
Part 1: What is Lean AI? [# Lean AI](#)

◆ Definition:

Lean AI is a minimalist, outcome-oriented approach to building AI systems. Inspired by Lean Startup principles, it emphasizes:

- **Rapid Prototyping**
- **Using Pretrained Models**
- **Avoiding Reinventing the Wheel**
- **Focusing on Real Value, Not Just Accuracy**

◆ Traditional AI vs. Lean AI:

Traditional AI	Lean AI
Train large models from scratch	Use pretrained models (fine-tune or prompt)
Requires big datasets and infrastructure	Small datasets, free tools like Colab
Long development cycles	Build & deploy MVPs in hours or days
Heavy focus on accuracy	Focus on usefulness and speed

Part 2: AI vs ML vs DL — Deep Dive

◆ Artificial Intelligence (AI)

Broad concept of machines acting intelligently.

- Examples: Siri, chess bots, self-driving cars

◆ Machine Learning (ML)

Subfield of AI where models learn from data.

- Types:
 - Supervised (labeled data)

- Unsupervised (find patterns)
- Reinforcement (learn by feedback)

◆ Deep Learning (DL)

Subset of ML using deep neural networks. Needs more data and compute.

- Use Cases: Image recognition, speech-to-text, language generation

✦ Real-World Tip: Most startup-ready AI products today don't need deep learning—they can use pretrained DL models with minimal code.

🔧 Part 3: AI Project Lifecycle (The Lean Way)

Step-by-step Lean Workflow:

1. Define the Problem Clearly

- What is the input?
- What's the expected output?
- Is AI actually needed?

2. Get a Small, Usable Dataset

- Use Kaggle, Hugging Face Datasets, scraping, or even manual data creation.

3. Use Pretrained or Classical Models

- NLP: Hugging Face Transformers
- Vision: CLIP, ViT, MobileNet
- Tabular: scikit-learn

4. Deploy Early

- Use Streamlit or Hugging Face Spaces
- Collect feedback

5. Iterate Fast

- Improve only based on user feedback
 - Track versions with GitHub
-

💻 Part 4: Tools & Setup Guide

💻 Coding Environment

1. Google Colab

- Free hosted Jupyter Notebook
- GPUs available (in free or Pro tier)
- Run Python in the browser
- Shareable links, auto-saving

2. Jupyter Notebook & uv Python package and project manager([page](#)) (Local)

- Great for offline use
- Requires Python and Jupyter installation
- Requires uv installation [uv](#)

3. Visual Studio Code (Advanced Option) [offline Extensions downloader](#)

- For power users who want full control
- Needs local Python environment

4. Git Version control system (VCS) [Install](#)

5. Orange

Python Libraries You'll Use Often:

Tool	Purpose
pandas	DataFrames, CSVs, data wrangling
numpy	Arrays, math operations
matplotlib / seaborn	Plotting, EDA
scikit-learn	Classic ML models (trees, regression, clustering)
transformers	NLP and vision models by Hugging Face
datasets	Access thousands of datasets easily
streamlit	Build AI web apps fast
gradio	Alternative to Streamlit for AI demos
openai	Access to GPT-4 and other LLMs

Tool	Purpose
torch or tensorflow	For low-level deep learning (optional in this course)

Part 5: Hands-On Lab

✓ Task 1: Launch a Google Colab Notebook

- Visit: colab.research.google.com
- Create a new notebook and name it: `LeanAI_Week1_Intro.ipynb`

✓ Task 2: Install Tools

Paste this into a cell:

```
!pip install pandas numpy scikit-learn matplotlib transformers
## or using uv
!uv add pandas numpy scikit-learn matplotlib transformers
```


✓ Task 3: Run a Sample ML Model

```
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LogisticRegression()
model.fit(X_train, y_train)

print("Model accuracy:", model.score(X_test, y_test))
```

 You've just trained and evaluated your first ML model in under 10 minutes!
