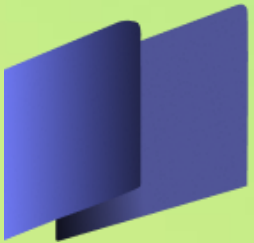


RAG-Based OpenAI Training



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



Session-1: Introduction to Python Developers Guide

Shady A. Mohammed (S.A.M)

Proud Partner



Agenda

⇒ Foundational Concepts

⇒ Overview: Generative AI

⇒ Python As Development Language

⇒ Environment Setup

⇒ External Package (Installation & Usage)

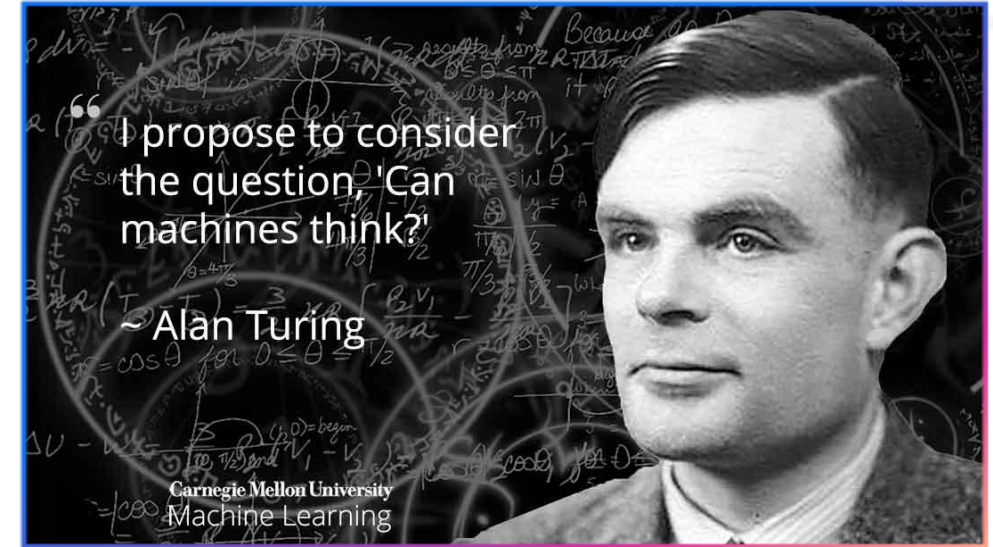


The background features a soft, abstract design. A white, mountain-like peak is centered at the top. A large, smooth arc with a rainbow gradient (from blue to red to yellow) curves from the top right down towards the bottom right corner. The overall color palette is light and airy, with pastel tones.

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

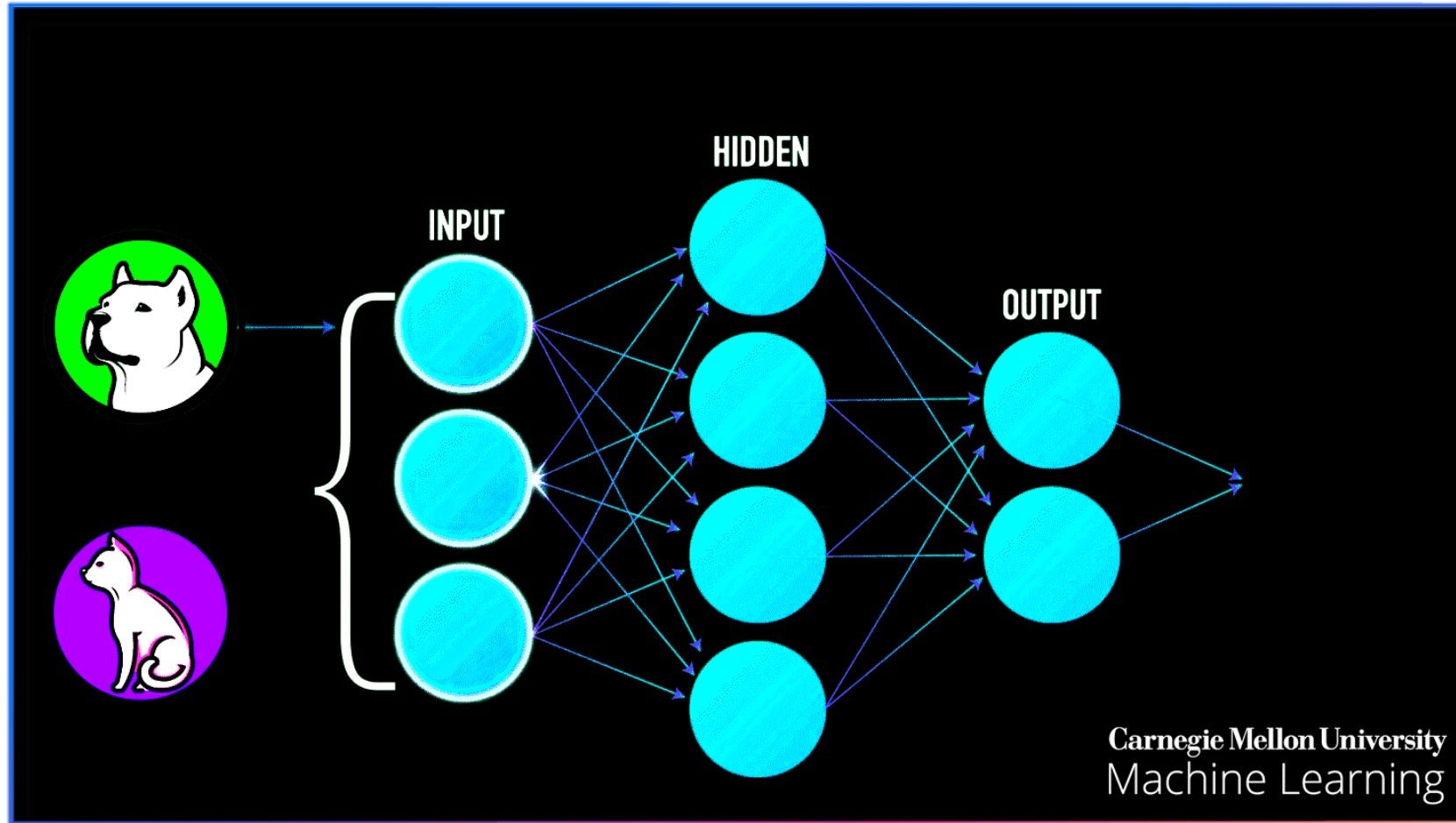
I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words "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?



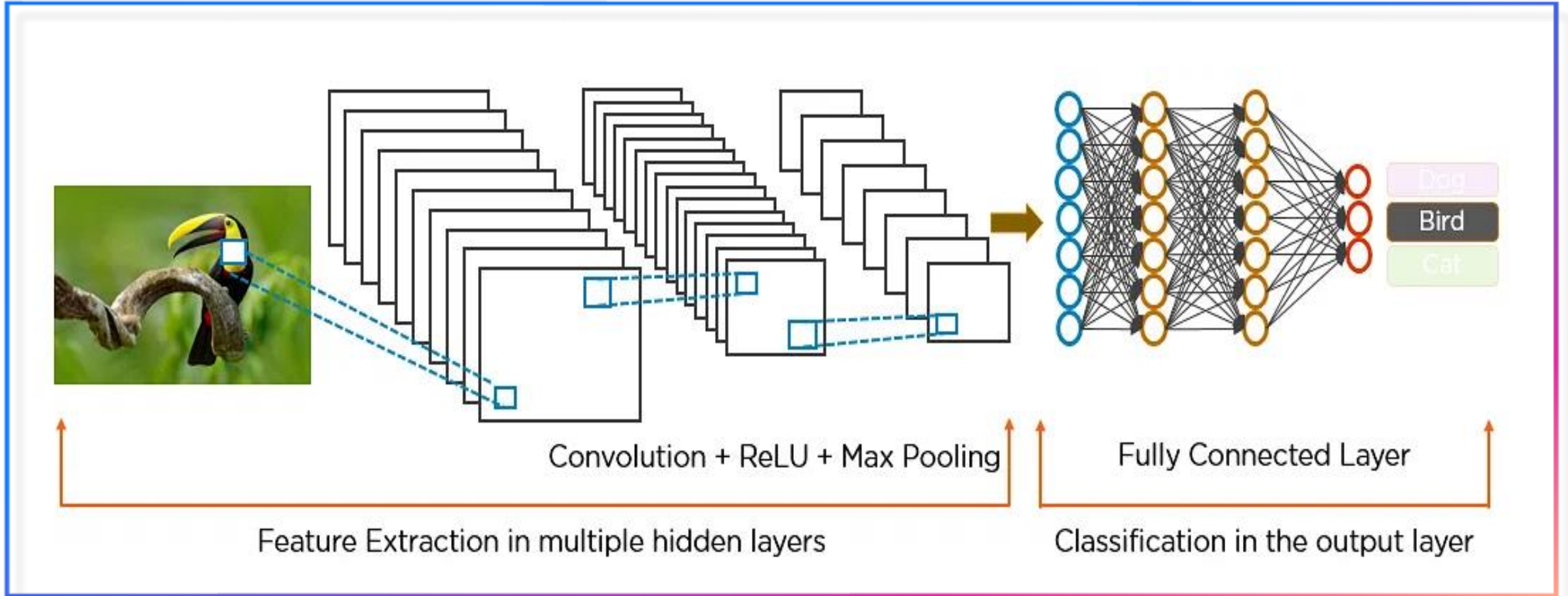
Human Brain has neurons, either fire or rest to exchange Signals!

How Machine Intelligence Works?



Simply Matrix Multiplications with additives 😊
 ≥ 1 "hidden layers" between inputs & output

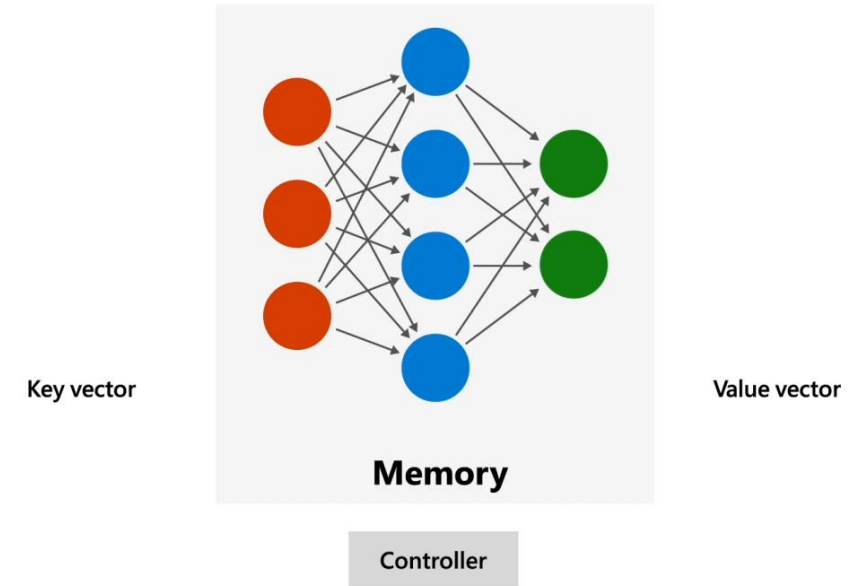
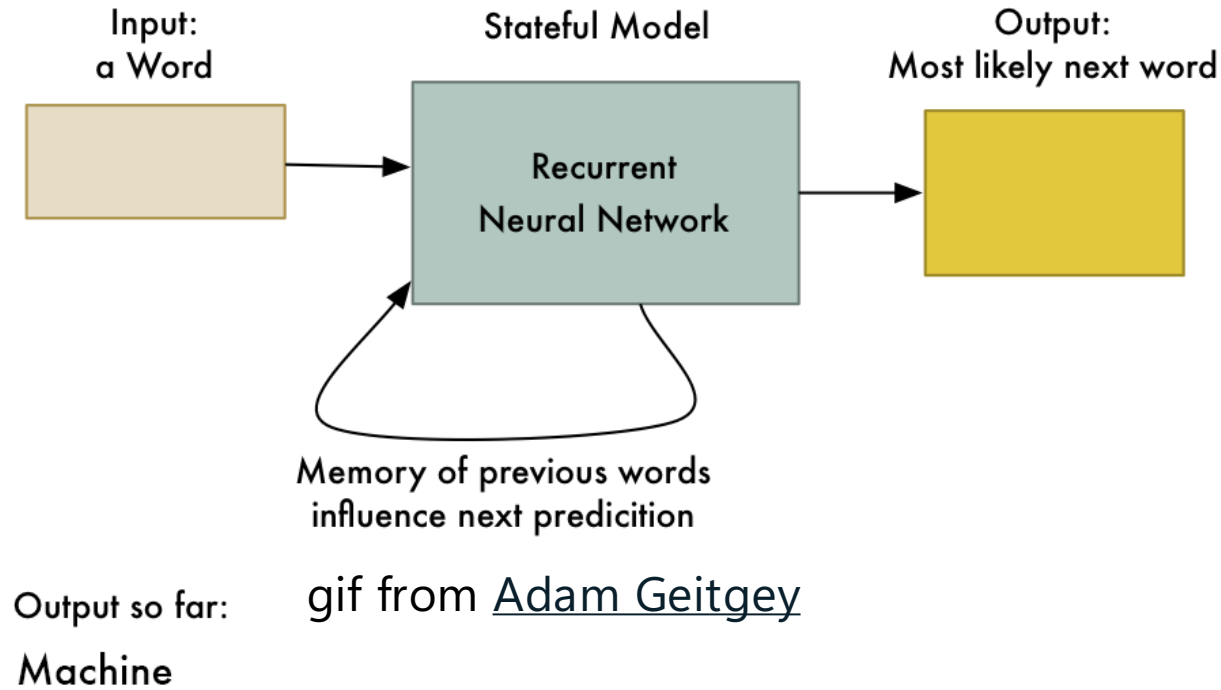
We see and identify Images!



Convolution layers are the machine eyes!

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

We have memory?



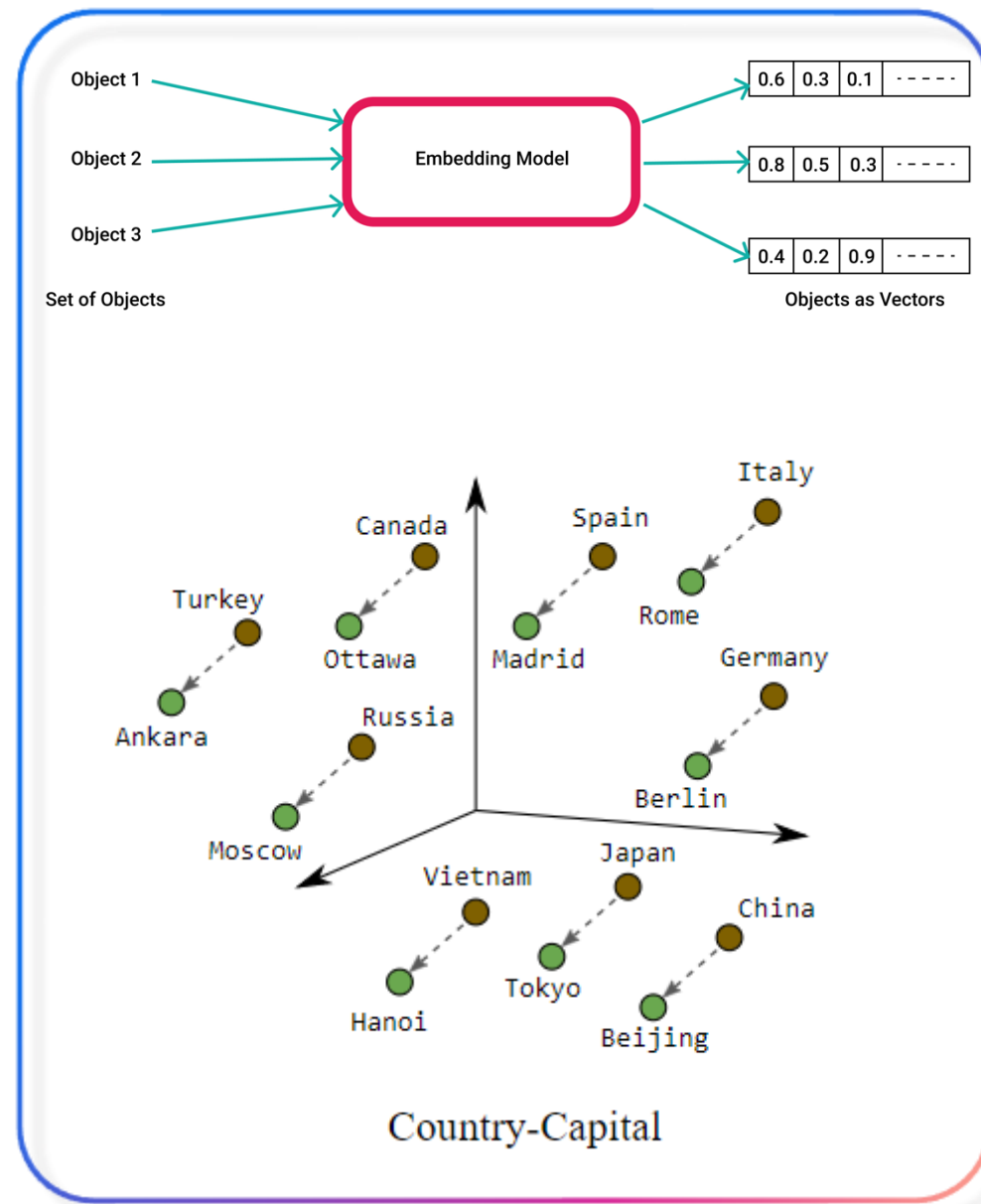
Recurrent layers are the machine memory & conscious!
Good for learning over sequences of data, e.g., a sentence of words

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)

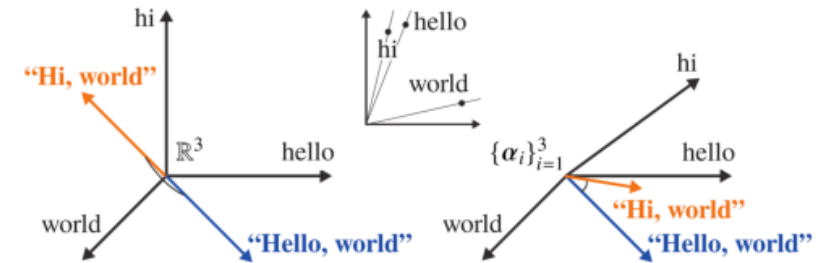
Closer Together: Words with similar meanings are placed closer.



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°).



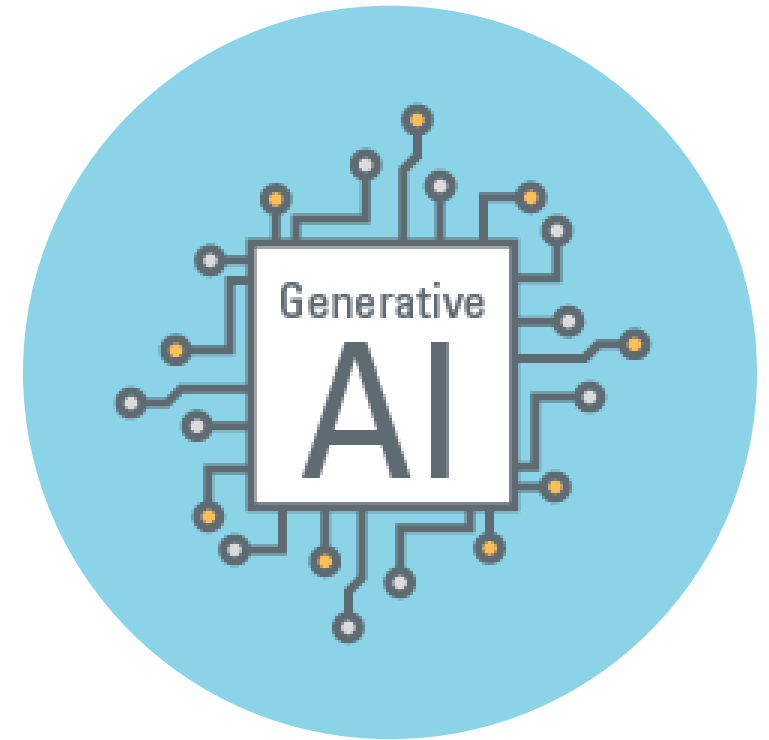
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Generative AI

The background features a large, white, curved shape that resembles a stylized mountain or a wave, set against a light gray background. A vibrant rainbow gradient, transitioning from blue to purple, pink, orange, and yellow, curves along the right side of the white shape.

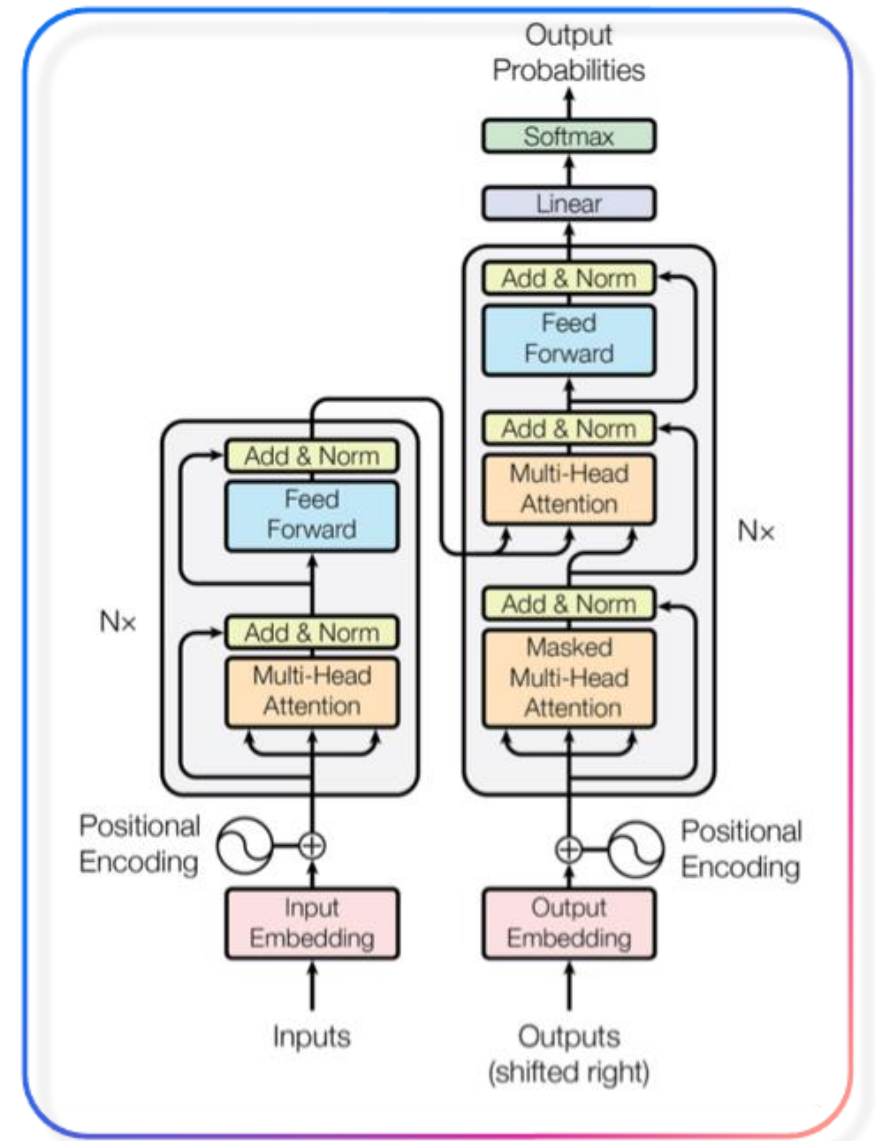
What is Gen AI?

- 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



- 1.5 Flash
- 1.5 Flash Pro
- AQA

ANTHROPIC

- Sonnet
- Opus
- Haiku

Small Language Models (SLMs)



Microsoft Phi

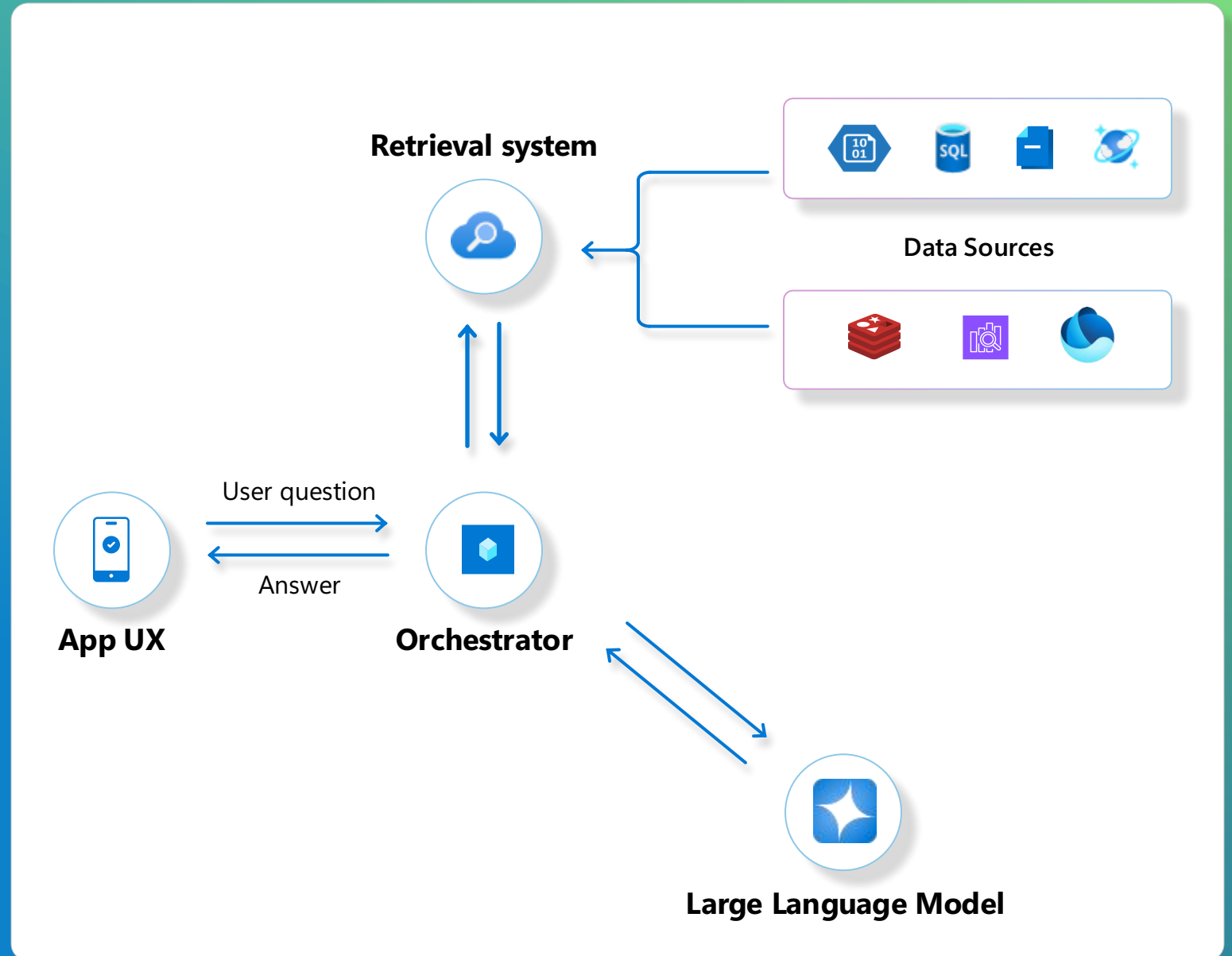
- Phi-1
- Phi-2
- Phi-3



Microsoft Orca

- Orca 1
- Orca 2

Retrieval Augmented Generation



The background features a soft, abstract design. A white, mountain-like peak is centered at the top. Below it, a smooth gradient transitions from light blue on the left, through purple and pink in the center, to bright yellow on the right. A thin, curved line in the rainbow colors (blue, purple, pink, orange, yellow) arches over the text.

Introduction to Python

Python

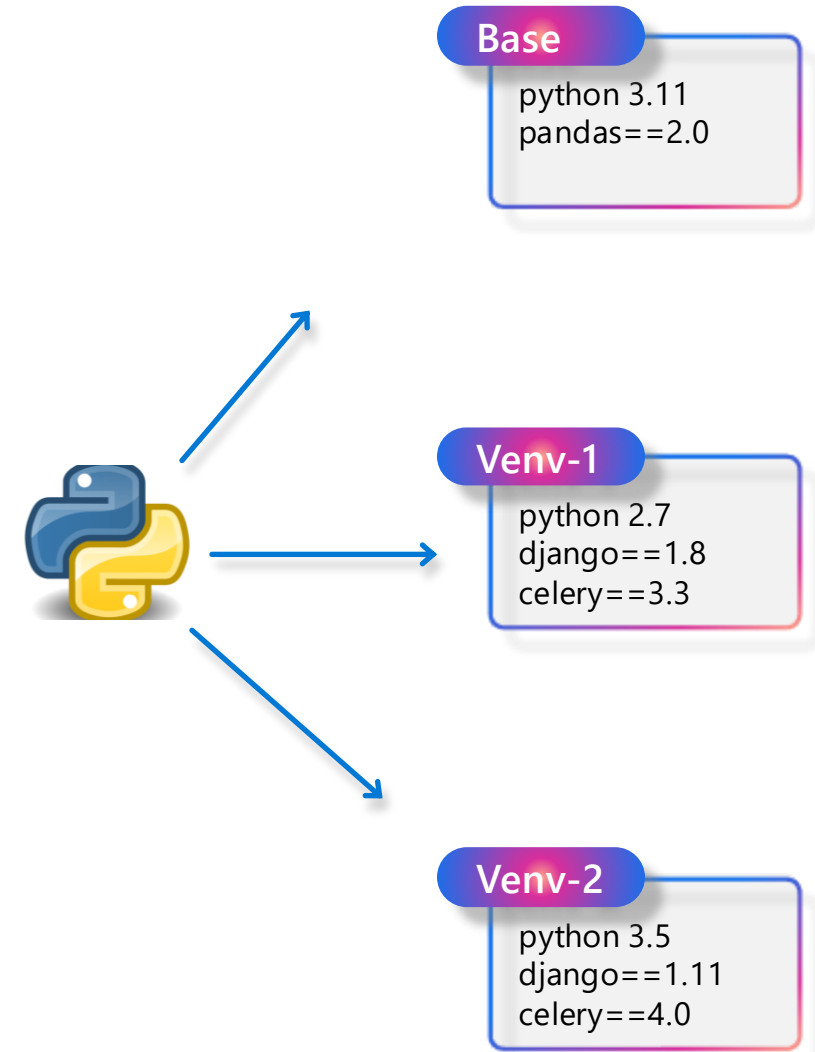
- Popular high-level programming language used in various applications
- Easy language to learn because of its simple syntax
- Case sensitive and indentation/space sensitive
- Can be used for simple tasks such as plotting or for more complex tasks like machine learning
- Types of operations: Interactive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



Environment Setup

Virtual Environment

- Environment Isolation
- Dependency conflicts Free
- Reproducible



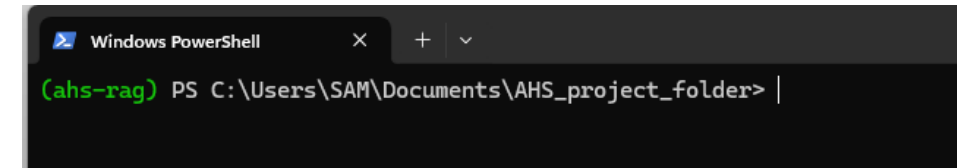
Virtual Environment Managers

- **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.

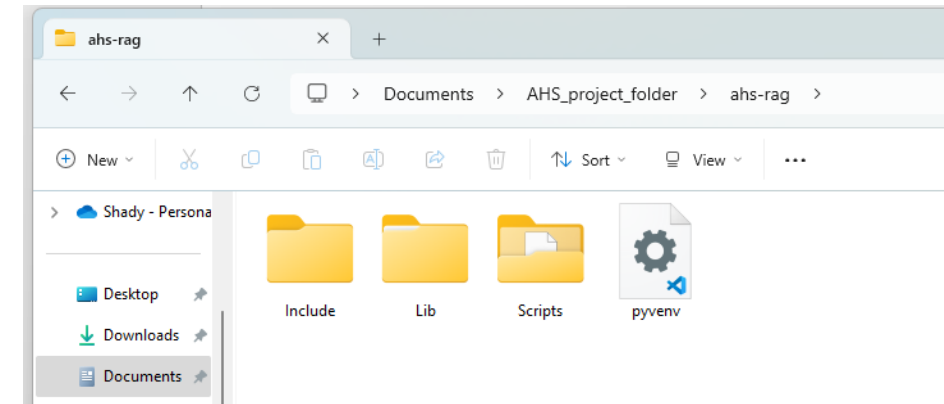


Virtual Environment Creation - venv

- 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`



```
Windows PowerShell
(ahs-rag) PS C:\Users\SAM\Documents\AHS_project_folder> |
```



Jupyter 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:

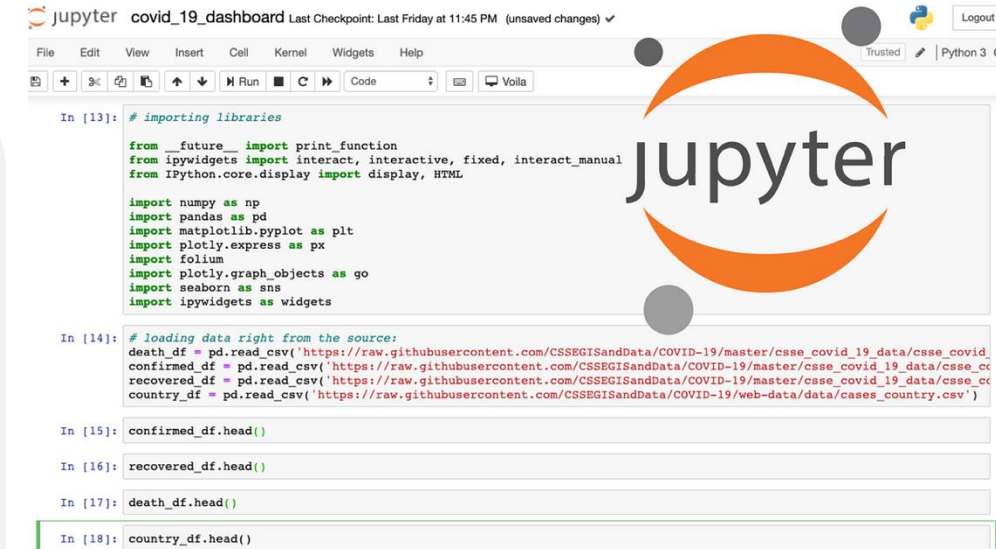
- Lightweight code editor with powerful extensions.
- Supports multiple programming languages and debugging.
- Seamlessly integrates with Jupyter for interactive notebooks.

- 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



```
In [13]: # importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

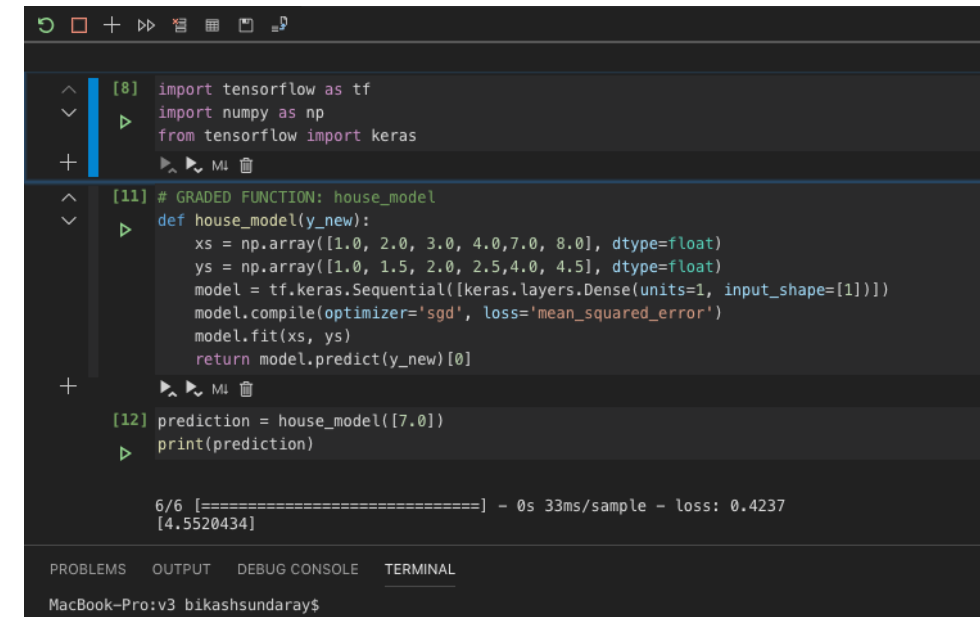
In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/death_df.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/confirmed_df.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/recovered_df.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```



```
[8] import tensorflow as tf
import numpy as np
from tensorflow import keras

[11] # GRADED FUNCTION: house_model
def house_model(y_new):
    xs = np.array([1.0, 2.0, 3.0, 4.0, 7.0, 8.0], dtype=float)
    ys = np.array([1.0, 1.5, 2.0, 2.5, 4.0, 4.5], dtype=float)
    model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
    model.compile(optimizer='sgd', loss='mean_squared_error')
    model.fit(xs, ys)
    return model.predict(y_new)[0]

[12] prediction = house_model([7.0])
print(prediction)

6/6 [=====] - 0s 33ms/sample - loss: 0.4237
[4.5520434]
```


Python Walkthrough

The background is a solid blue color. On the right side, there is a large, abstract shape that is a lighter shade of blue, resembling a stylized wave or a large letter 'C'. A thick orange line curves from the bottom left towards the top right, passing behind the light blue shape. A thin purple line curves from the bottom right towards the top right, also passing behind the light blue shape.

The background features a large, white, curved shape that resembles a stylized mountain or a wave, set against a soft rainbow gradient. The gradient transitions from light blue on the left, through purple and pink, to orange and yellow on the right. The white shape is positioned in the upper half of the frame, with its peak towards the center.

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>