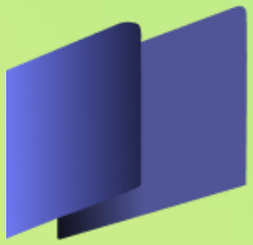
RAG-Based OpenAI Training



REVOLUTION
Data Platforms

RAG Based-OpenAI Training

- Introduction to Python
- Introduction to Generative AI with Azure Services
- Introduction to Azure AI Services Pt-1
- Introduction to Azure AI Services Pt-2
- LangChain Framework Pt-1
- LangChain Framework Pt-2
- RAG Essentials Pt-1
- RAG Essentials Pt-2

REVOLUTION
DATA PLATFORMS

Session-1: Introduction to Python Developers Guide

Shady A. Mohammed (S.A.M)

Proud Partner

Microsoft
Solutions Partner
Data & AI – Azure

databricks

snowflake

Profisee

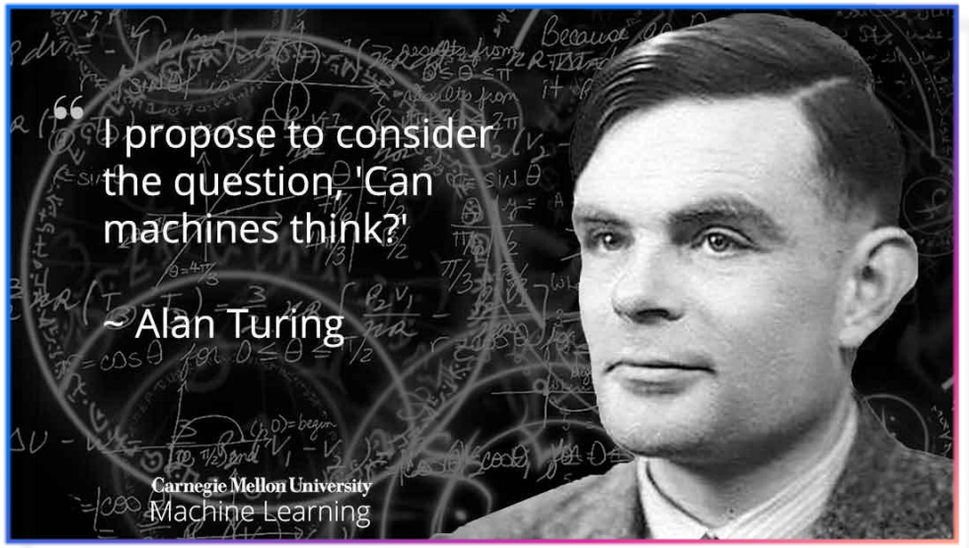
Agenda

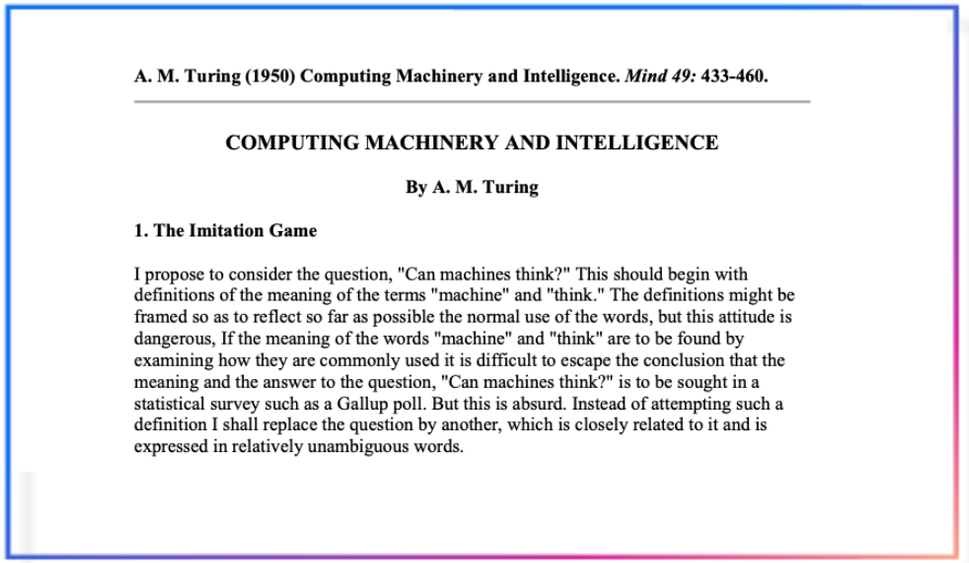
- Foundational Concepts
- Overview: Generative AI
- Python As Development Language
- Environment Setup
- External Package (Installation & Usage)

Foundational Concepts

History of Artificial Intelligence (AI)

- AI has its roots in the 1950s
- **Alan Turing** wondered if machines could think like humans
- This led to the development of early AI systems that could perform tasks that require human-like intelligence, such as understanding language.





A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

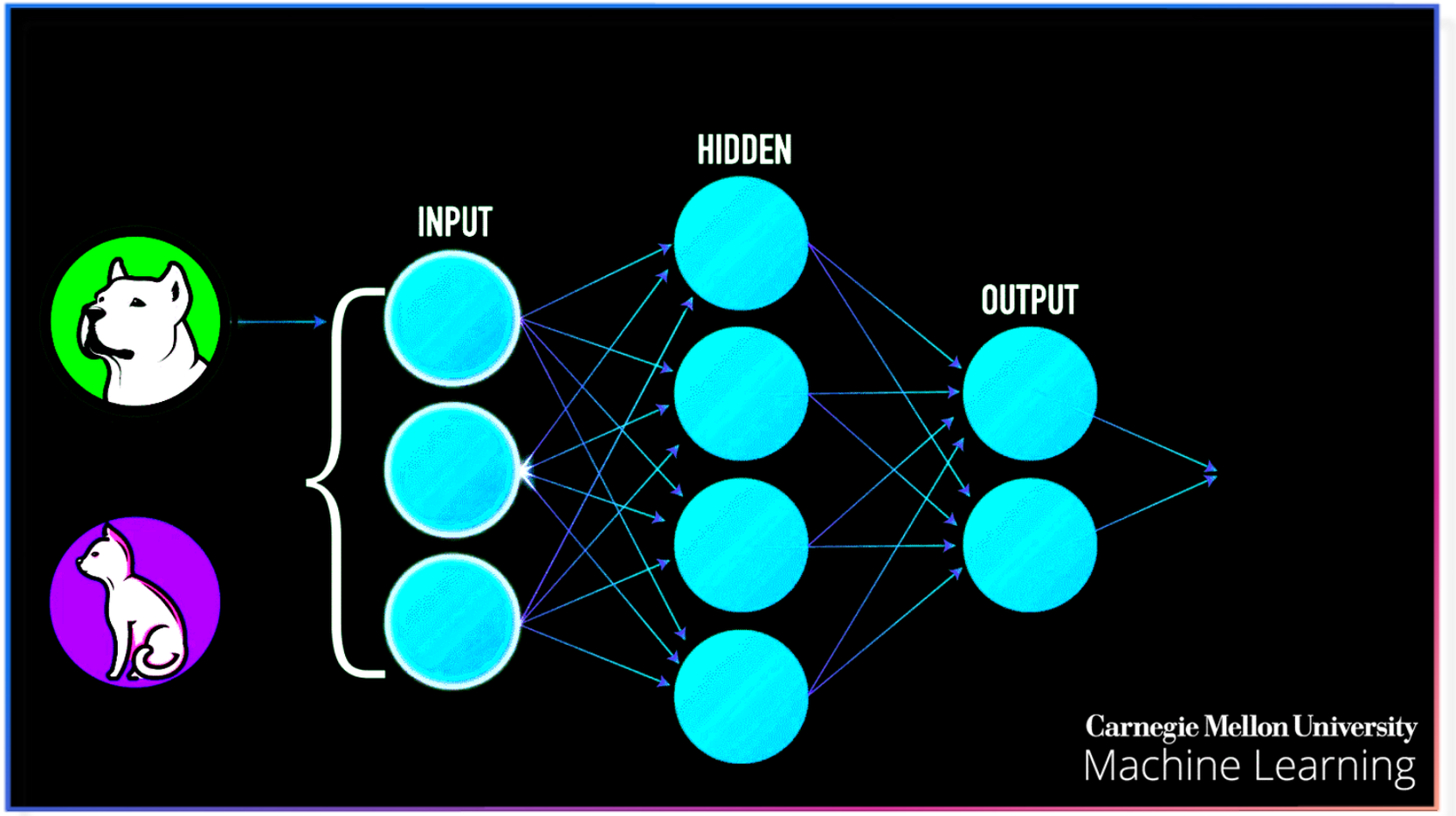
1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

How our Brain works?

Human Brain has neurons, either fire or rest to exchange Signals!

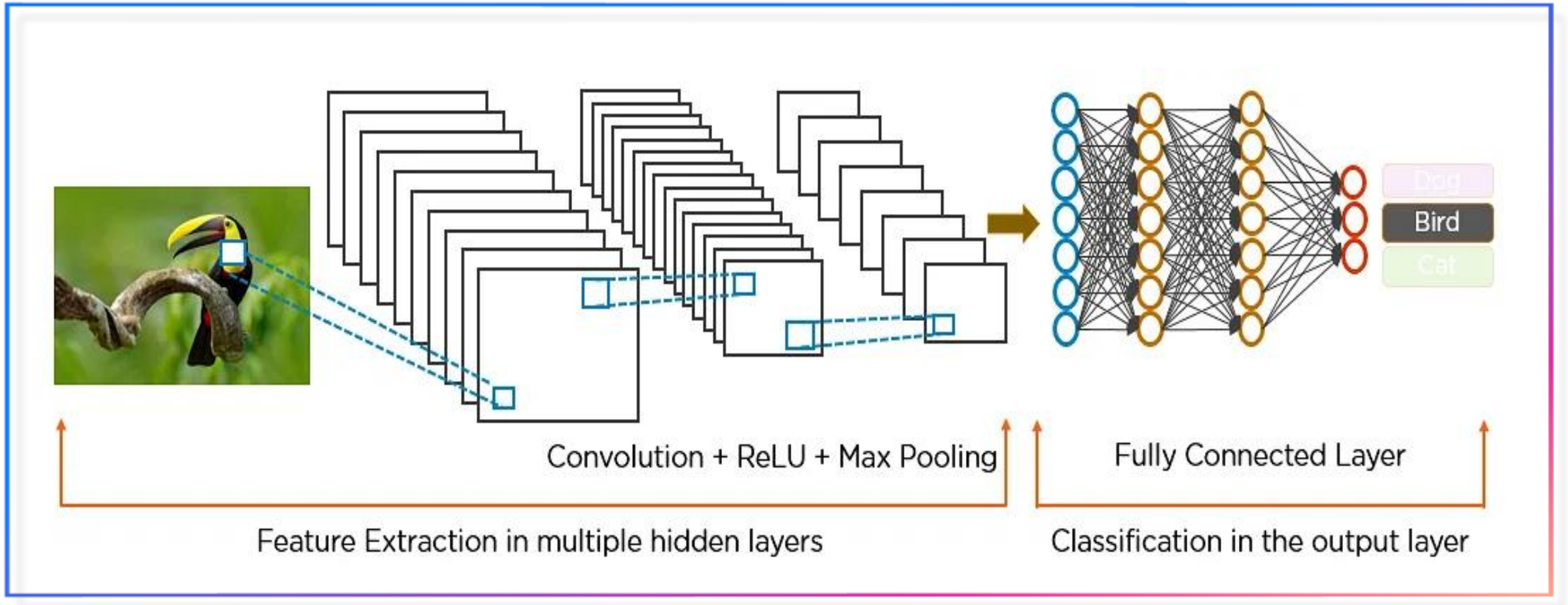
How Machine Intelligence Works?



Simply Matrix Multiplications with additives ☺

≥ 1 “hidden layers” between inputs & output

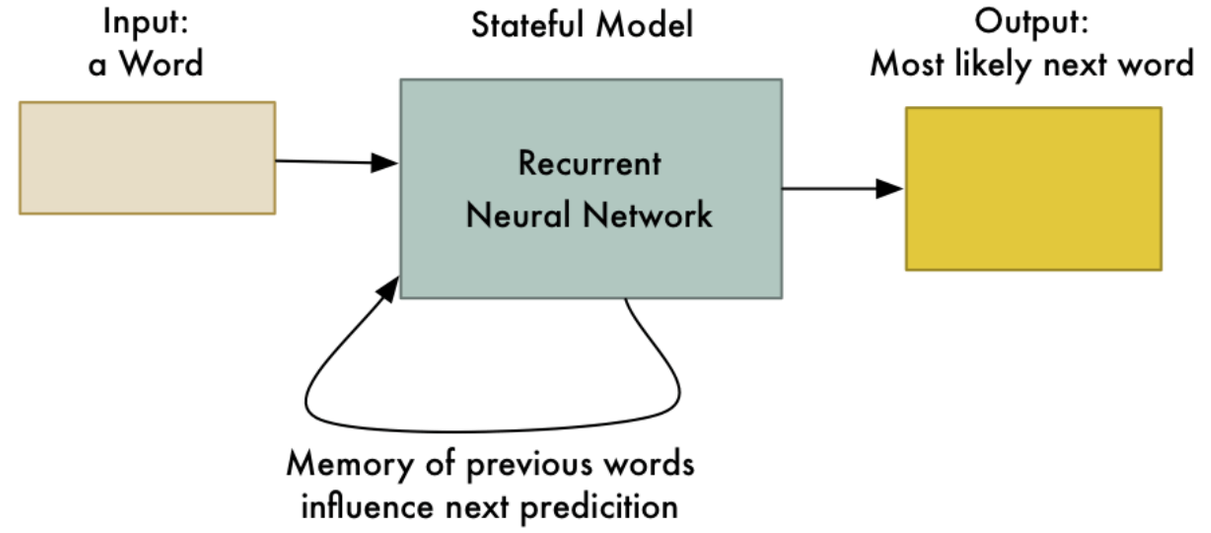
We see and identify Images!



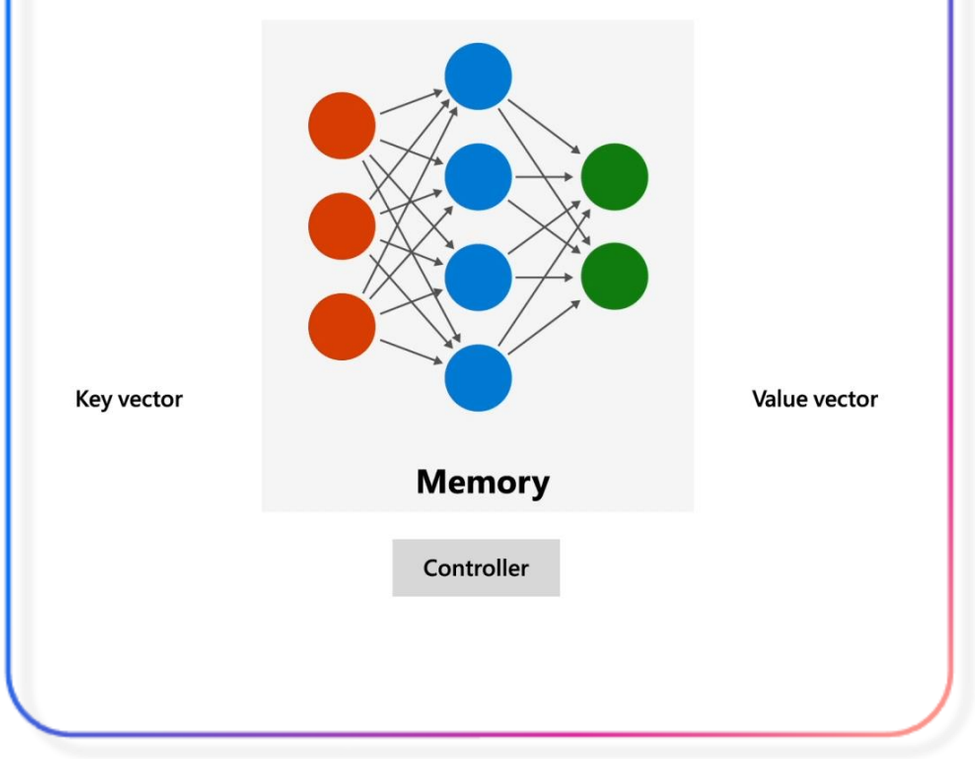
Convolution layers are the machine eyes!

Good for 2D image processing: classification, object recognition, etc.

We have memory?



gif from Adam Geitgey



Recurrent layers are the machine memory & conscious!

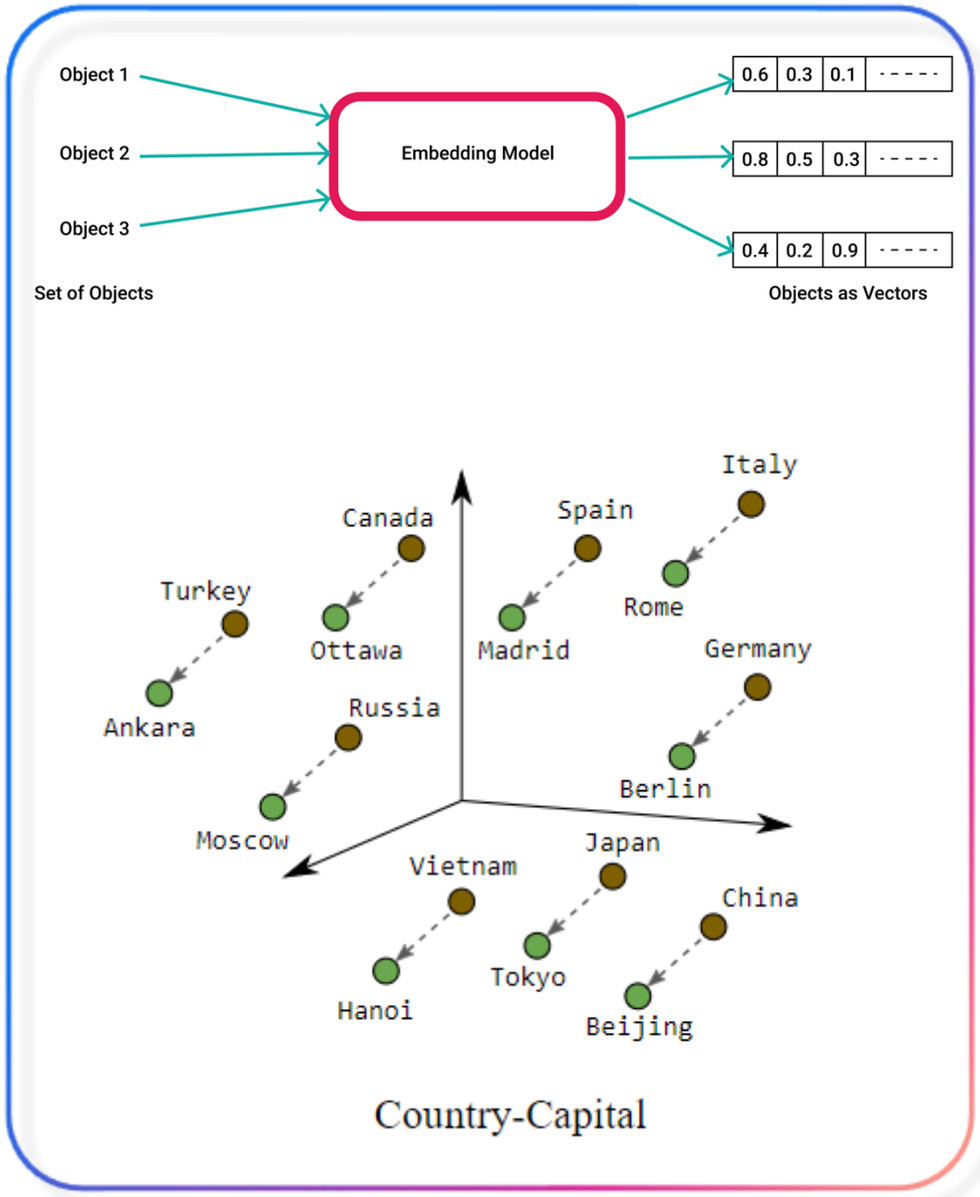
Good for learning over sequences of data, e.g., a sentence of words

Vector Representation

Words as Numbers: Instead of treating words as just text, they are converted into numbers (vectors)

Positioning in Space: Each word is placed in a multi-dimensional space (think of a graph)

Closer Together: Words with similar meanings are placed closer.

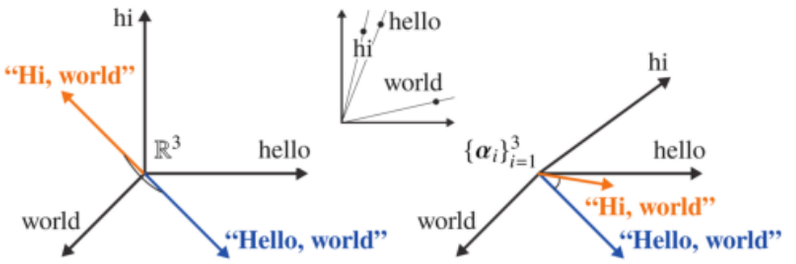


Closer? How to Tell!

Cosine Similarity

- A way to measure how similar two items are by comparing the angle between their vectors.

- Cosine of the Angle:
  - 1: Items are very similar (angle close to 0°).
  - 0: Items are completely different (angle is 90°).

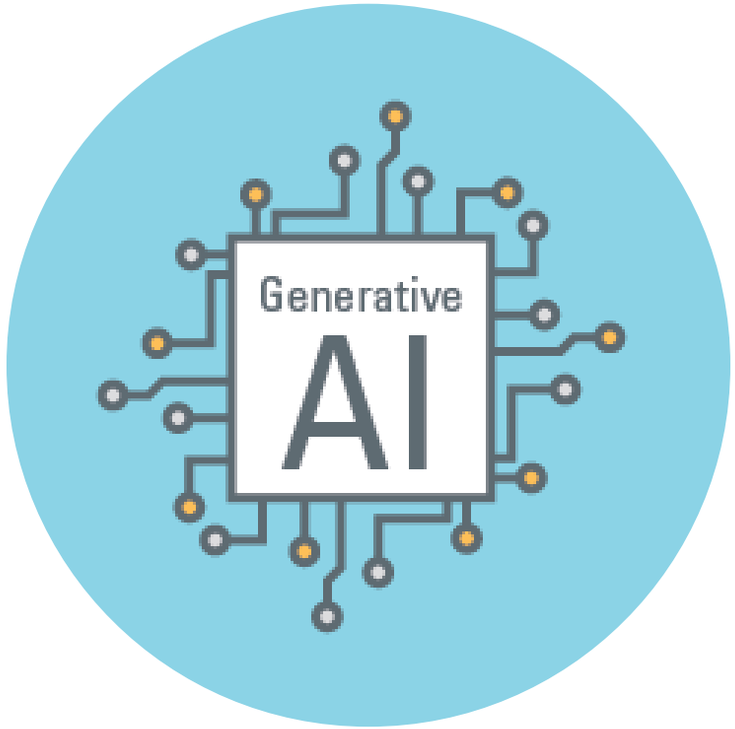


$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Generative AI

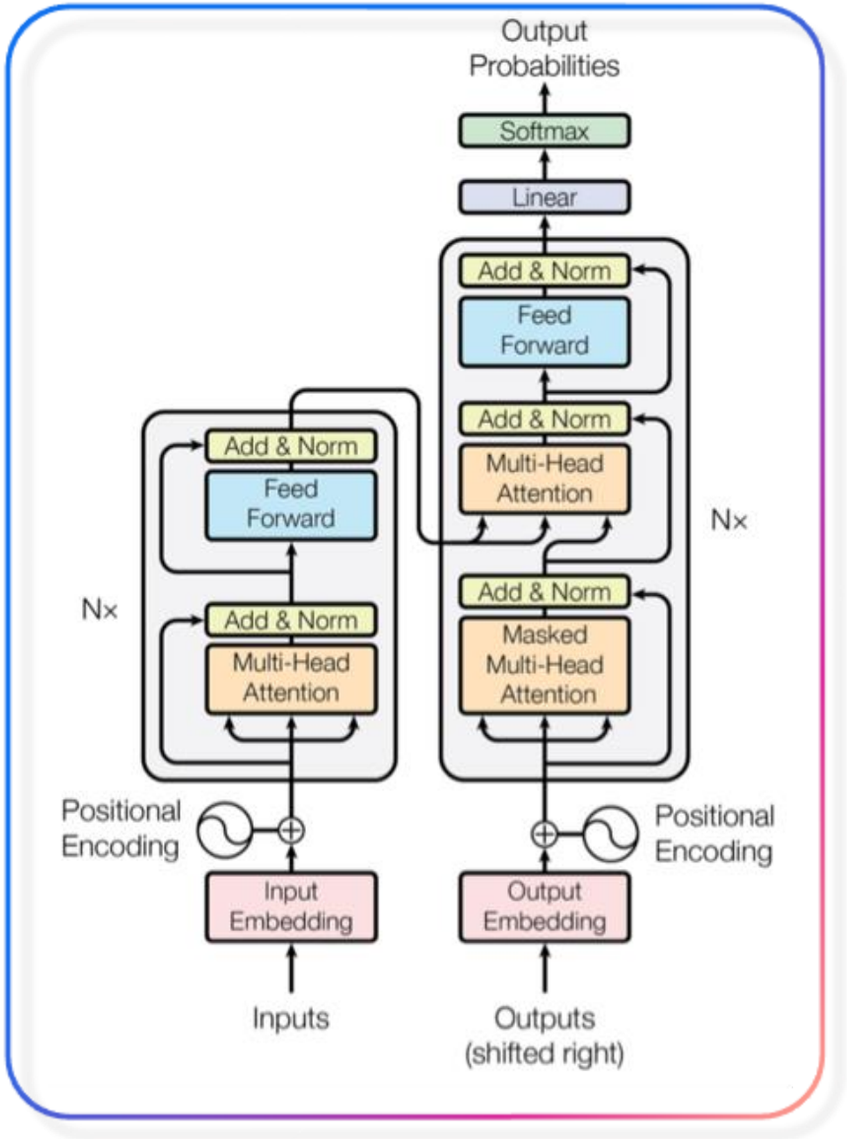
What is Gen AI?

- Designed to create new content
- Introduced in 2014 by Ian Goodfellow
- Used to create: Text, Image, Audio, Video
- Used in different Industry:
    - **Content Creation**: Personalized marketing, article writing.
    - **Healthcare**: Drug discovery, medical imaging analysis.
    - **Entertainment**: Game development, movie special effects.



Language Models (Transformers)

- Built using deep learning architectures, a.k.a. **transformers**.
- Billions of **parameters** to capture complex language patterns.
- Trained on **general data** and later fine-tuned for specific tasks.



Language Models – Market Availability

Large Language Models (LLMs)

Azure/OpenAI

- GPT-4-Turbo
- GPT-4o/mini
- GPT-4V
- Text-embedding-ada-002
- GPT-3.5-Turbo

Meta

- Llama-3
- Llama-2
- CodeLlama

Hugging Face

- Falcon/TII
- Stable Diffusion/ Stability AI
- Dolly/Databricks
- CLIP/OpenAI

Google

- 1.5 Flash
- 1.5 Flash Pro
- AQA

ANTHROP\C

- Sonnet
- Opus
- Haiku

Small Language Models (SLMs)

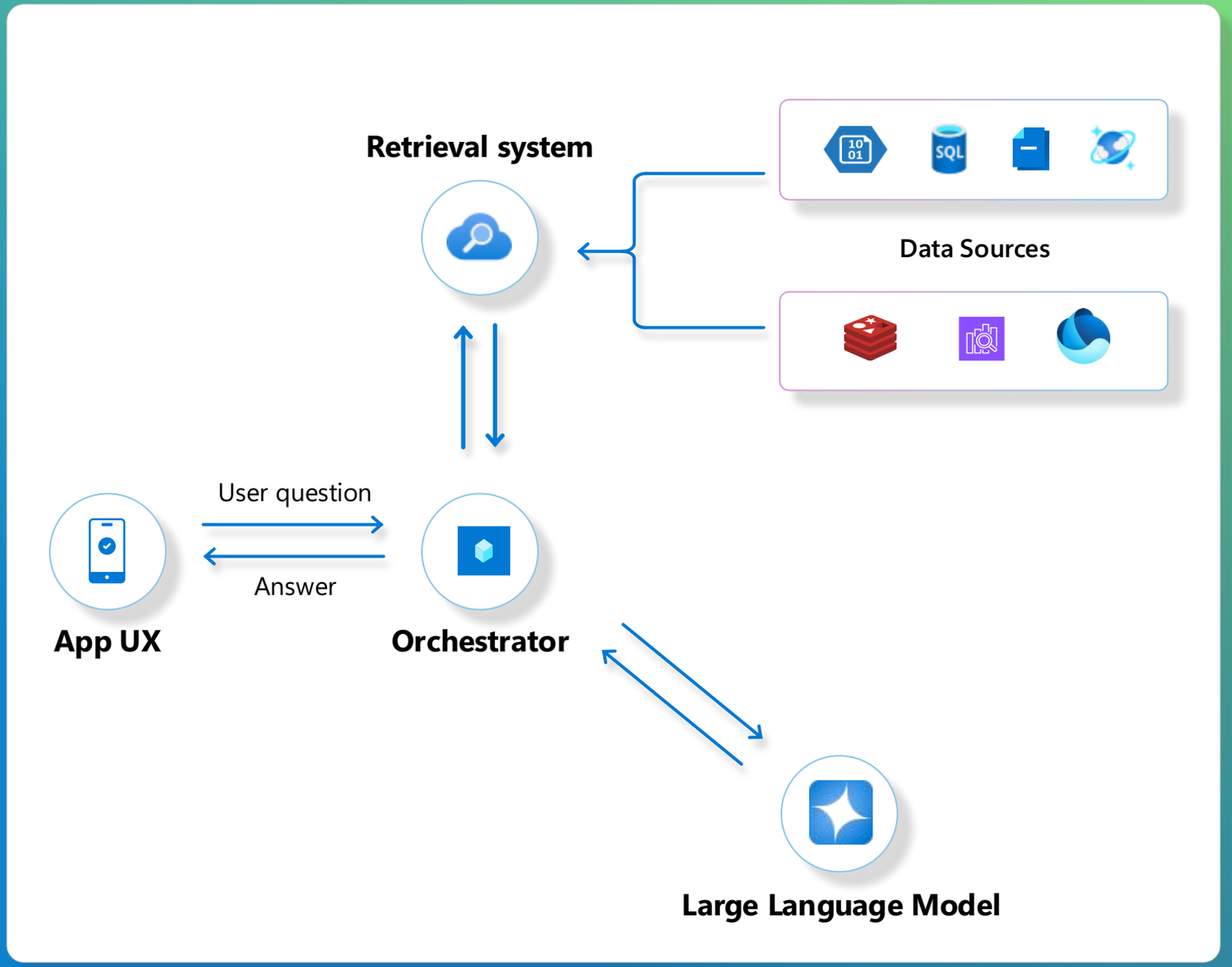
Microsoft Phi

- Phi-1
- Phi-2
- Phi-3

Microsoft Orca

- Orca 1
- Orca 2

Retrieval Augmented Generation



Introduction to Python

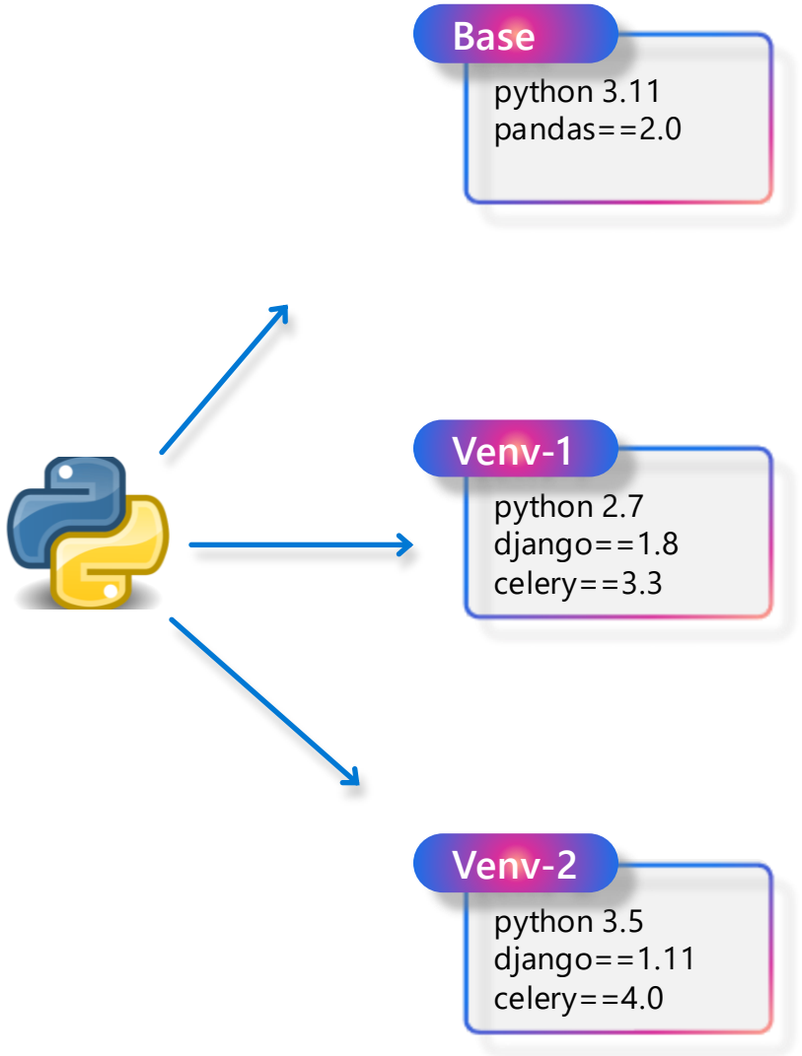
Python

- Popular high-level programming language used in various applications
- Easy language to learn because of its simple syntax
- Case sensitive and indentation/space sensitive
- Can be used for simple tasks such as plotting or for more complex tasks like machine learning
- Types of operations: Interactive and files
- Extension (.py, ipynb)

Environment Setup

Virtual Environment

- Environment Isolation
- Dependency conflicts Free
- Reproducible



Virtual Environment Managers

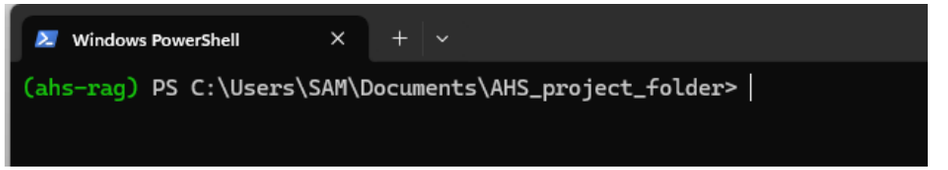
- **Anaconda**:
  - Full platform with environment and package management via conda.
  - Ideal for data science and machine learning.

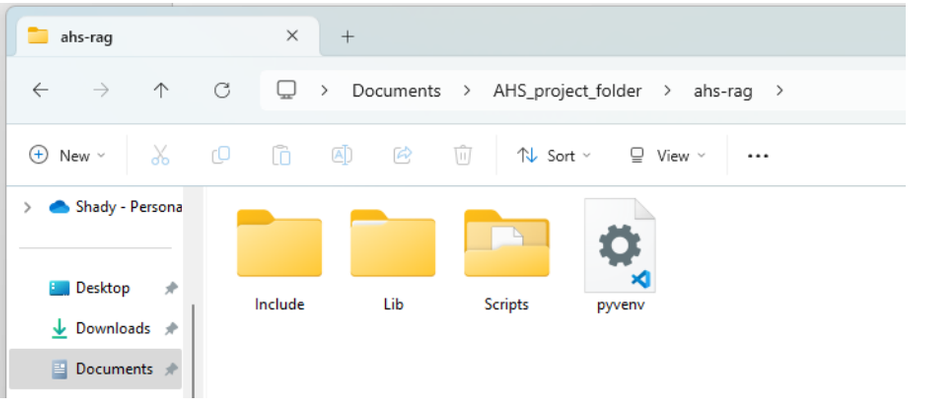
- **venv**:
  - Lightweight built-in tool for creating isolated environments.

- **Poetry**:
  - Modern manager for dependencies and project packaging.
  - Streamlines Python package creation and publishing.

Virtual Environment Creation - venv

- Setup:
  - Installed with python in windows
- Creating venv “ahs”:
  - Create a project folder in windows explorer
  - Open PowerShell
  - Navigate to the project folder created in step-1
  - Invoke: python –m venv ahs
  - Invoke: .\ahs-rag\Scripts\activate
- You may not be a script executer:
  - Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser





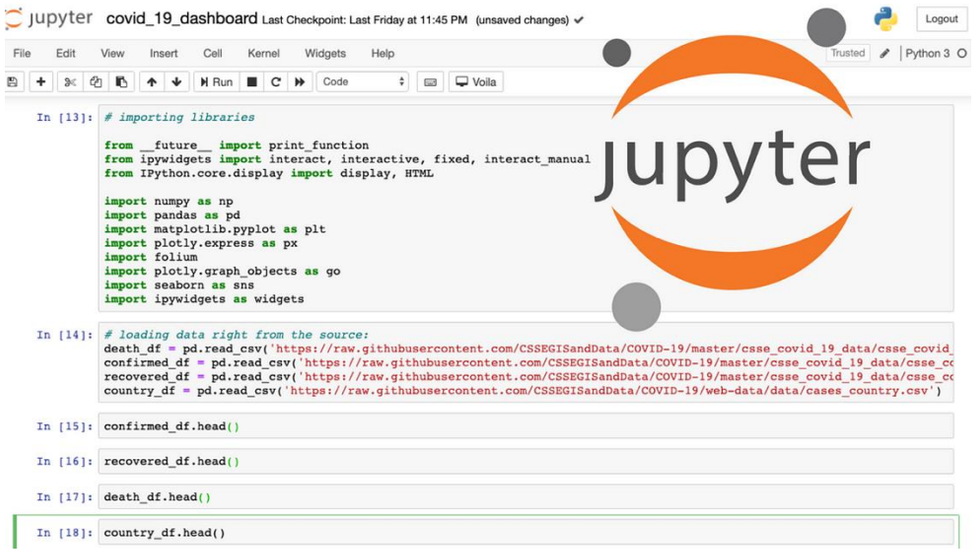
Juypter Notebooks & IDE (Vscode)

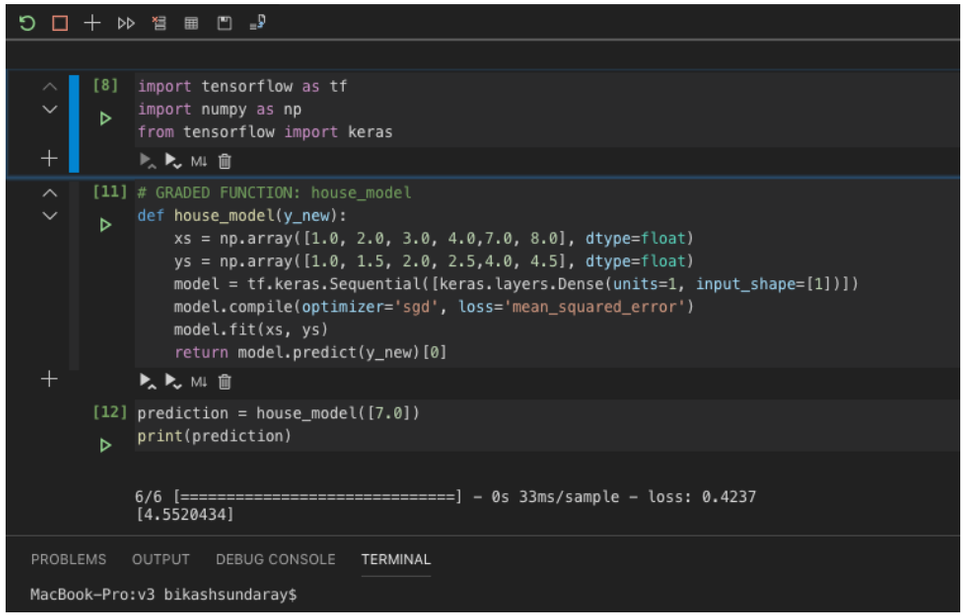
- Jupyter Notebook:
  - Interactive coding with real-time outputs.
  - Ideal for data analysis, visualization, and documentation.
  - Supports multiple languages, primarily Python.

- VS Code:
  - Lightweight code editor with powerful extensions.
  - Supports multiple programming languages and debugging.
  - Seamlessly integrates with Jupyter for interactive notebooks.

- VS Code Extensions:
  - Python: Python - Visual Studio Marketplace
  - Jupyter: Jupyter - Visual Studio Marketplace

- 3rd Party Library:
  - Activate “ahs”
  - pip install ipykernel
  - python -m ipykernel install --user --name=ahs





Python Walkthrough

External Packages

Installing a 3rd party library

- Installation:
  - Using “pip”, pip install <library_name>
- Library in code:
  - Including in code: import <library_name>
  - Giving alias: import <library_name> as <alias>