiếu kiến thức về Data Engineering & MLOps**

* AI thực tế không chỉ là build model, mà còn phải **thu thập, làm sạch, lưu trữ dữ liệu** → Mày cần học thêm **Pandas, Spark, Airflow, Kubernetes**.
* MLOps giúp deploy mô hình AI **(TensorFlow Serving, FastAPI, Docker, Kubernetes)**.

🔥 **Thiếu dự án AI chuyên sâu**

* Các project của mày vẫn còn **khá cơ bản**, chỉ dừng ở mức áp dụng mô hình có sẵn (Google Translate API, Stable Diffusion).
* **Cần làm thêm dự án nâng cao**: Fine-tune Transformer (BERT, GPT), Chatbot RAG, AI tự động tạo nội dung.

🔥 **Thiếu kinh nghiệm thực tế (Internship hoặc Freelance)**

* AI không giống lập trình web, hầu hết nhà tuyển dụng cần **kinh nghiệm thực chiến** với dữ liệu lớn.
* Nếu chưa có internship, mày nên tìm ngay trên **LinkedIn, TopCV, hoặc làm freelance AI/NLP**.

🔥 **Chưa có đóng góp Open Source hoặc Kaggle Competition**

* Các công ty AI rất thích ứng viên có đóng góp vào **Hugging Face, TensorFlow, PyTorch** hoặc từng tham gia **Kaggle**.

### **2. Khả năng tìm được việc AI sau khi ra trường?**

**Tóm lại: Có thể tìm được việc, nhưng phải chuẩn bị kỹ hơn để có lợi thế cạnh tranh.**

#### **Tình hình thị trường AI hiện tại**

* AI Engineer đang rất hot, nhưng cũng **cạnh tranh cao**, nhất là khi nhiều người có **thạc sĩ AI hoặc kinh nghiệm từ sớm**.
* Nhiều công ty Việt Nam (VNG, VinAI, FPT AI, Zalo AI) đang tuyển AI Engineer, nhưng yêu cầu phải có **dự án thực tế mạnh hoặc nghiên cứu chuyên sâu**.
* Nếu không có internship hoặc dự án tốt, mày sẽ phải cạnh tranh với nhiều ứng viên giỏi hơn.

#### **Làm sao để đảm bảo tìm được việc AI?**

🔹 **Tìm internship AI ngay bây giờ**

* Nếu ra trường mà chưa có kinh nghiệm AI thực tế, sẽ rất khó cạnh tranh.
* Apply các vị trí **AI Engineer Intern, Data Science Intern** ở FPT AI, Zalo AI, VNG, Shopee, VinAI.
* Hoặc làm **freelance AI/NLP** trên Upwork, Fiverr, Freelancer.

🔹 **Xây dựng portfolio AI mạnh trên GitHub**

* Không chỉ làm dự án nhỏ, mà phải làm dự án **có dataset lớn, fine-tune mô hình từ đầu, có API, có UI**.
* Ví dụ:  
  ✅ **Chatbot Y tế** (Fine-tune GPT-3.5/BERT với dữ liệu bệnh án).  
  ✅ **AI Viết bài tự động** (Tóm tắt tin tức, tạo bài viết).  
  ✅ **Nhận diện cảm xúc từ giọng nói** (Kết hợp NLP & Speech Recognition).

🔹 **Tham gia Kaggle hoặc Open Source**

* Làm **Kaggle Competitions** giúp mày chứng minh kỹ năng xử lý dữ liệu & tối ưu mô hình.
* Đóng góp **Pull Requests** cho Hugging Face hoặc PyTorch giúp CV nổi bật hơn.

🔹 **Networking với cộng đồng AI**

* Tham gia **AI Club, hội thảo AI** để có cơ hội việc làm tốt hơn.
* Kết nối với **những người trong ngành AI trên LinkedIn**.

### **3. Kết luận - Mày có đủ khả năng không?**

🔥 **CÓ, nếu mày tập trung và làm đúng hướng!** 🔥

* Mày đã có nền tảng tốt, nhưng để chắc chắn có việc AI khi ra trường, mày cần **nâng cấp dự án, có internship và đóng góp AI thực tế**.
* Nếu bắt tay vào làm ngay từ bây giờ, mày hoàn toàn có thể **có job AI tốt sau khi tốt nghiệp!** 🚀

### **Đánh giá CV của mày và lộ trình để trở thành chuyên gia AI/ML/NLP**

CV của mày hiện tại khá tốt với nền tảng kỹ thuật vững chắc, nhưng để trở thành **chuyên gia AI/ML/NLP**, mày cần một chiến lược rõ ràng để nâng cao kiến thức và kinh nghiệm thực tế.

## **1. Những điểm mạnh trong CV**

✅ **Kỹ năng kỹ thuật tốt**:

* Thành thạo **Python, Java, C/C++, JavaScript**.
* Có nền tảng về **Machine Learning & Deep Learning (PyTorch, Scikit-learn, Computer Vision, NLP)**.
* Biết sử dụng **Git, GitHub, GitLab (CI/CD)**.

✅ **Dự án liên quan đến AI**:

* **Text To Image** (Stable Diffusion) → Tốt cho Generative AI.
* **Machine Translation** (Google Translate API) → Có nền tảng NLP.
* **Website Đặt Phòng Khách Sạn** → Có kinh nghiệm Fullstack.

✅ **Chứng chỉ**:

* **IBM AI Developer Professional Certificate** (sắp có).
* **Software Development Lifecycle** → Hiểu quy trình phát triển phần mềm.
* **CertNexus Certified Ethical Emerging Technologist** (sắp có).

✅ **Tham gia AIClub** → Chứng tỏ mày quan tâm đến AI.

## **2. Những điểm cần cải thiện**

🔥 **Thiếu kinh nghiệm thực tế với dự án NLP lớn**

* Mày mới có một project **Machine Translation** (có vẻ chỉ dùng Google Translate API).
* Nếu muốn làm chuyên gia NLP, cần xây dựng các mô hình NLP từ đầu (Transformer, LLM như BERT, GPT).

🔥 **Chưa có bài báo khoa học hoặc blog AI/NLP**

* Nếu muốn trở thành chuyên gia, cần đóng góp kiến thức qua **Medium, GitHub Blog hoặc các hội nghị AI**.

🔥 **Chưa có kinh nghiệm làm việc thực tế trong ngành**

* Hãy **tìm internship hoặc freelance AI/NLP**.

## **3. Lộ trình để trở thành chuyên gia AI/ML/NLP**

### **Giai đoạn 1: Nâng cao kiến thức AI/NLP (3-6 tháng)**

🔹 **Học sâu về NLP**:

* **Speech & Text Processing**: Xây dựng chatbot, dịch ngôn ngữ, tóm tắt văn bản.
* **Transformer & LLM**: Hiểu rõ BERT, GPT, T5.
* **Libraries**: Hugging Face, TensorFlow/Keras.

🔹 **Tạo dự án chuyên sâu** (Open-source nếu có thể):

* **Chatbot y tế nâng cao** (kết hợp LLM và Retrieval-Augmented Generation - RAG).
* **Fine-tune GPT/BERT** trên dữ liệu tiếng Việt.
* **AI viết bài tự động** (tóm tắt báo, tạo nội dung).

### **Giai đoạn 2: Xây dựng thương hiệu cá nhân (6 tháng)**

🔥 **Chia sẻ kiến thức**:

* Viết blog AI/NLP trên Medium, GitHub Blog.
* Đóng góp vào các dự án AI/NLP trên GitHub.

🔥 **Đăng ký hội nghị AI**:

* ICML, NeurIPS, ACL, EMNLP (bắt đầu từ hội nghị nhỏ trước).

🔥 **Làm việc trong lĩnh vực AI**:

* Tìm **internship AI Engineer / NLP Engineer**.
* Hoặc **làm freelance AI** trên Upwork, Freelancer.

### **Giai đoạn 3: Chuyên môn hóa và hướng đến chuyên gia (1-2 năm)**

✅ **Nâng cấp kỹ năng bằng nghiên cứu AI chuyên sâu**. ✅ **Học thêm về MLOps, tối ưu mô hình NLP**. ✅ **Tham gia dự án AI thực tế tại công ty lớn**.

## **4. Tóm lại - Mày có thể trở thành chuyên gia AI/NLP không?**

🔥 **CÓ, nếu mày kiên trì và liên tục học hỏi**.

* Hiện tại mày **đang có nền tảng tốt**, nhưng để đạt **trình độ chuyên gia**, cần **dự án thực tế, nghiên cứu, và làm việc thực tế**.
* Bắt đầu ngay với **Fine-tune Transformer Models, tham gia các cuộc thi AI, và đóng góp cho cộng đồng AI**.

Linkin Follower

📌 Here’s my go-to list of AI blogs that help you learn something new:

1️⃣ [Chip Huyen](https://www.linkedin.com/in/chiphuyen/) – Practical insights on ML systems, production AI, and scaling AI in real-world applications.

2️⃣ [Eugene Yan](https://www.linkedin.com/in/eugeneyan/) – Covers practical ML, data science, and building AI systems that actually work in production.

3️⃣ Lil’Log by [Lilian Weng](https://www.linkedin.com/in/lilianweng/) – Deep dives into AI/ML research, reinforcement learning, and neural networks.

4️⃣ [Sebastian Raschka, PhD](https://www.linkedin.com/in/sebastianraschka/) – A mix of AI research, ML engineering, and Python implementations explained in depth.

5️⃣ [Simon Willison](https://www.linkedin.com/in/simonwillison/) – Focuses on LLMs, AI tooling, and making AI more accessible for developers.

6️⃣ Andrej Karpathy – One of the best minds in deep learning, with highly technical and philosophical takes on AI.

7️⃣ [Nathan Lambert](https://www.linkedin.com/in/natolambert/) ([Interconnects AI](https://www.linkedin.com/company/interconnects-ai/)) – Covers the latest AI research mainly focusing on Reinforcement Learning.

8️⃣ [Ethan Mollick](https://www.linkedin.com/in/emollick/) – AI’s impact on business, creativity, and society, with an engaging writing style.

9️⃣ Gwern – In-depth essays on AI, rationality, and long-term AI implications.

🔟 [Sebastian Ruder](https://www.linkedin.com/in/sebastianruder/) – One of the best blogs for NLP, multilingual AI, and AI research summaries.

Must-Follow AI Company Blogs

🔹 [Hugging Face](https://www.linkedin.com/company/huggingface/) – The latest in open-source AI, LLMs, and AI model releases.

🔹 [Anthropic](https://www.linkedin.com/company/anthropicresearch/) – Safety, interpretability, and responsible AI research.

🔹 [Sakana AI](https://www.linkedin.com/company/sakana-ai/) – A fresh take on AI model design and innovation.

🔹 [Sequoia Capital](https://www.linkedin.com/company/sequoia/) – AI trends and startup insights from a VC perspective.

🔹 a16z – AI, technology, and business trends from [Andreessen Horowitz](https://www.linkedin.com/company/a16z/).

🔹 [Google DeepMind](https://www.linkedin.com/company/googledeepmind/) – Cutting-edge AI research and breakthroughs.

🔹 [AI at Meta](https://www.linkedin.com/company/aiatmeta/) – LLMs, AI advances, and Meta’s AI research.

🔥 These blogs cut through the noise and provide real value. Which ones do you follow? Any great AI blogs I should add?

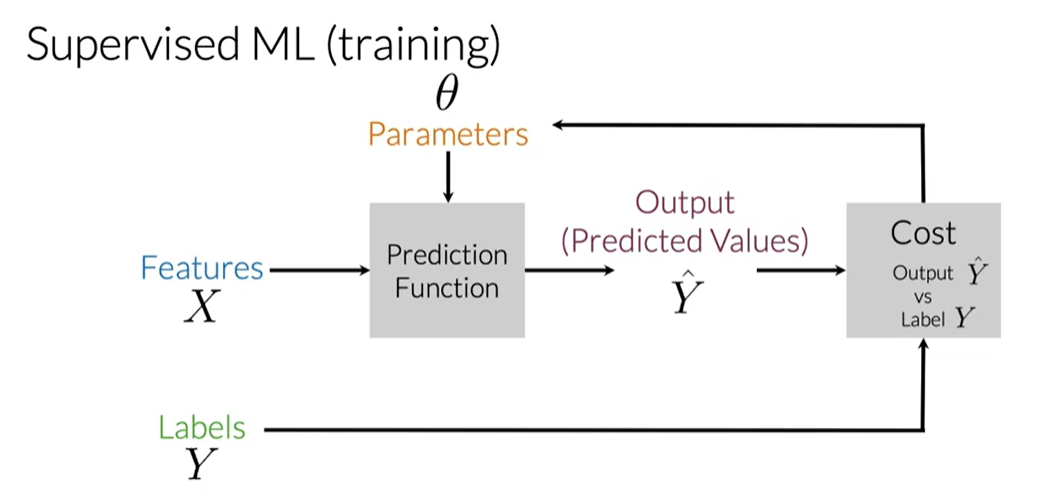
Let’s discuss in the comments! Links in the comments ⬇️

Study-AI

NLP

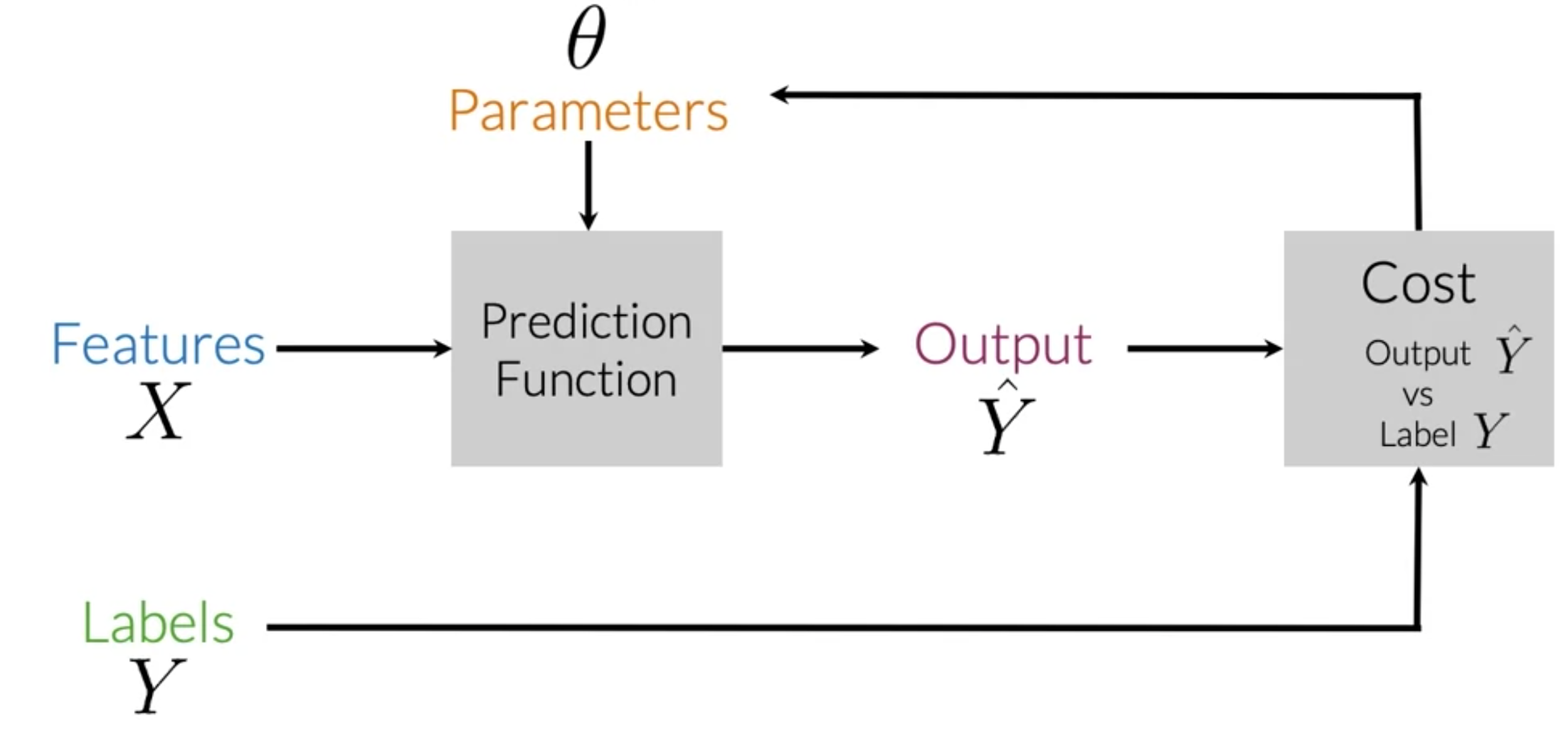
# **Natural Language Processing**

# Supervised ML

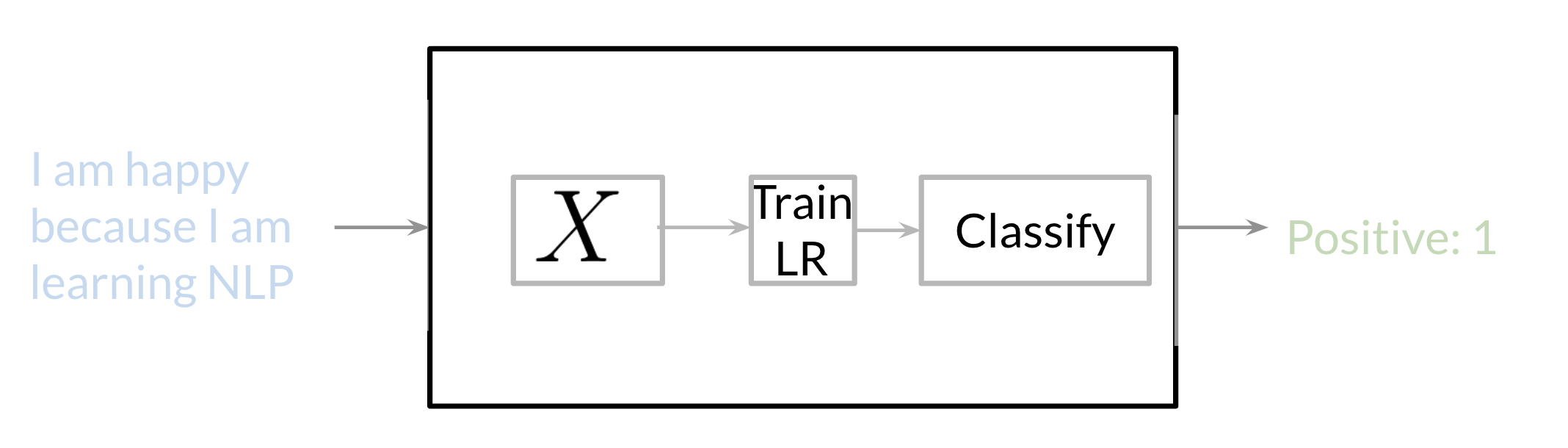


# **Supervised ML & Sentiment Analysis**

In supervised machine learning, you usually have an input *X*, which goes into your prediction function to get yourY. You can then compare your prediction with the true value *Y*. This gives you your cost which you use to update the parameters *θ*. The following image, summarizes the process.



To perform sentiment analysis on a tweet, you first have to represent the text (i.e. "I am happy because I am learning NLP ") as features, you then train your logistic regression classifier, and then you can use it to classify the text.

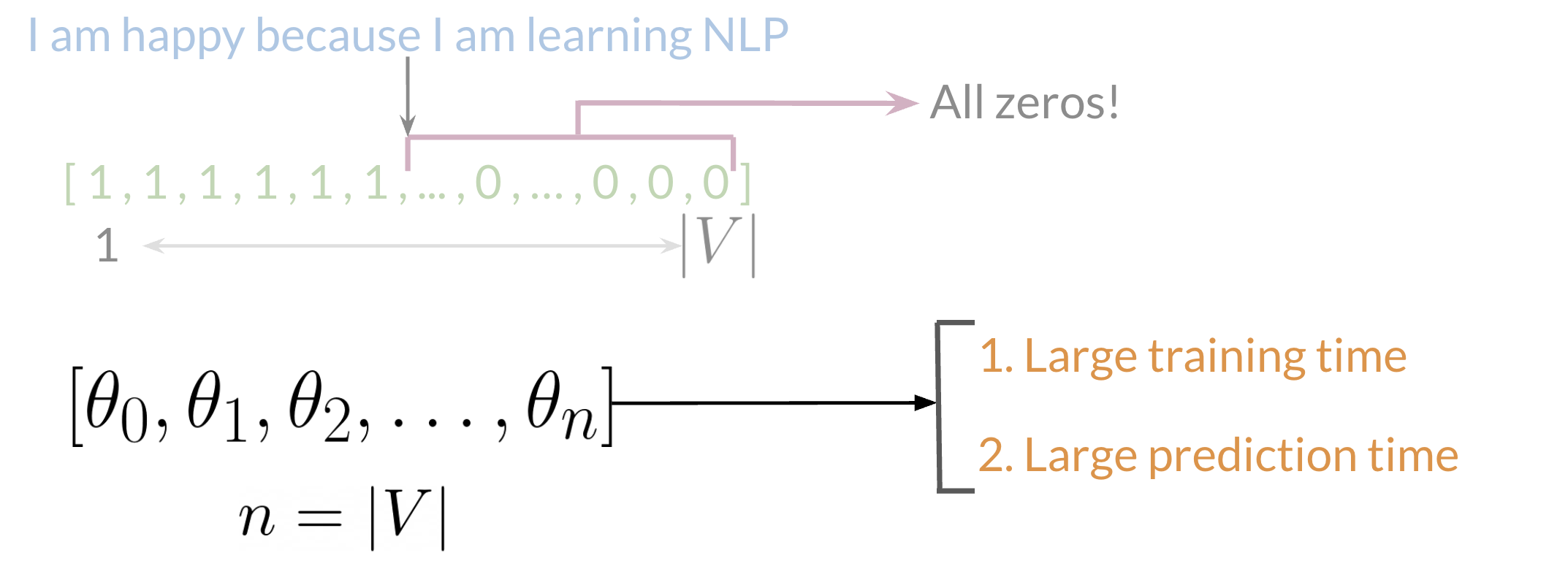


Note that in this case, you either classify 1, for a positive sentiment, or 0, for a negative sentiment.

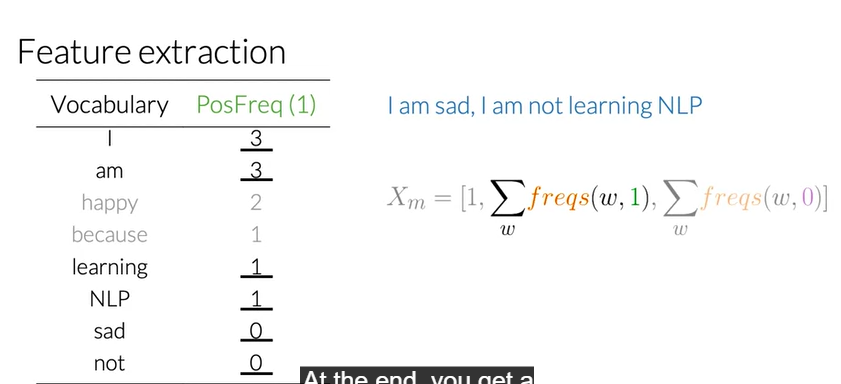
# **Vocabulary & Feature Extraction**

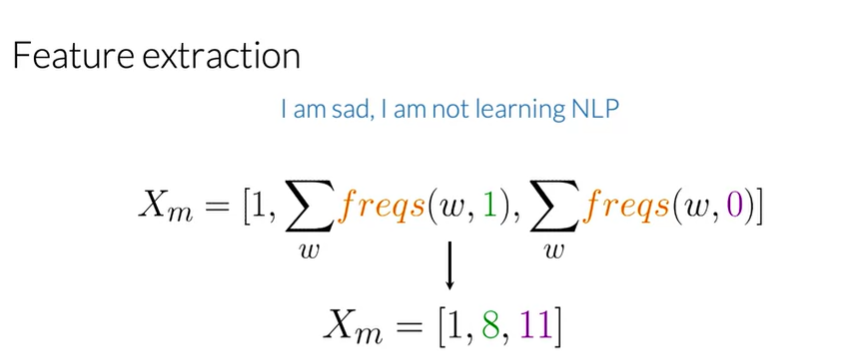
Given a tweet, or some text, you can represent it as a vector of dimension

*V*, where corresponds to your vocabulary size. If you had the tweet "I am happy because I am learning NLP", then you would put a 1 in the corresponding index for any word in the tweet, and a 0 otherwise.



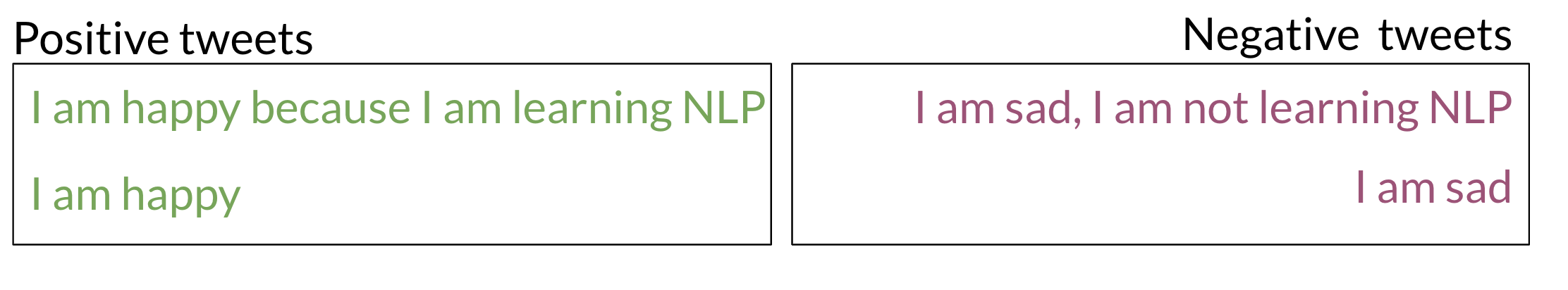
As you can see, as *V* gets larger, the vector becomes more sparse. Furthermore, we end up having many more features and end up training *V* parameters. This could result in larger training time, and large prediction time.





# **Feature Extraction with Frequencies**

Given a corpus with positive and negative tweets as follows:



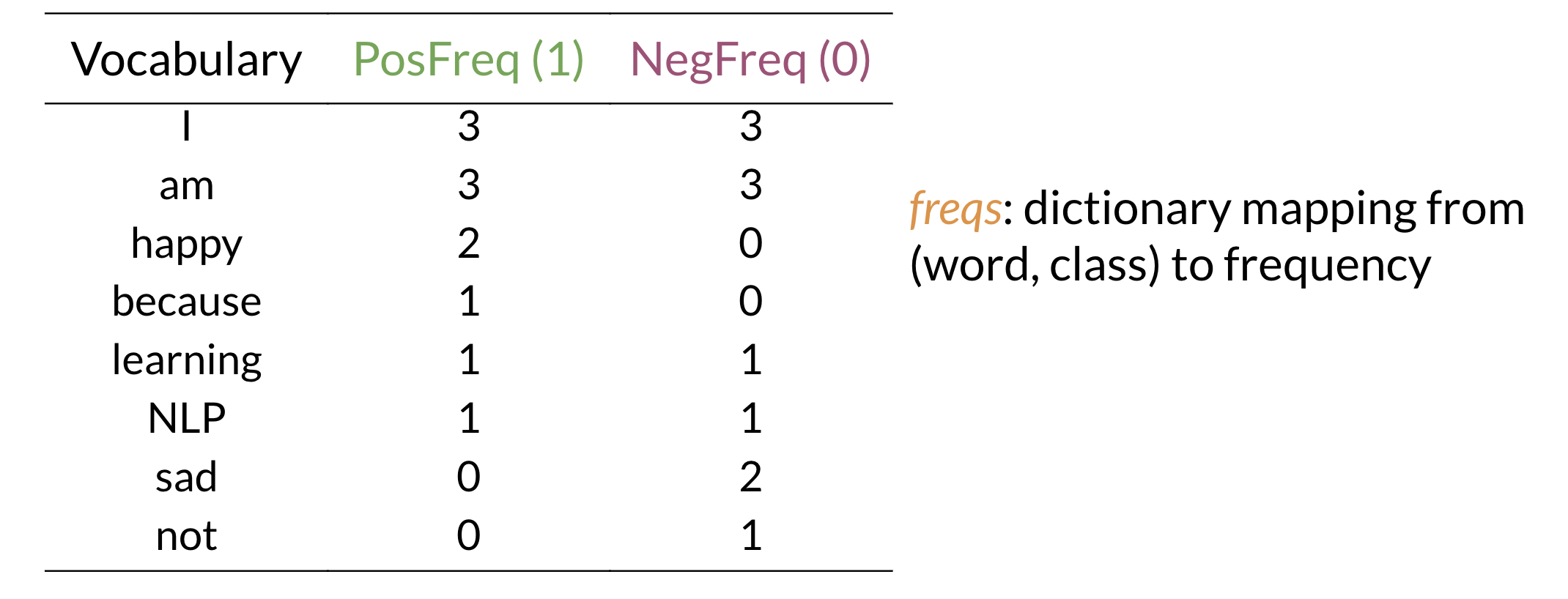
You have to encode each tweet as a vector. Previously, this vector was of dimension

V

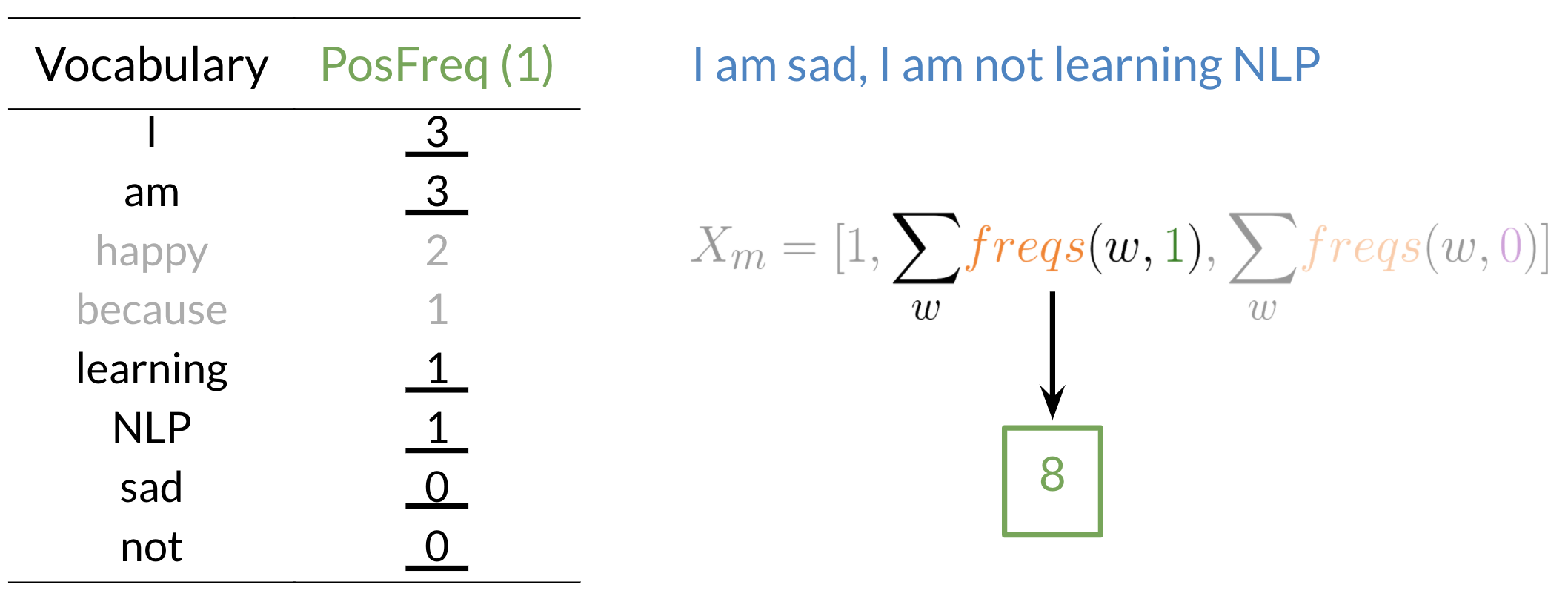
*V*. Now, as you will see in the upcoming videos, you will represent it with a vector of dimension

3

3. To do so, you have to create a dictionary to map the word, and the class it appeared in (positive or negative) to the number of times that word appeared in its corresponding class.



In the past two videos, we call this dictionary `freqs`. In the table above, you can see how words like happy and sad tend to take clear sides, while other words like "I, am" tend to be more neutral. Given this dictionary and the tweet, "I am sad, I am not learning NLP", you can create a vector corresponding to the feature as follows:



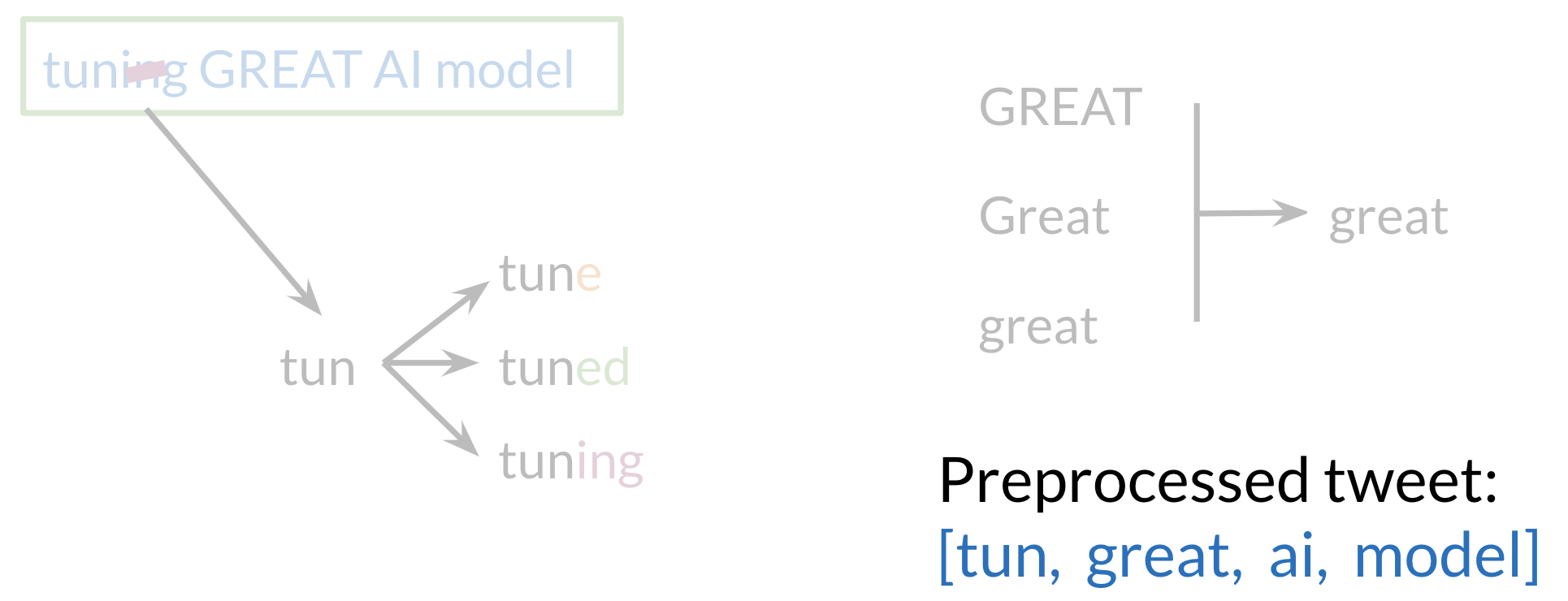
To encode the negative feature, you can do the same thing.

# **Preprocessing**

When preprocessing, you have to perform the following:

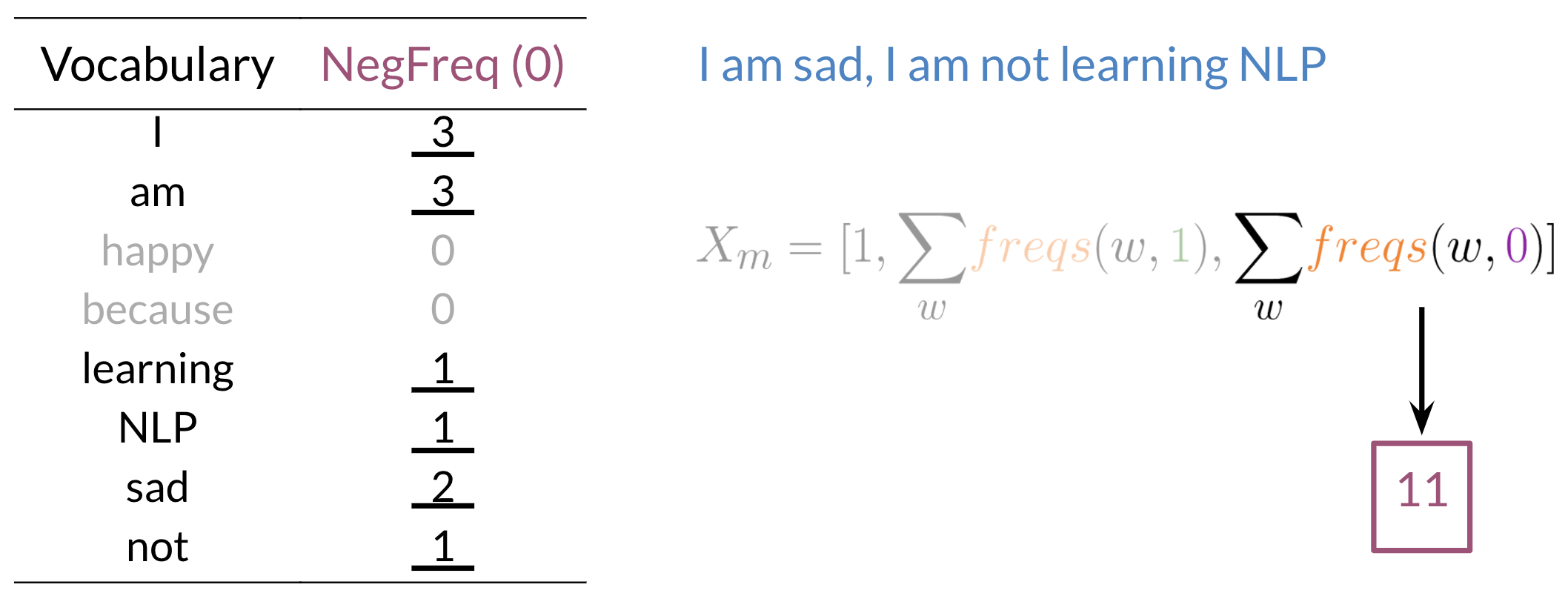
1. Eliminate handles and URLs
2. Tokenize the string into words.
3. Remove stop words like "and, is, a, on, etc."
4. Stemming- or convert every word to its stem. Like dancer, dancing, danced, becomes 'danc'. You can use porter stemmer to take care of this.
5. Convert all your words to lower case.

For example the following tweet "@YMourri and @AndrewYNg are tuning a GREAT AI model at https://deeplearning.ai!!!" after preprocessing becomes



[tun,great,ai,model]

[*tun*,*great*,*ai*,*model*]. Hence you can see how we eliminated handles, tokenized it into words, removed stop words, performed stemming, and converted everything to lower case.



Hence you end up getting the following feature vector

[1,8,11]

[1,8,11].

1

1 corresponds to the bias,

8

8 the positive feature, and

11

11 the negative feature.

# NLTK:

There are:

Dataset

Stopwords

Punctuation

Step to tokenization