Generative AI in 2 Hours

Arup Nanda

Machine Learning

What is the projected sales, from the data
Should we give the loan, or decline?
Does this look like a bird, or a cat?

Machine Learning → *Predictive*

What is the projected sales, from the data

Should we give the loan, or decline?

Does this look like a bird, or a cat?

Generative AI

- Write an email
- Write a poem
- Generate a picture, video

Generative AI→ *Generate*

Write an email

Write a poem

Generate a picture, video

Language Models

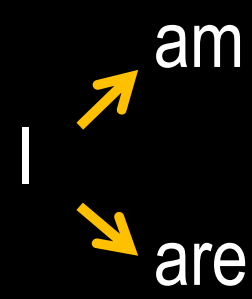
Generative AI→ *Generate*

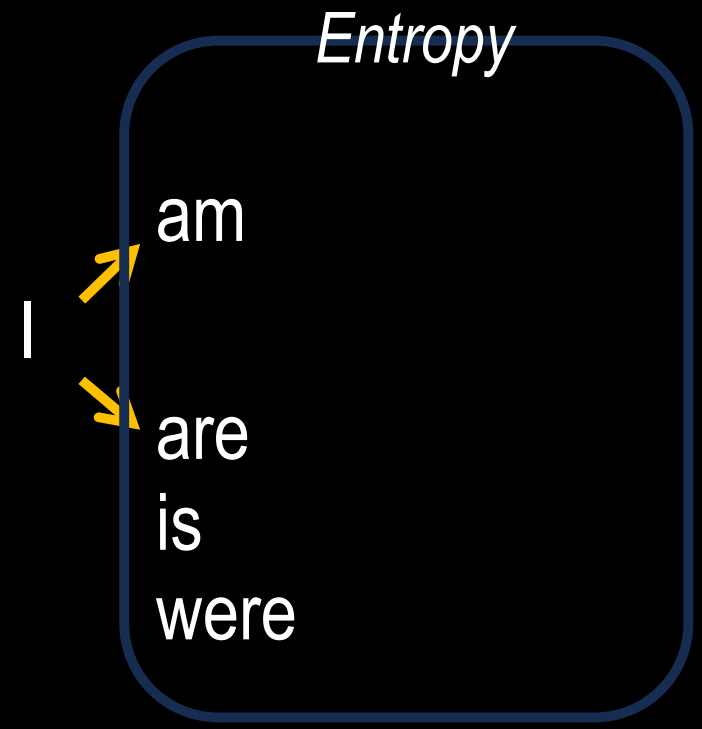
Write an email

Write a poem

Generate a picture, video

What is a Language Model

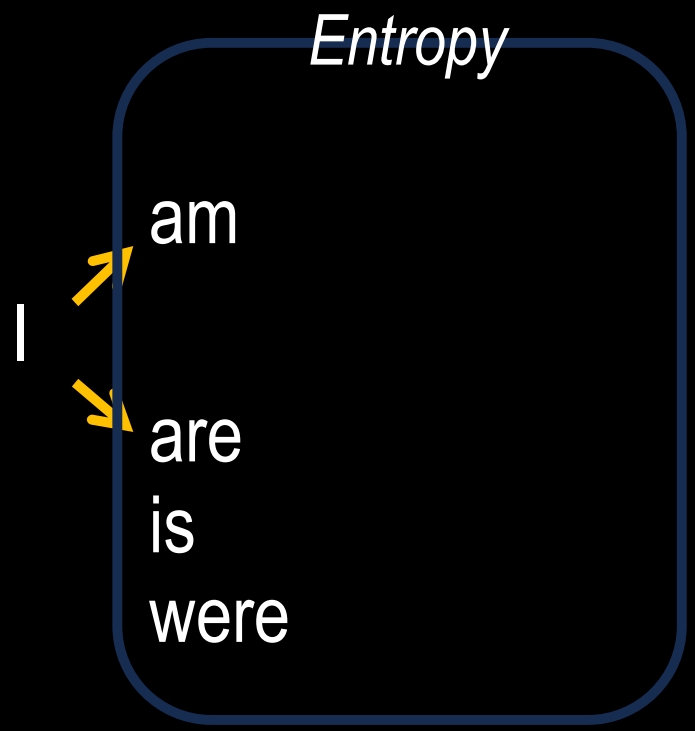

am
I
are


Entropy
I
am
are
is
were

I was going

were

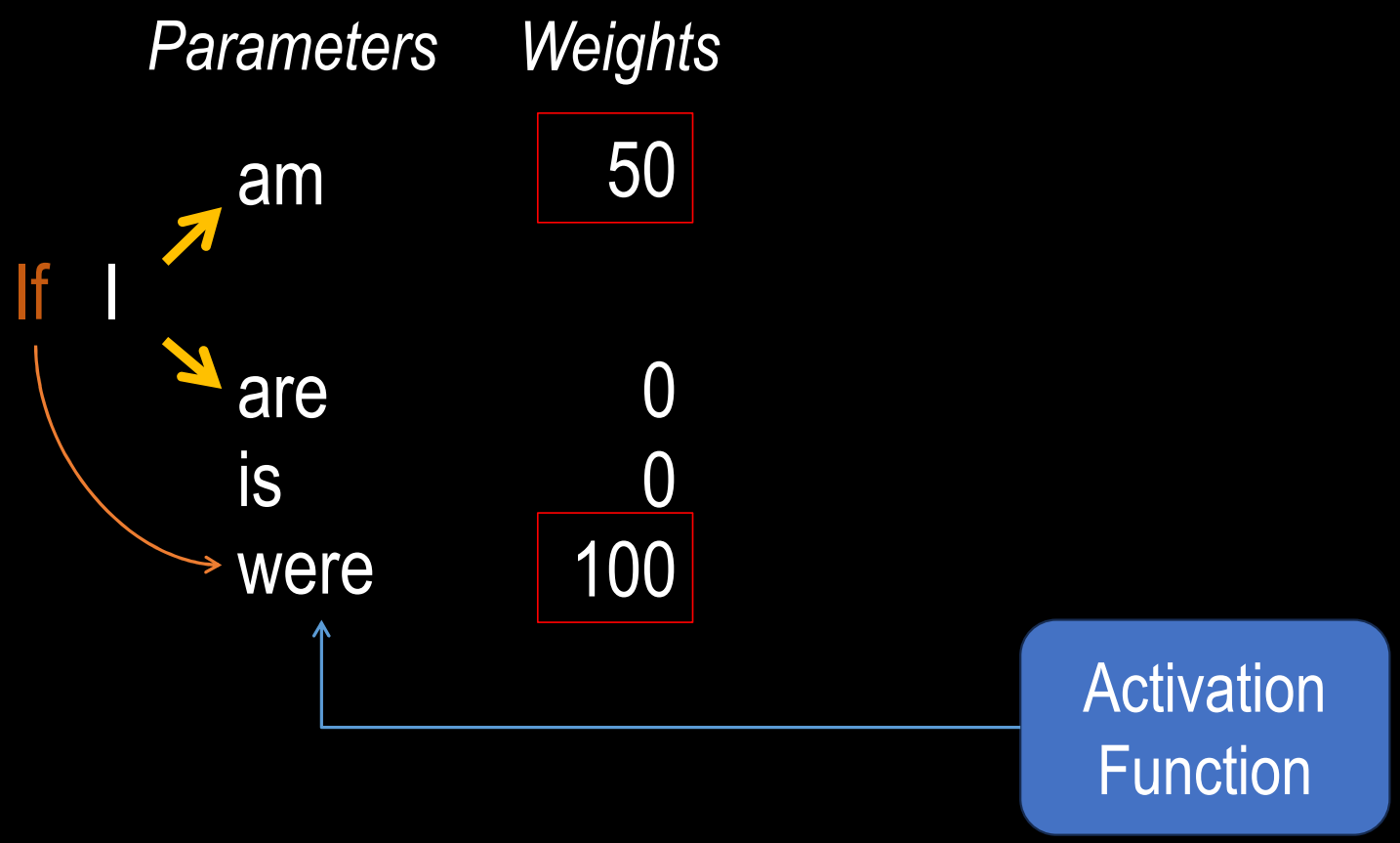

was
If
I
were
going



Source

Textual data, such as Wikipedia, reddit, Quora, Facebook, etc.


Weights
I
am
100
are
0
is
0
were
50


Parameters
Weights
If I
am
50
are
0
is
0
were
100
Activation Function

Generative AI actually does not generate anything.

It merely predicts the next word from a list of words

Encoders

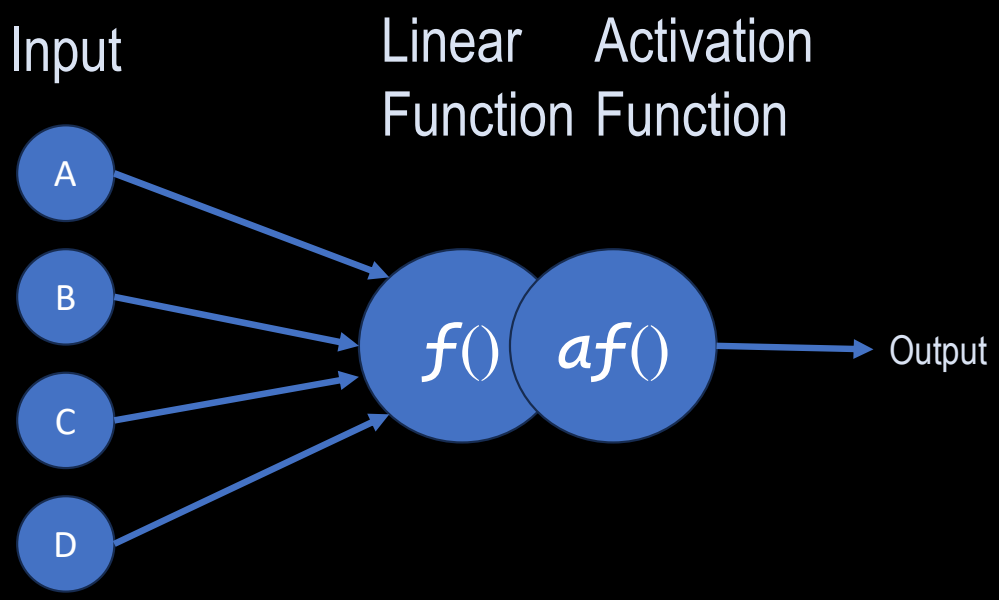
Tokens

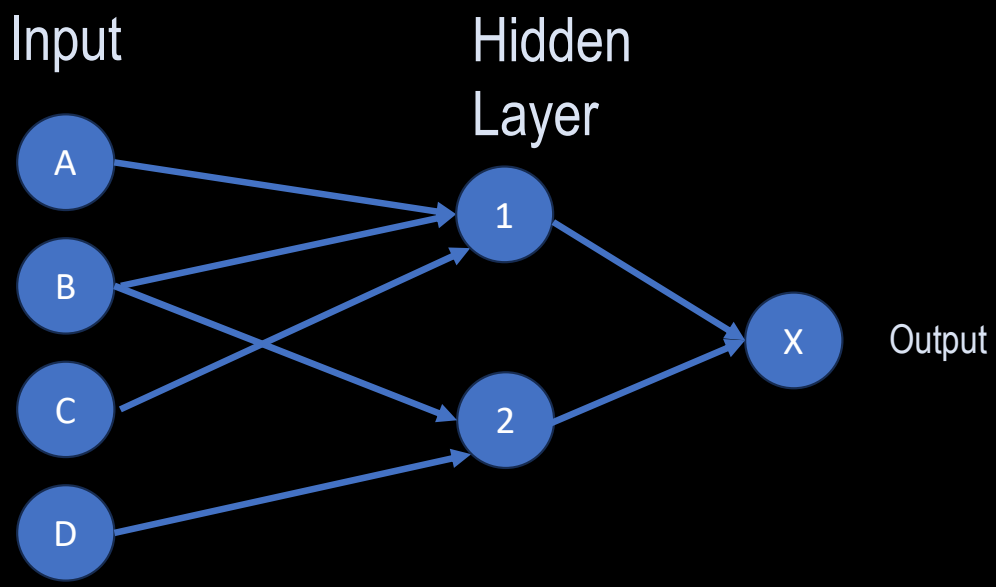
Lyson gave $1M to Arup

Positional Encoding

- Tokenization of words
- Converting to vectors
- Positional Encoding
- Encoders feed forward
- Decoders look at the past output
- Output: based on probability

Transformer Architecture

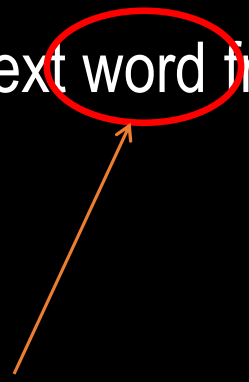

Input
Linear Function
Activation Function
A
B
C
D
f()
af()
Output



Neural Network

Generative AI actually does not generate anything.

It merely predicts the next word from a list of words



This could be a bit of knowledge

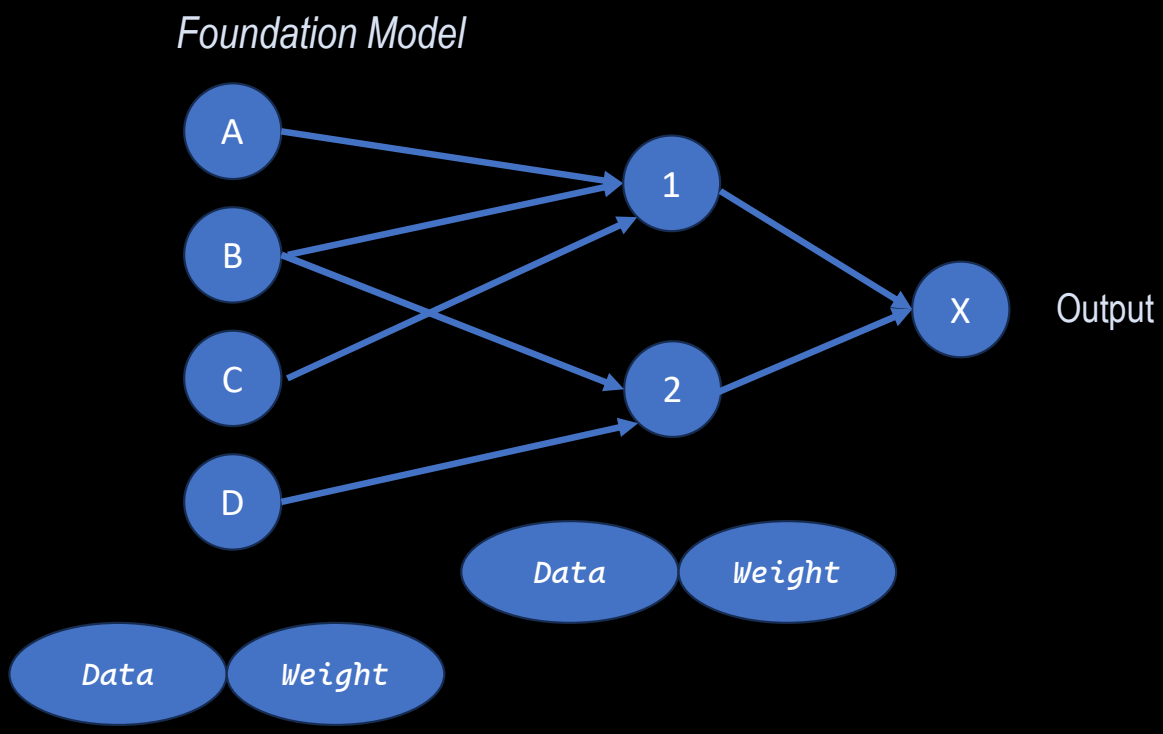
The ticker symbol of Twitter is TWTR

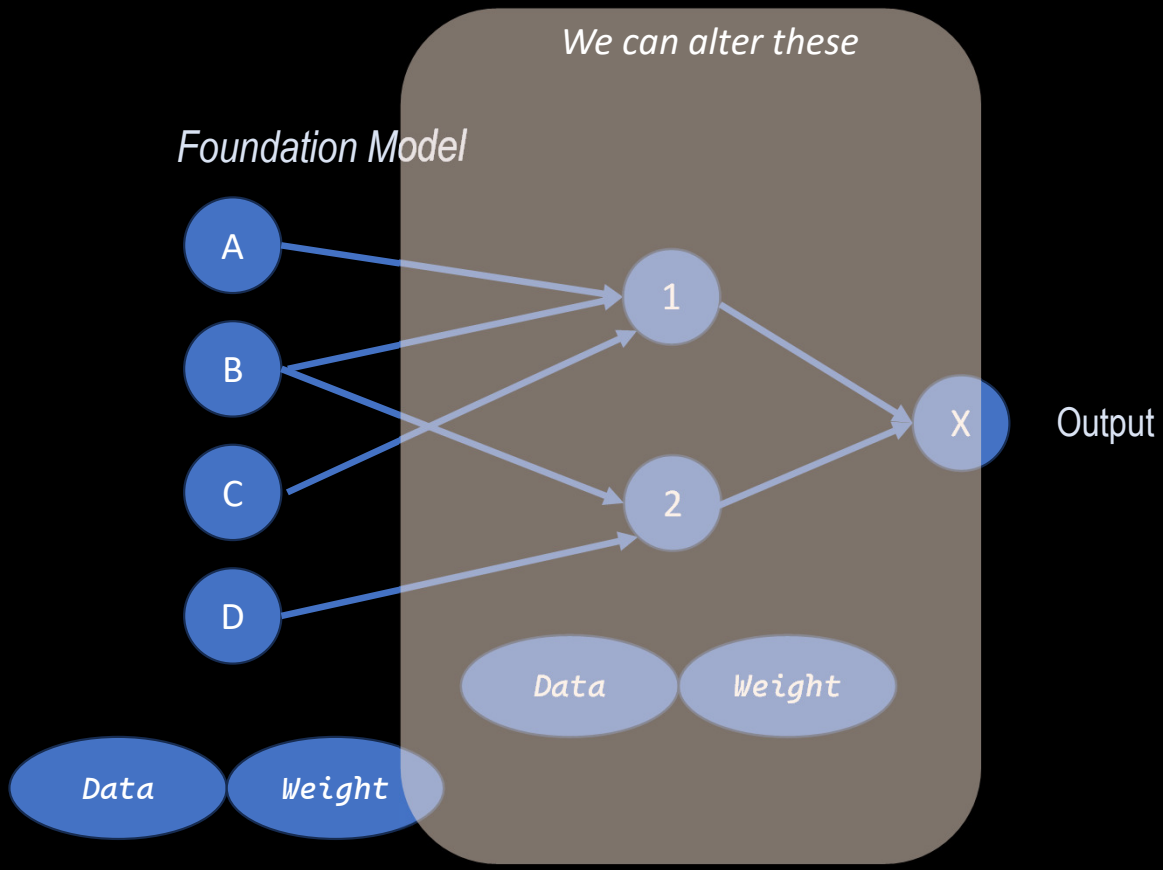
As of the training of this model

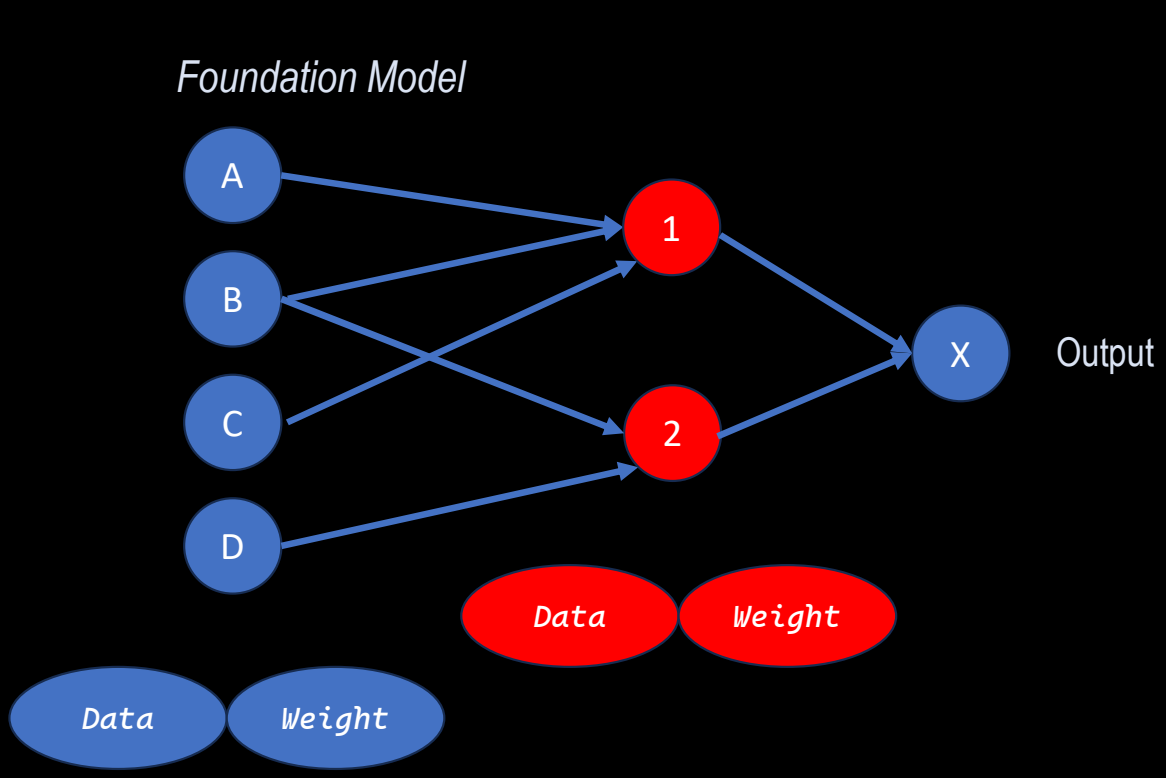
This is wrong now

The ticker symbol of Twitter is TWTR

This is called hallucination

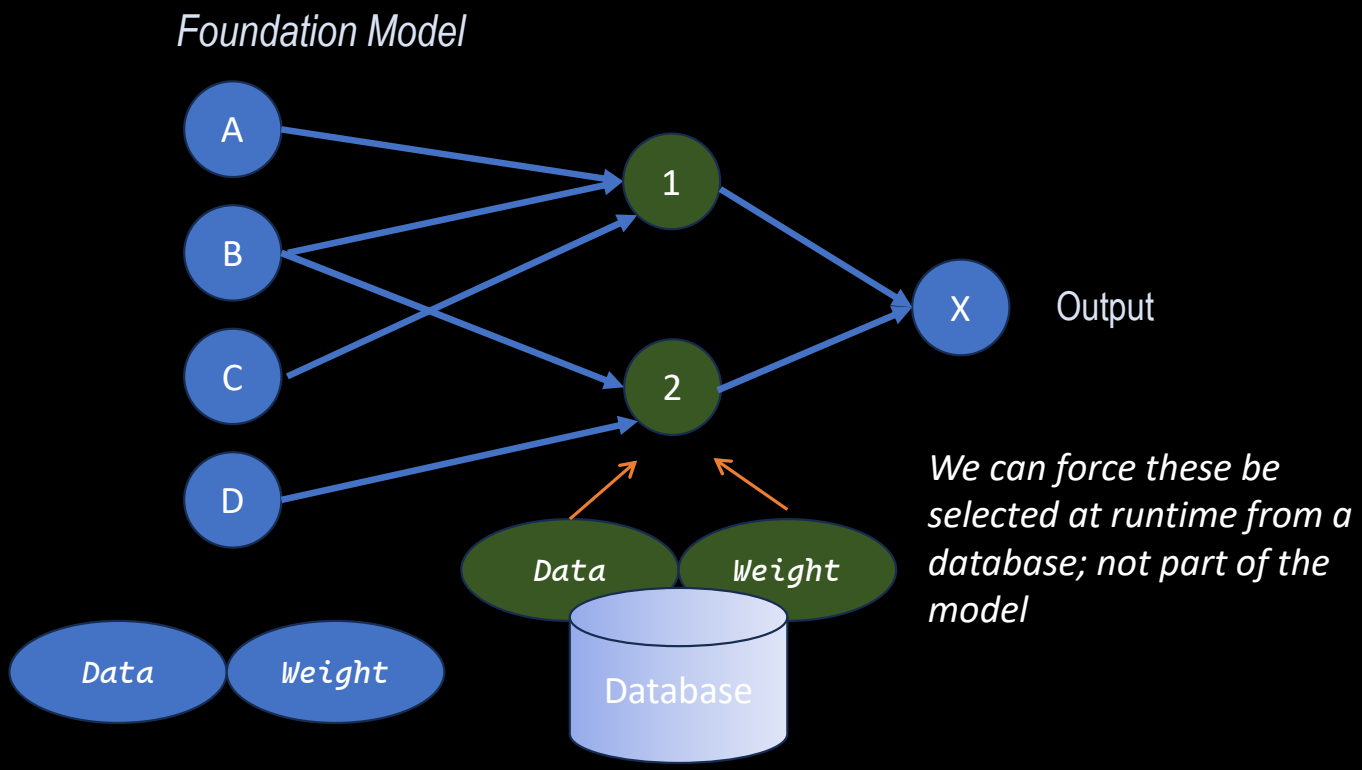

Foundation Model
A
B
C
D
1
2
X
Output
Data
Weight
Data
Weight


We can alter these
Foundation Model
A
B
C
D
1
2
X
Output
Data
Weight
Data
Weight



We can change the parameters and the weights

Fine Tuning



Retrieval Augmented Generation (RAG)

Vector Comparison

github.com/arupnanda/tif-vector-talk

Find me a customer *like* Lisa.

Someone with the same:

Name?
Age?
Networth?

Someone with the same:

Name? `where name = ‘lisa’`
Age? `where age = <lisa’s age>`
Networth?

```
where name = …
where age = …
```

Traditional databases do it very well

Who is *like* Lisa?

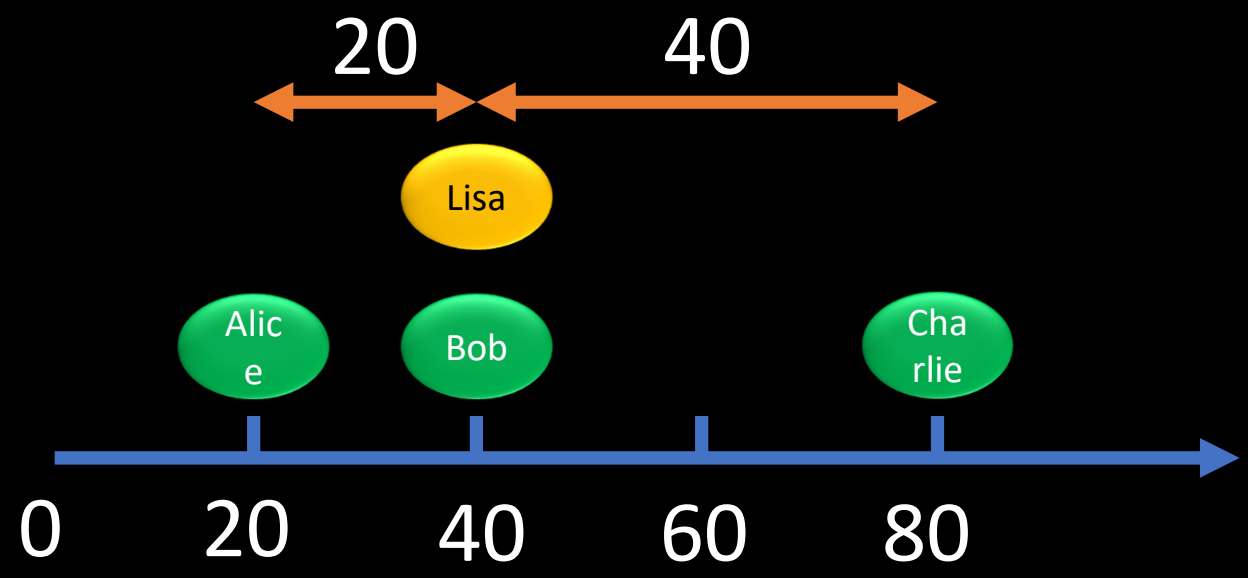
Not exactly the same age; but close?

How close? Within 5%? 10%?

Someone 10% more in age but 5% closer in Networth?

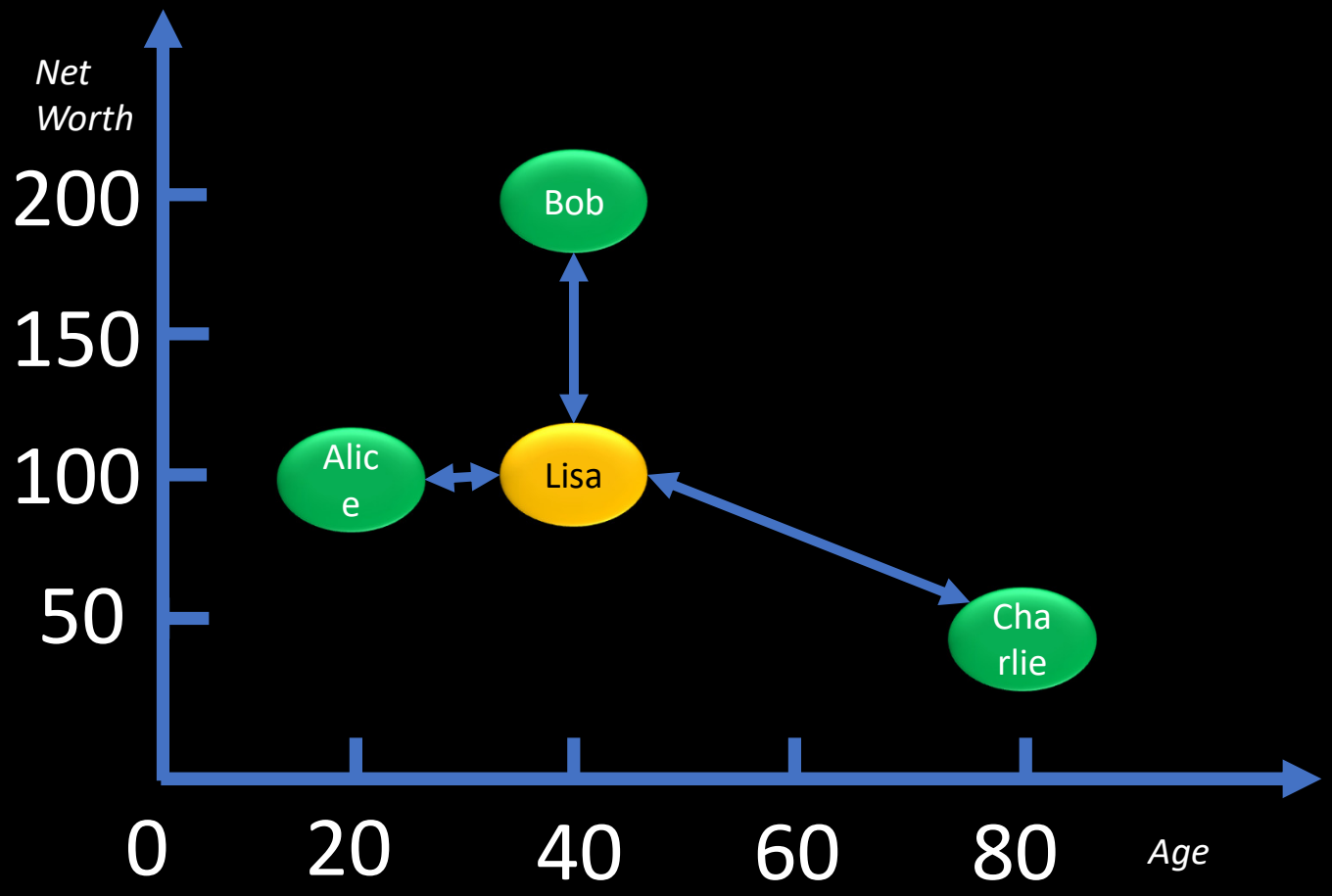
Lisa's Age = 40

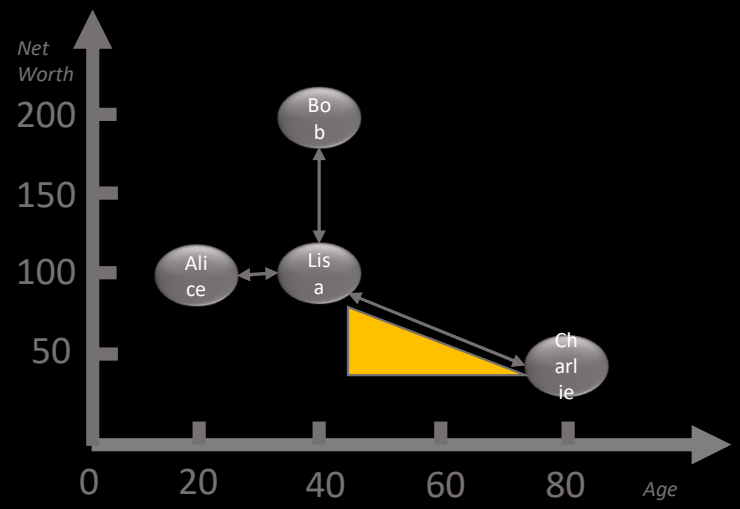
Customer	Age
Alice	20
Bob	40
Charlie	80


20
40
Lisa
Alic
e
Bob
Cha
rlie
0
20
40
60
80

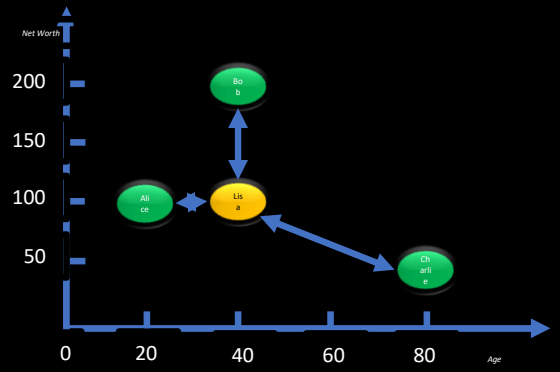
Lisa's Age = 40, Net Worth = 100,000

Customer	Age	Net Worth (in '000s)
Alice	20	150
Bob	40	200
Charlie	80	50


Net Worth
200
150
100
50
Bob
Alic e
Lisa
Cha rlie
0
20
40
60
80
Age



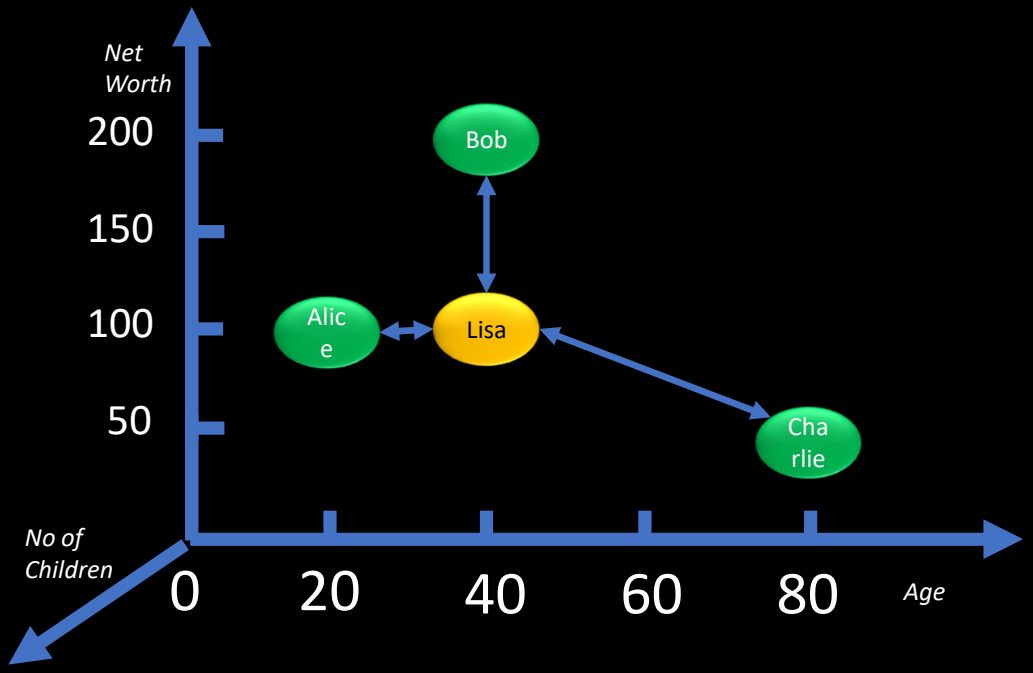
$$Composite\ Distance = \sqrt{Age\ Distance^2 + Networth\ Distance^2}$$



Customer	Age	Age Distance from Lisa	Net Worth (in '000s)	Net Worth Distance from Lisa	Composite Distance from Lisa
Alice	20	0.33	100	0.0	0.33
Bob	40	0.0	200	0.67	0.67
Charlie	80	0.67	50	0.33	0.75

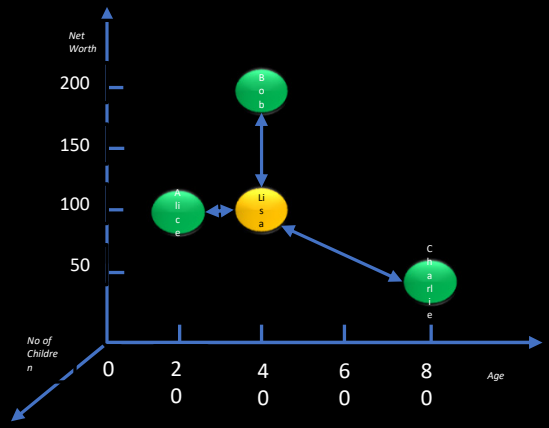
Adding additional comparison dimensions

Number of Children
ZipCode
Favorite color


Net Worth
200
150
100
50
Bob
Alic e
Lisa
Cha rlie
No of Children
0
20
40
60
80
Age

[Age, Networth, No of Children, Zipcode, Fav Color, …]

[0.0001, -2.0003, 1.2134, -5.0001, 0.1112, …]



Step 1

Create and store all dimensions of data in the dataset as vectors

Step 2

Create a vector from a data you want to search for

Step 3

Find the distance between the searched value vector and the stored vectors
The more the distance, the less similar they are

Create and store all dimensions of datapoints in the dataset as vectors

Vector Database

Create a vector from a datapoint you want to search for

Find the distance between the searched value vector and the stored vectors
The more the distance, the less similar they are

Create and store all dimensions of datapoints in the dataset as vectors

Vector Database

Create a vector from a datapoint you want to search for

Vector Library

Find the distance between the searched value vector and the stored vectors
The more the distance, the less similar they are

Create and store all dimensions of datapoints in the dataset as vectors

Vector Database

Create a vector from a datapoint you want to search for

Vector Library

Find the distance between the searched value vector and the stored vectors
The more the distance, the less similar they are

Vector Search

Vector >

```
[Age, Networth, No of Children, Zipcode, Fav Color, …]
```

How do you know how many dimensions need to be vectorized?

And how do you vectorize them?

How do you know how many dimensions need to be vectorized?

And how do you vectorize them?

You can pass them as parameters to a vector library

You can ask the library to generate the vectors

```
import chromadb
client = chromadb.Client()
coll = client.create_collection(name='my_collection’)
coll.add (
    embeddings=[[20,100000], [40,200000], [80,50000]],
    documents=["Alice","Bob","Charlie"],
    ids=["1","2","3"]
)
```

How do you know how many dimensions need to be vectorized?

And how do you vectorize them?

Specialized Machine Learning Models

Example → `sentence_transformers`

Professions

Customer	Profession
Alice	Engineer
Bob	Accountant
Charlie	Artist

Lisa is a “Painter”

Creating a Vector Collection

Customer	Profession
Alice	Engineer
Bob	Accountant
Charlie	Artist

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
alice_vector=model.encode('Engineer').tolist()
bob_vector=model.encode('Accountant').tolist()
charlie_vector=model.encode('Artist').tolist()
coll = client.create_collection(name='my_collection')
coll.add (
    embeddings=[alice_vector,bob_vector,charlie_vector],
    documents=["Alice","Bob","Charlie"],
    ids=["1","2","3"]
)
```

Performing a Vector Search

Customer	Profession
Alice	Engineer
Bob	Accountant
Charlie	Artist

```
coll.query(model.encode('Painter').tolist())
```

```
{'ids': [['3', '1', '2']],
 'embeddings': None,
 'documents': [['Charlie', 'Alice', 'Bob']],
 'metadatas': [[None, None, None]],
 'distances': [[0.638052821159362 8, 1.2819364070892334, 1.3379256725
```

Asking a Question to Wikipedia

Complete code and detailed article series on medium.com. Search on my name.

“why did Americans fight their own”

```
Distance   ID Question
0.986254  223 what made the civil war different from others
1.121846   91 when was america pioneered
1.144610 2072 what triggered the civil war
1.171306  836 What did native americans  do all day
1.224675 1452 who staRTED WORLD WAR I?
1.249743 1598 what date did the american civil war start
1.272201 1586 what where the most important factors that led to the defeat of
the democrates in 1968?
1.272933  589 what happened in 1877 in us
1.277706  602 when did the civil war start and where
```

Asking a Question to Wikipedia

Complete code and detailed article series on medium.com. Search on my name.

“why did Americans fight their own”

```
Distance   ID Question
0.986254  223 what made the civil war different from others
1.121846   91 when was america pioneered
1.144610 2072 what triggered the civil war
1.171306  836 What did native americans  do all day
1.224675 1452 who staRTED WORLD WAR I?
1.249743 1598 what date did the american civil war start
1.272201 1586 what where the most important factors that led to the defeat of
the democrates in 1968?
1.272933  589 what happened in 1877 in us
1.277706  602 when did the civil war start and where
```

Possible Uses

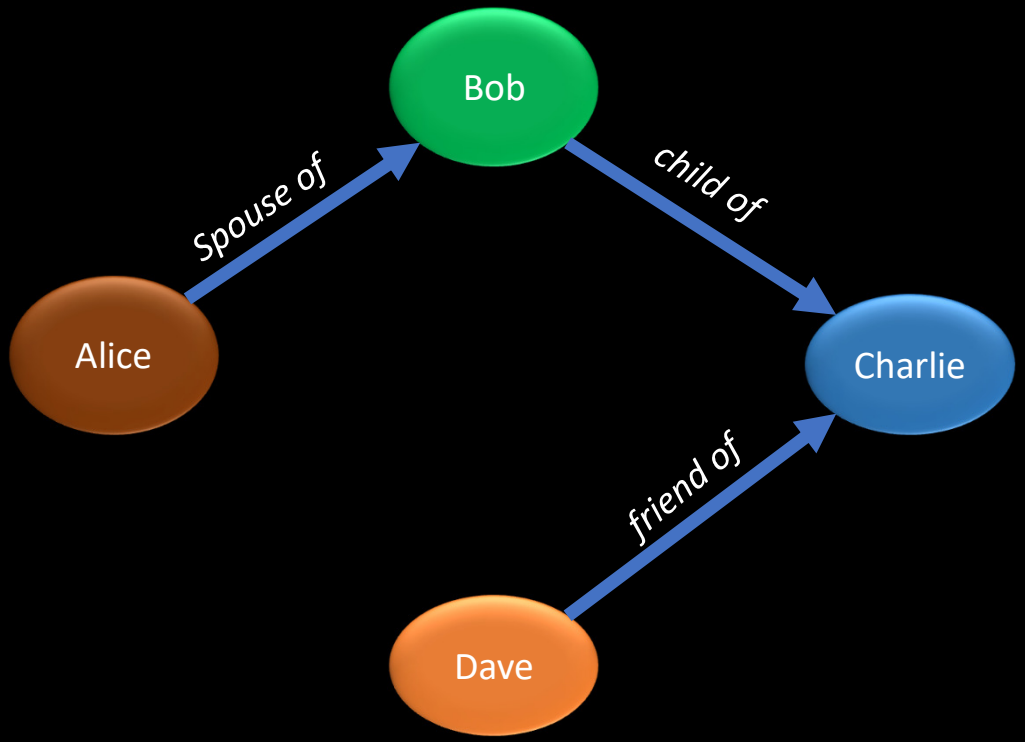
Long Term Memory in LLMs

Recommendation Engines

Data Taxonomy

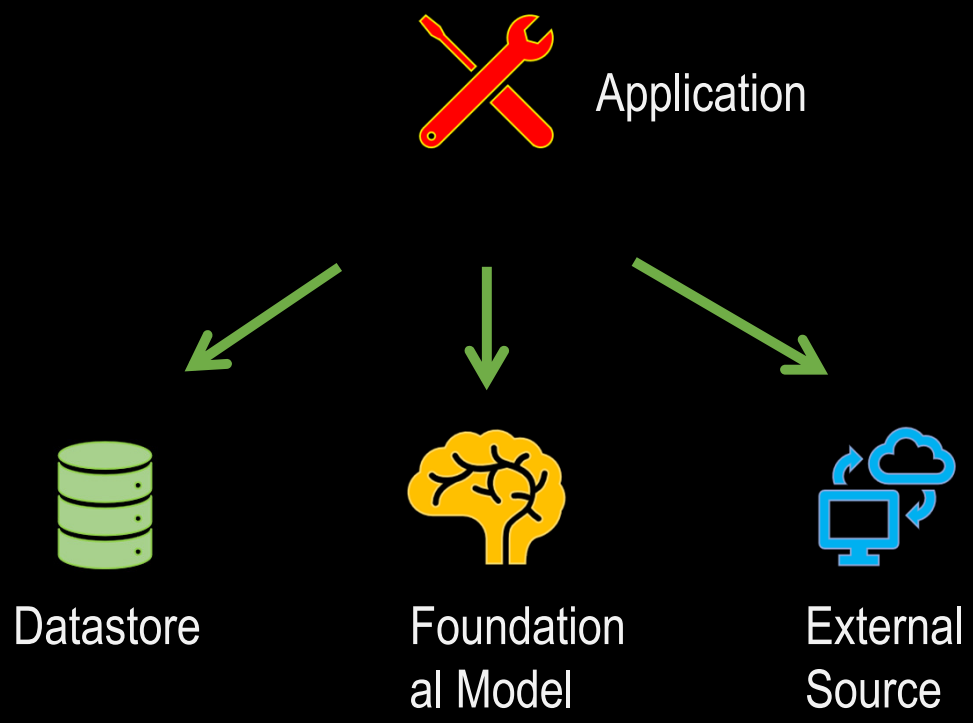
Removing Irrelevant Duplicates

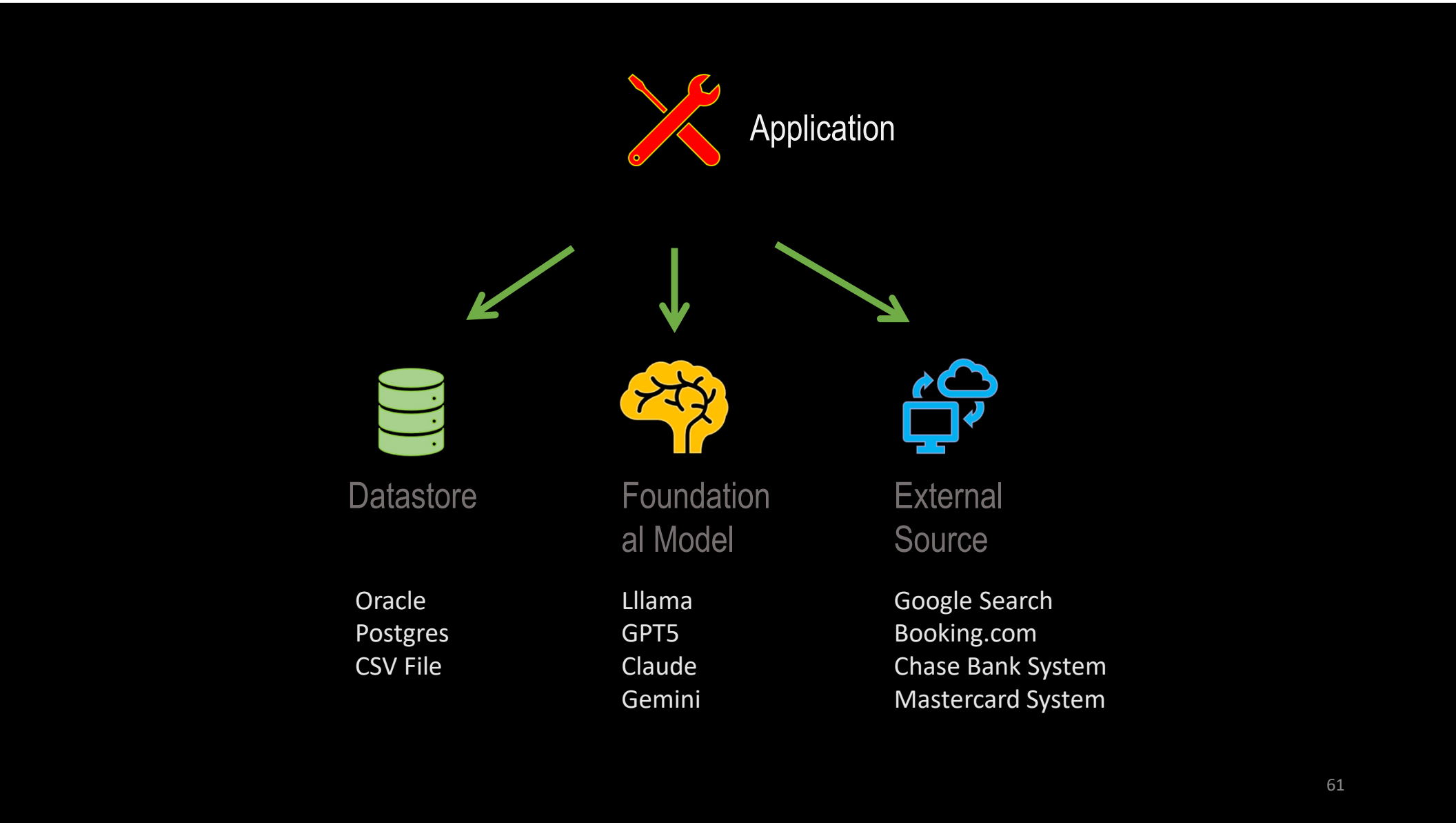
Graph Database



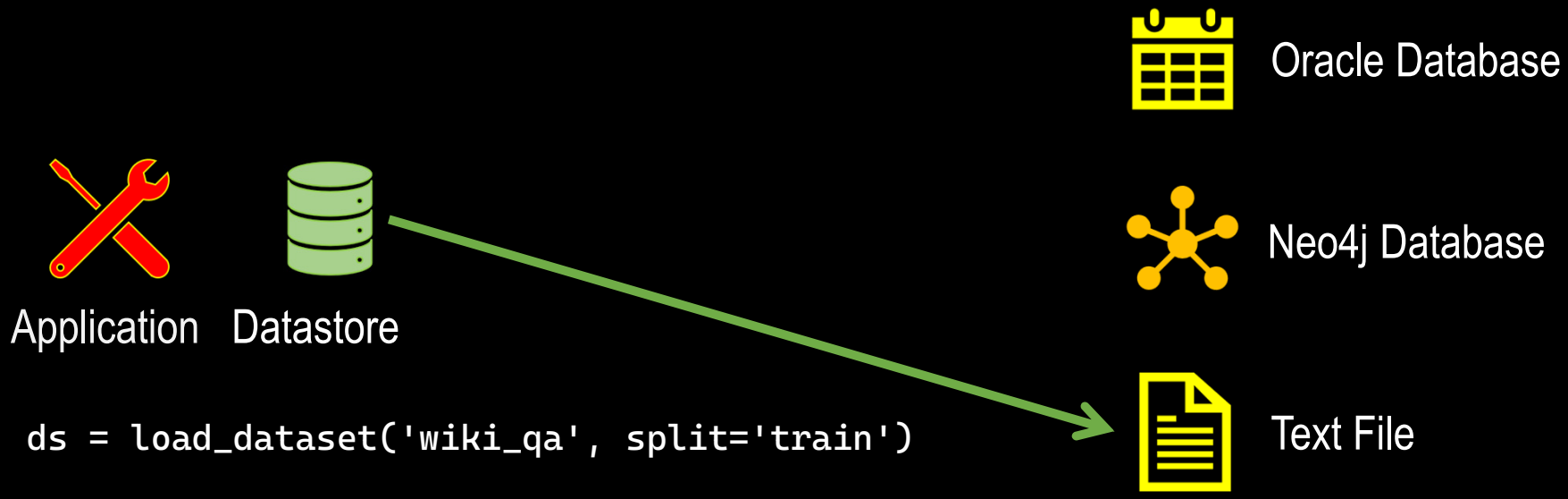
In summary

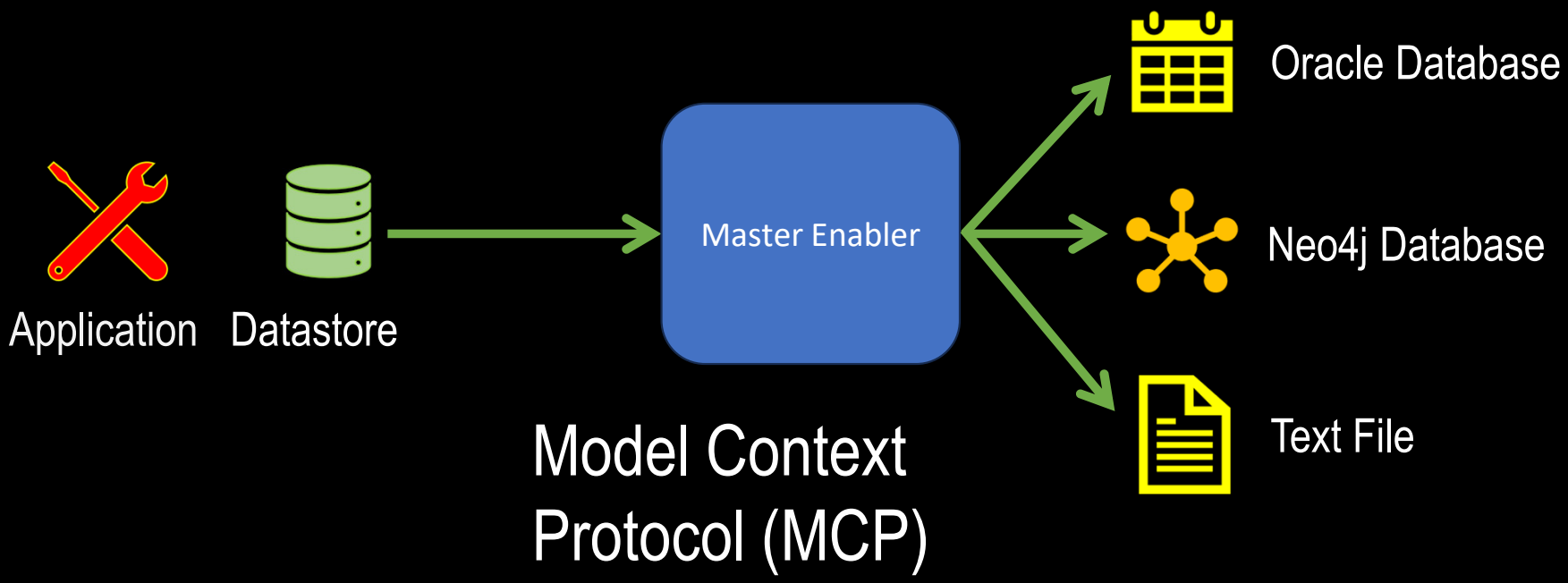
1. Vectors are numerical representation of dimensions of data
2. ML used to derive the dimensions
3. Vectors allow querying for data similar in meaning; not the same.
4. Vector distance between data elements shows similarity
5. Can be used for additional context to LLMs
6. Also used for other “meaning” searches such as in a data catalog.

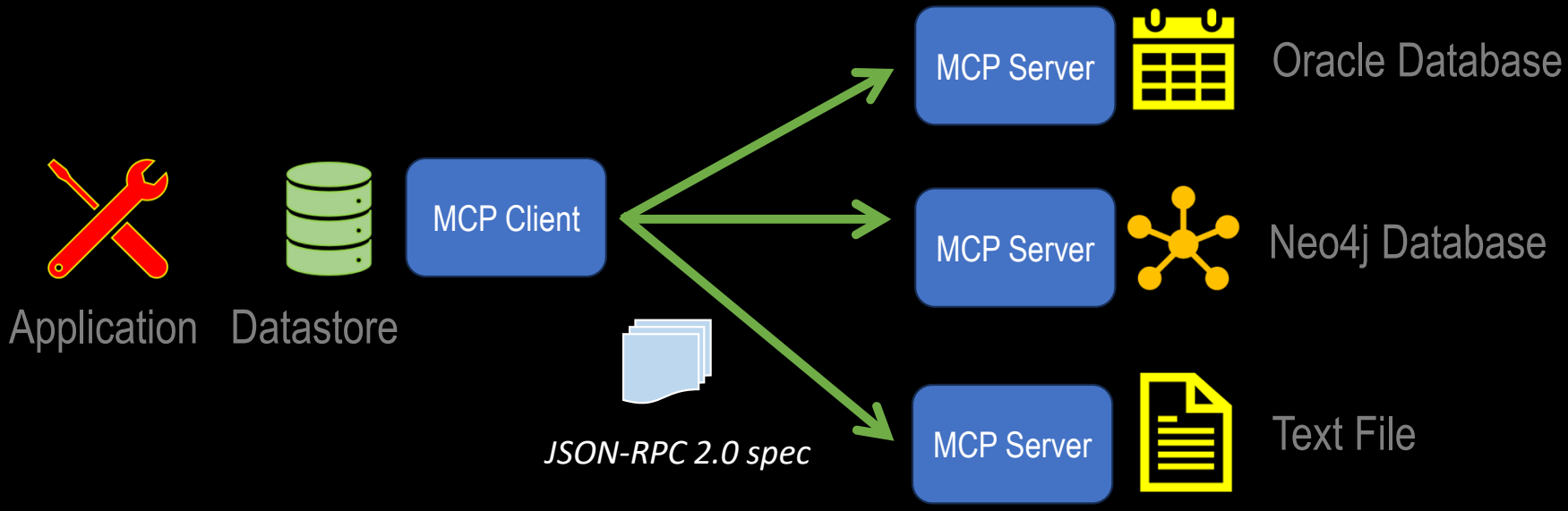

Application
Datastore
Foundation
al Model
External
Source

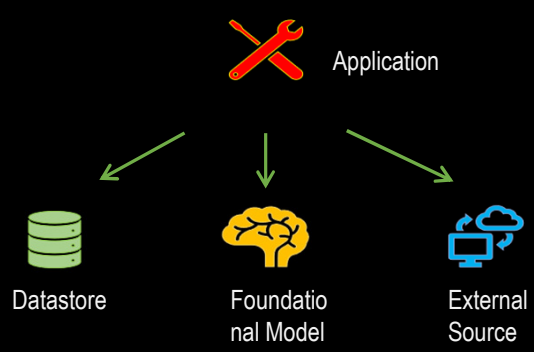

Application
Datastore
Foundation al Model
External Source
Oracle
Postgres
CSV File
Lllama
GPT5
Claude
Gemini
Google Search
Booking.com
Chase Bank System
Mastercard System

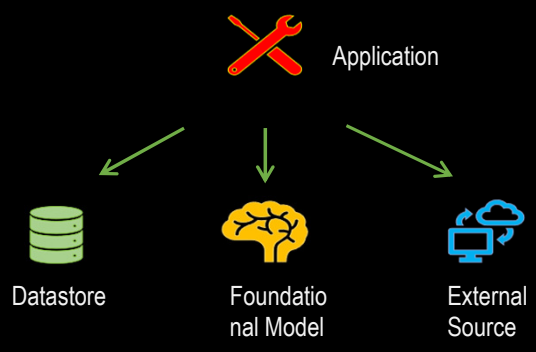

```
conn =
oracledb.connect(user="noug",password="noug",
dsn="localhost:1521/freepdb1")
```


Oracle Database
Neo4j Database
Text File
Application
Datastore
ds = load_dataset('wiki_qa', split='train')

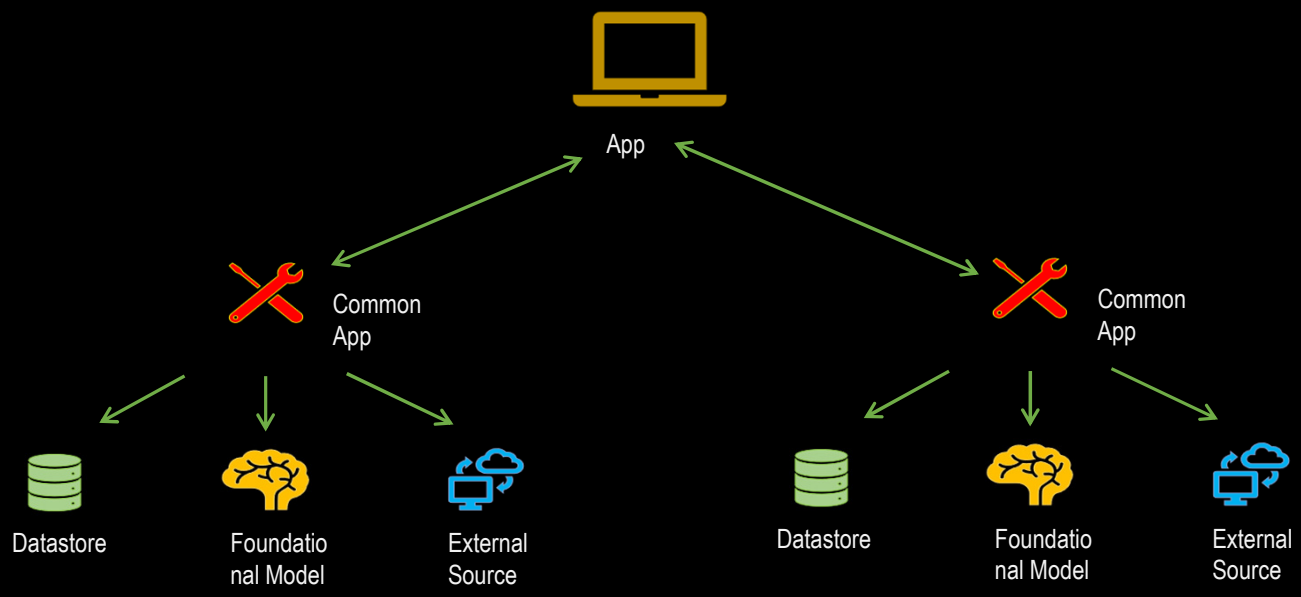

Application
Datastore
Master Enabler
Oracle Database
Neo4j Database
Text File
Model Context Protocol (MCP)

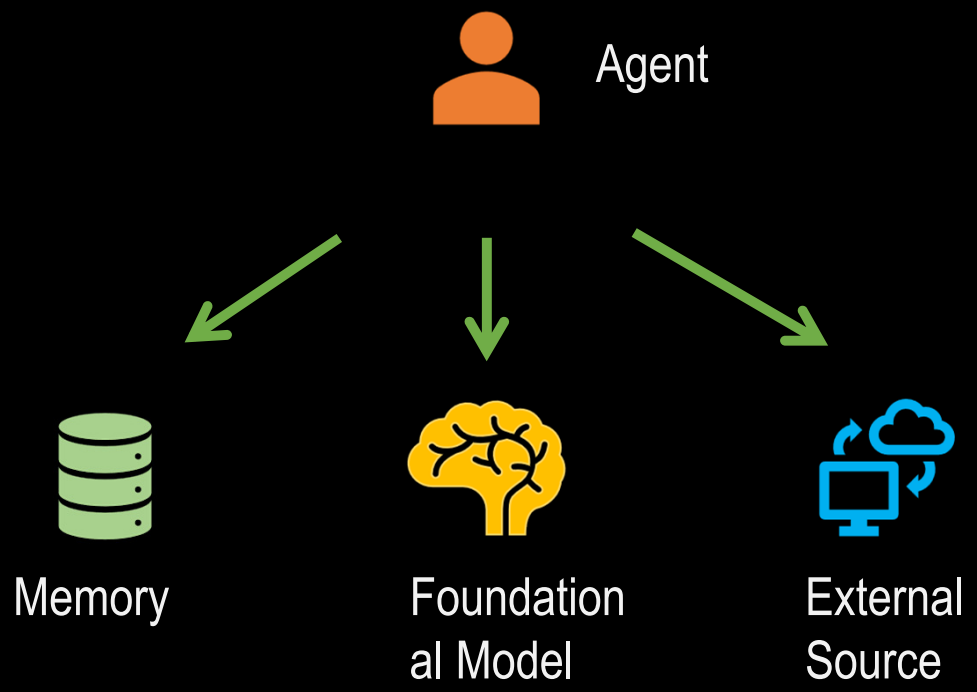

Application
Datastore
MCP Client
JSON-RPC 2.0 spec
MCP Server
MCP Server
MCP Server
Oracle Database
Neo4j Database
Text File


Application
Datastore
Foundatio
nal Model
External
Source


Application
Datastore
Foundatio
nal Model
External
Source

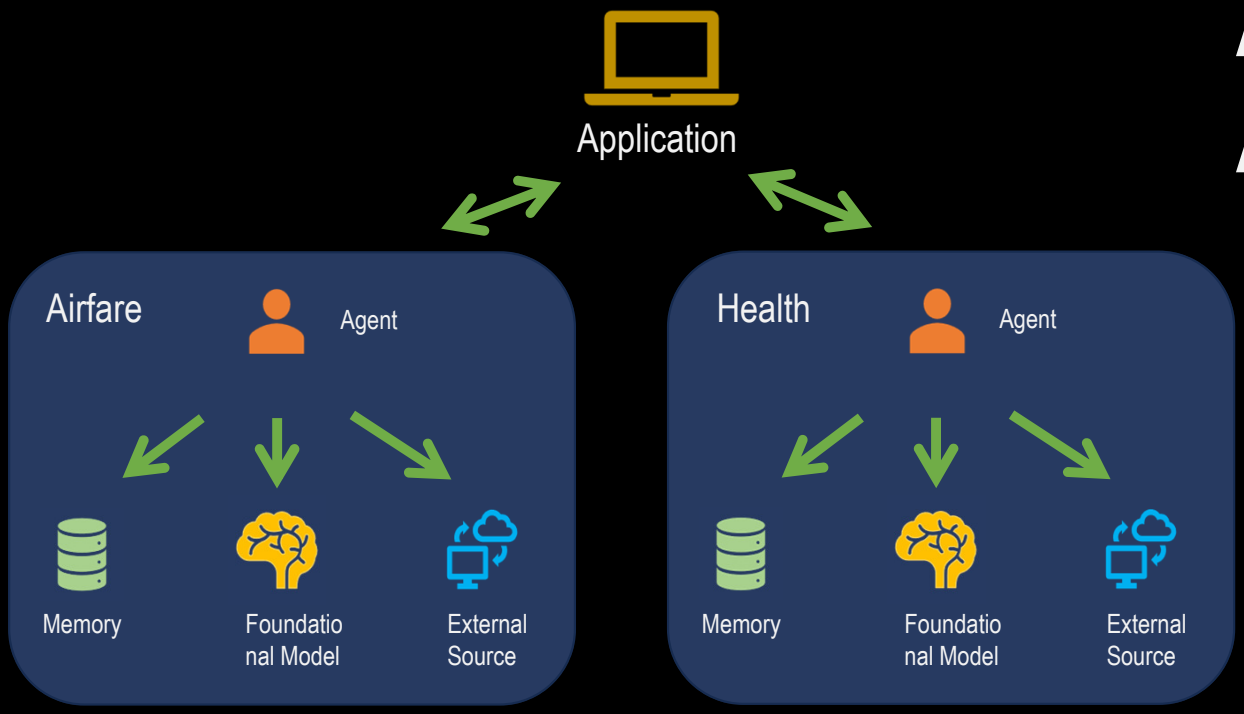
Agent

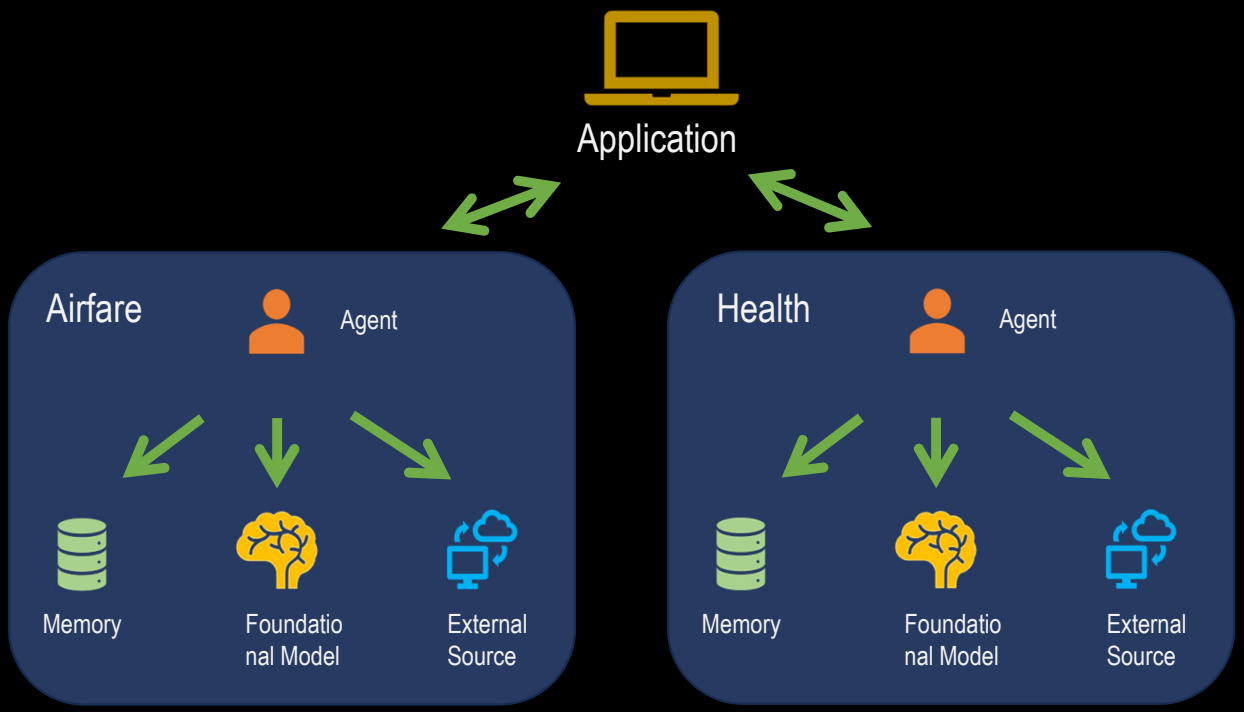

App
Common App
Common App
Datastore
Foundatio nal Model
External Source
Datastore
Foundatio nal Model
External Source



Agent is a specialized application that executes a specific type of task, e.g. finding out the best travel itinerary, or the best portfolio for wealth building. It can also take action such as booking the airfare.

Agentic Architecture





Agents need to:

- Advertise what they can and cannot do
- Specify how they will accept requests and send replies

A2A

Agent to Agent Comm Protocol

In Summary

- Generative AI is an effective prediction of next content – word, picture, pixel, audio, etc.
- Foundational models are trained on generic data to predict the next content. The underlying data are called parameters
- They use multiple layers of neural network to progressively predict the next content
- Fine tuning means changing one or more layers of NN by injecting your own parameters
- If a layer needs to get most up to date data, it can check an external data source, a process known as Retrieval Augmented Generation (RAG)

- Vector searches are useful for finding the nearest content, e.g. “Painter” is similar to “Artist” and not “Mason”. That is the foundation of models.
- Agents are generic applications that service other business applications, similar to utilities in application development. They are not standalone.
- Agents need three things: a foundation model, a memory and external sources
- Agents can talk to other agents using Agent 2 Agent (A2A) or Agent Context Protocol (ACP)
- Applications can insulate themselves from connecting to other external sources such as databases or web assets using Model Context Protocol (MCP)

Thank you!