

Understanding Big Data & AI



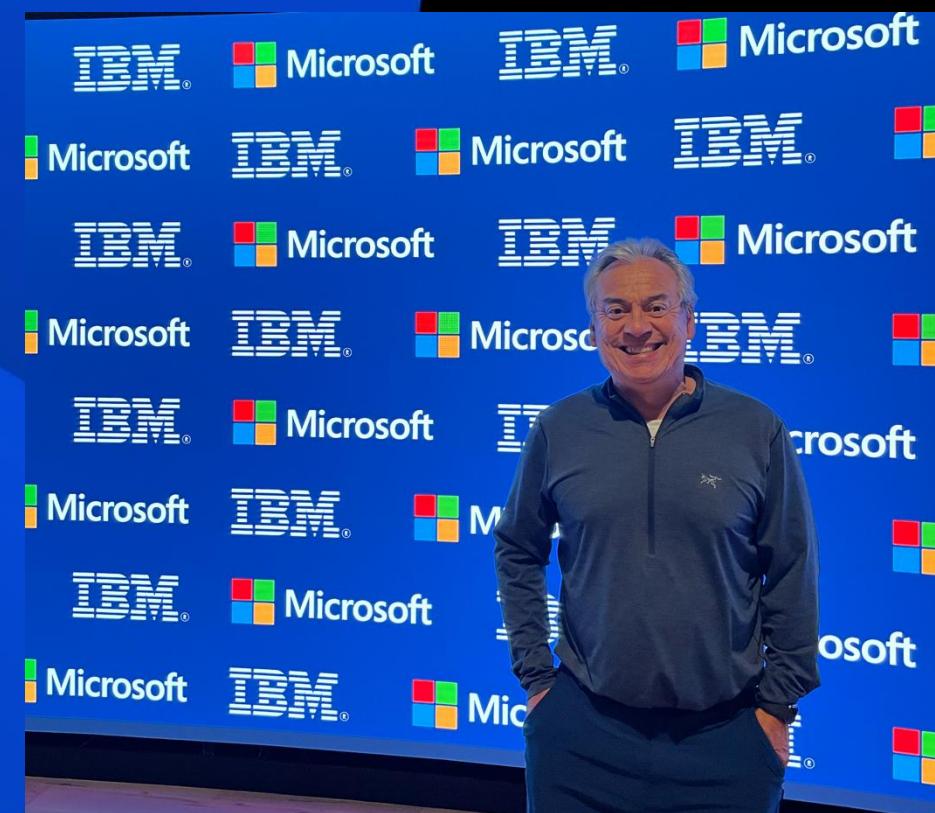
Ivan Portilla

Portilla@gmail.com

AI Technical Architect
02/26/2025

AI Fundamentals

- Ivan Portilla
- <https://github.com/jiportilla/giveback>





Robo del siglo

<https://www.bbc.com/news/world-latin-america-28783027>

Entity Resolution

Entity Resolution at Degrees of Difficulty

<https://senzing.com>

Exactly
Same

Bob Jones
123455

Bob Jones
123455

Fuzzy

Bob Jones
123455

Robert T Jonnes
000123455

Incompatible
Features

Bob Jones
123455

Agenda

Big Data & AI

Generative AI (ChatGPT)

Demo

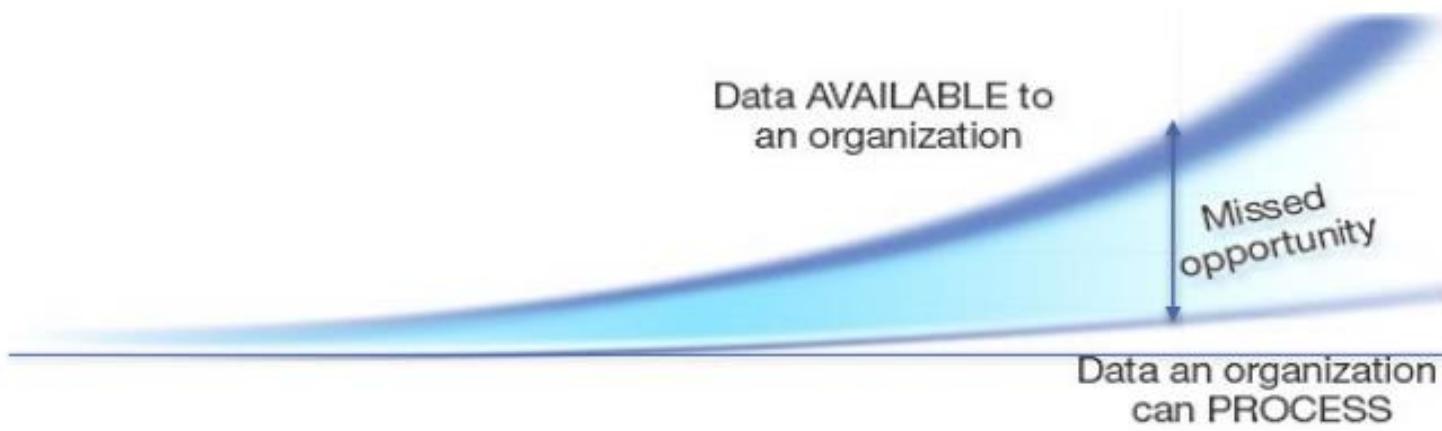
Agenda

Big Data & AI

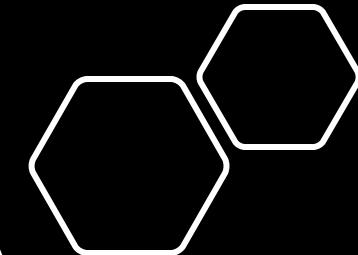
AI Essentials Framework

Demo

“Every day, we create 2.5 quintillion bytes of data”



Estimates are that less than 0.5% of data is ever analyzed!

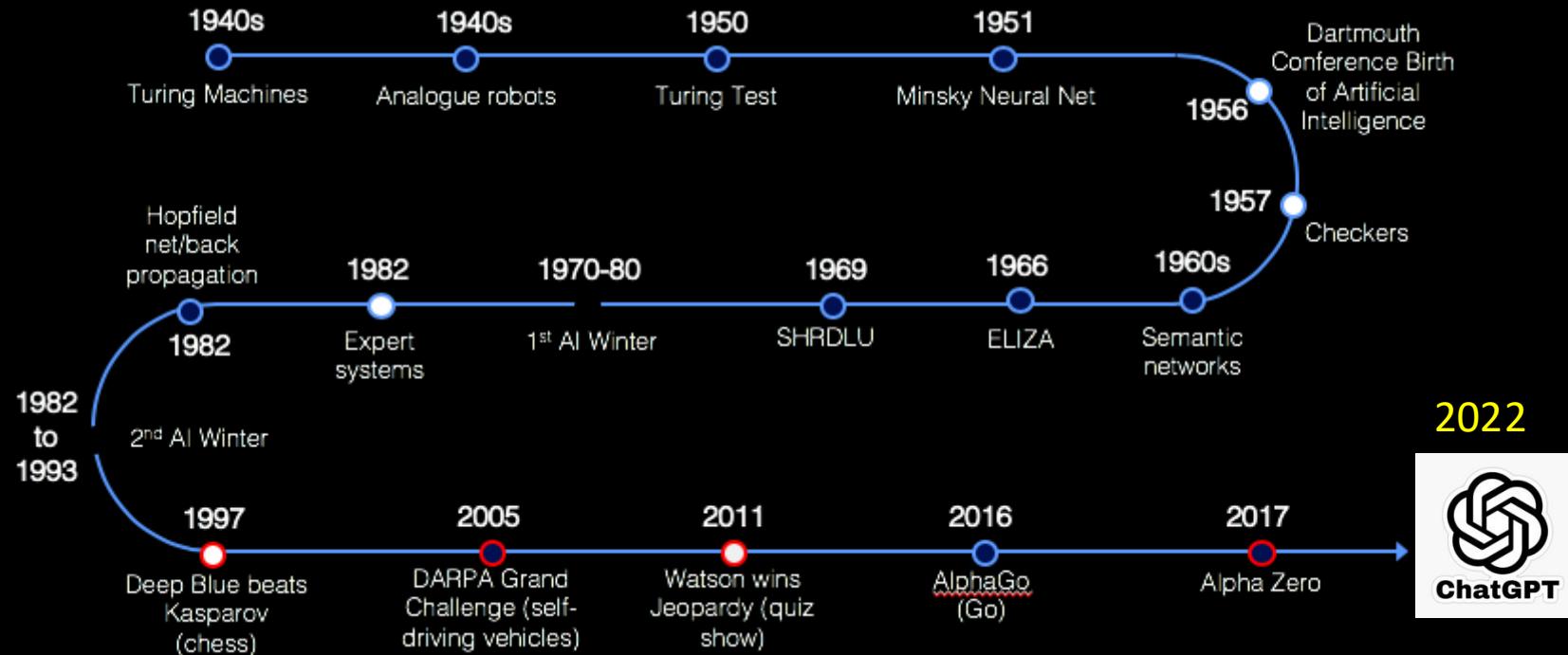


Understanding Big Data

Term	Factor	# RAMACs	# IPODs
1 Giga (GB)	10^9	200	
1 Tera (TB)	10^{12}	200K	200
1 Peta (PB)	10^{15}	200M	200K
1 Exa (EB)	10^{18}		200M
1 Zetta (ZB)	10^{21}		
?	10^{24}		
?	10^{100}		



A brief history of AI



ALAN TURING'S OFFICE

Here at his desk in Hut 8, Turing took the lead on breaking naval Enigma ciphers – something few thought could ever be done. His mathematical skills also enabled him to break other ciphers, including the complex Lorenz cipher where he used a method that became known as Turingery. Together with his fellow Codebreaker Gordon Welchman, he developed the Bombe machine to help speed up the codebreaking process.

IN THEIR WORDS

If anyone was indispensable to Hut 8 it was Turing. The pioneer work always tends to be forgotten when experience and routine later make everything seem easy, and many of us in Hut 8 felt that the magnitude of Turing's contribution was never fully realised by the outside world.

Hugh Alexander, Codebreaker, Hut 8



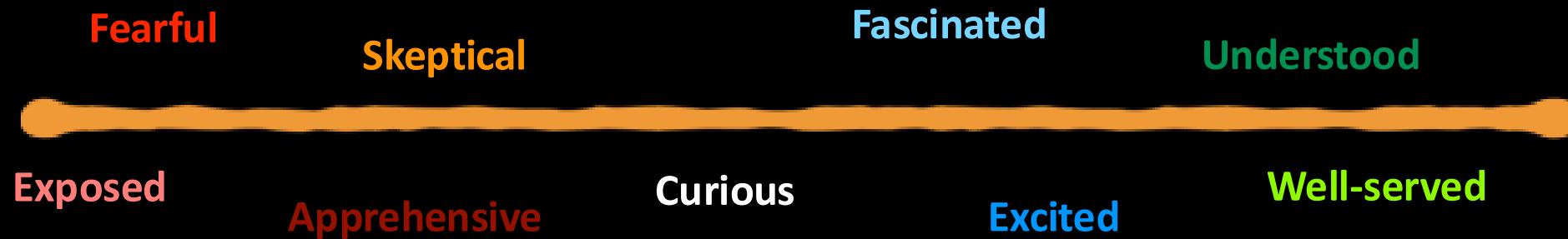
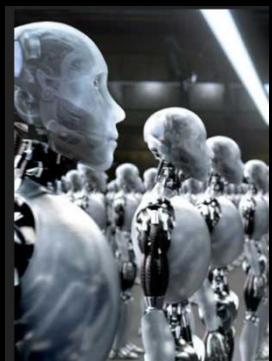
What is AI?

Slido.com

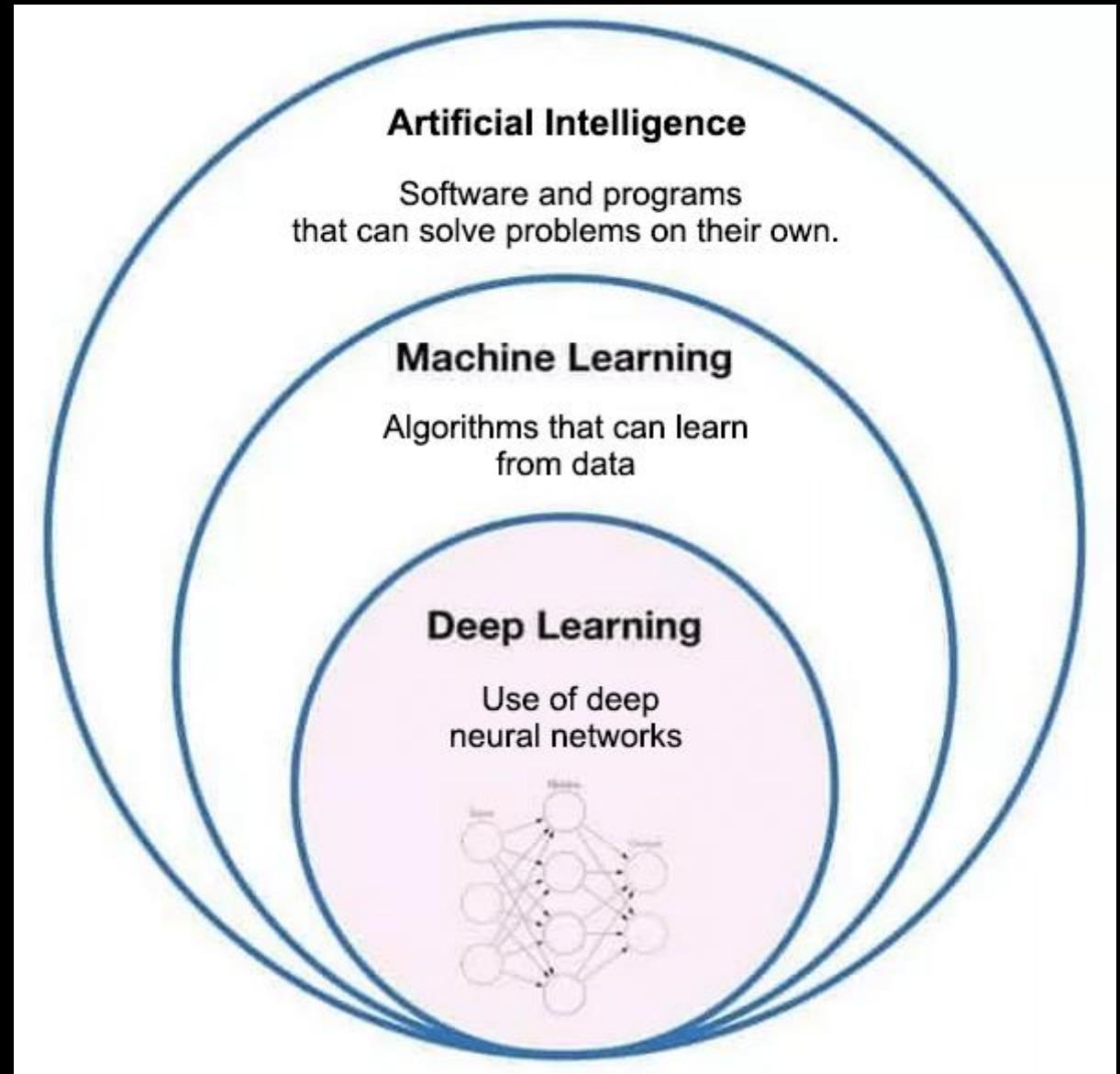
Code: 6772241

AI Perception

What do people feel about AI?



What we know

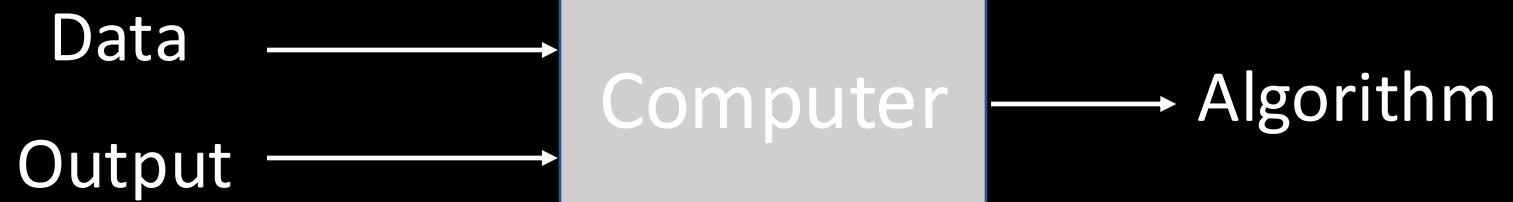


Why Machine Learning?

Traditional Programming



AI-Machine Learning



ML provides systems the ability to **automatically** learn from **experience** (without being explicitly programmed)

Intelligence demonstrated
by CU Students

Exercise three

PB & J Recipe

WikiHow to Make a Peanut Butter and Jelly Sandwich

Exercise 1

Program how to make a Peanut Butter & Jelly sandwich

1. Get ingredients (PB, Jelly, Bread)
2. Get cooking utensils (plate, knives)
3. Get two slices of bread
4. Open PB, Jelly jars
5. With a knife spread 2 Tablespoons of peanut butter on one piece of bread
6. With a different knife, spread 2 Tablespoons of jelly on the other slice of bread
7. Put the slices together
8. Toddler adaptation: cut off crusts before serving.
9. Place on plate & serve



Exercise two

Machine Learning.

Sample



Label



dog



cat



horse

Human Learning:

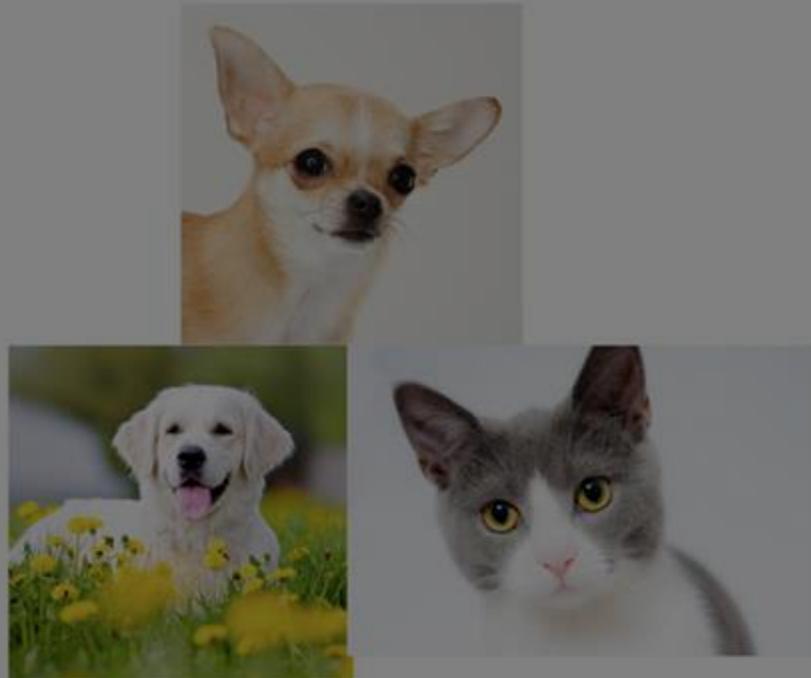
We learn through



Cat or Dogs

Long Ear Black nose

dog



Machine Learning:

Sample



Label



dog



cat



horse

Human Learning:

We learn through



Long Ear Black nose

dog



Examples

Diagrams

Comparisons

Exercise Two

A blurry photograph of a highway scene. In the foreground, the side of a white semi-truck is visible, showing its trailer and some red reflectors. On the left side of the road, there's a yellow diamond-shaped road sign with a black symbol on it. The road has a solid yellow line running down its center. In the background, there are green trees and hills under a clear sky.

Passing truck in
highway

Passing truck in highway

Don't Follow Closely Behind Prior to Passing

- As you prepare to pass, keep a healthy distance between your car and the truck in front of you. At minimum, 30 feet should separate the two vehicles. Signal clearly to indicate your next move.

Pass in the Left Lane

- The left lane is referred to as the 'passing lane' for a reason; it's the safest place to pass any vehicle, but particularly trucks and buses. It's frustrating to be stuck behind a slow truck in the left lane, but resist the urge to pass on the right side.

Don't Linger in the Truck's Blind Spot

- Trucks have huge blind spots on both sides. Pass promptly while still abiding by the speed limit. Don't hang out in the blind spot area, which, in the left (passing) lane, constitutes the truck's entire front half.

Recognition

- With GPS navigation, HiRes cameras, Laser-radar cameras: Location, speed of trucks, road conditions, emergency lane
- With Image Recognition: Turn signal (!lamp), paper or debris on the road

Insights

- Data is fed into multiple CPUs + collected data + rules
- Manufacturer knows who has the right of the way, when trucks can pass + MM miles driven have feedback on different driving style

Action (Cognitive Decision)

- Assist (alerts driver)
- Act (Pass or Not pass, Measure decision effects, feedback loop, and share w other cars)

NLU Demo

How is AI being used today?

Future of AI Example (Semantic Analysis)

The Wining Shot

The soccer game was nearly over. The two teams were tied, one to one.

Alice kicked the ball. Oops! She kicked it the wrong way.

"What a bad kick," Alice thought.

But wait!. Her teammate John jumped to stop the ball. The ball went off John into the goal!.

Alice's team won the game. Hooray! It was not a bad kick, after all.

Who won the game?

Why?

Future of AI Example

The Wining Shot

The soccer game was nearly over. The two teams were tied, one to one.

Alice kicked the ball. Oops! She kicked it the wrong way.

"What a bad kick," Alice thought.

But wait!. Her teammate John jumped to stop the ball. The ball went off John into the goal!!.

Alice's team won the game. Hooray! It was not a bad kick, after all.

Who won the game?

Answer: Alice's team
Correct

Explanation: Alice's team won the game because Alice's team had more points than the other team

✓ Correct

ChatGPT

JE

read this: The Wining Shot

The soccer game was nearly over. The two teams were tied, one to one.

Alice kicked the ball. Oops! She kicked it the wrong way.

"What a bad kick," Alice thought.

But wait!. Her teammate John jumped to stop the ball. The ball went off John into the goal!.

Alice's team won the game. Hooray! It was not a bad kick, after all.

ChatGPT

JE who won the game

 Alice's team won the game, even though Alice initially thought she had made a bad kick. John's quick thinking and intervention helped the team to score the winning goal, and they were able to secure victory in the game.

JE why?

 Alice's team won the game because John was able to use his skills and reflexes to intercept the ball, even though Alice's kick was not aimed in the right direction. John's intervention caused the ball to go into the goal, which resulted in a score for Alice's team. Since this happened near the end of the game and there was no time left for the other team to catch up, Alice's team won the game with a score of 2-1.

Agenda

Big Data & AI

Generative AI (ChatGPT)

Demo

Brief history of AI

Artificial Intelligence

Machine Learning

Deep Learning

Generative AI



Artificial Intelligence

Intelligence demonstrated by machines



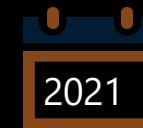
Machine Learning

Learn from data



Deep Learning

Model after the human brain (Neural Networks)



Generative AI

Create new written, visual, and auditory content

MSFT partnership with OpenAI



Ensure that artificial general intelligence (AGI) benefits humanity.



Empower every person and organization on the planet to achieve more

GPT-3

Generate and Understand Text

Codex

Generate and Understand Code

DALL·E

Generate images from text prompts



Generative AI

GPT-3

Prompt:

Write a tagline for an ice cream shop.

Response:

We serve up smiles with every scoop!

Codex

Prompt:

```
Table customers, columns =  
[CustomerId, FirstName,  
LastName, Company, Address,  
City, State, Country,  
PostalCode]
```

Create a SQL query for all customers in Texas named Jane
query =

Response:

```
SELECT *  
FROM customers  
WHERE State = 'TX' AND  
FirstName = 'Jane'
```

DALL·E

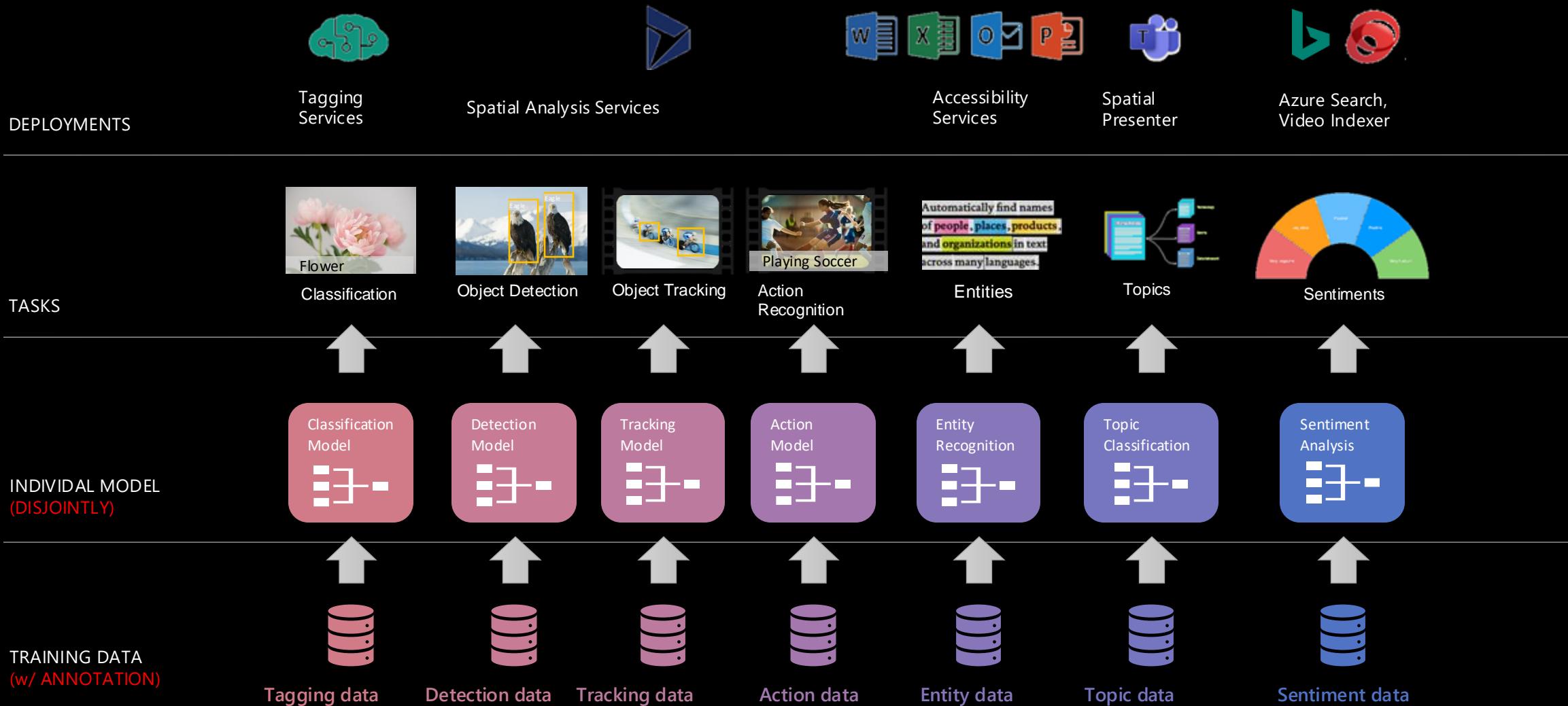
Prompt: A white Siamese cat

Response:

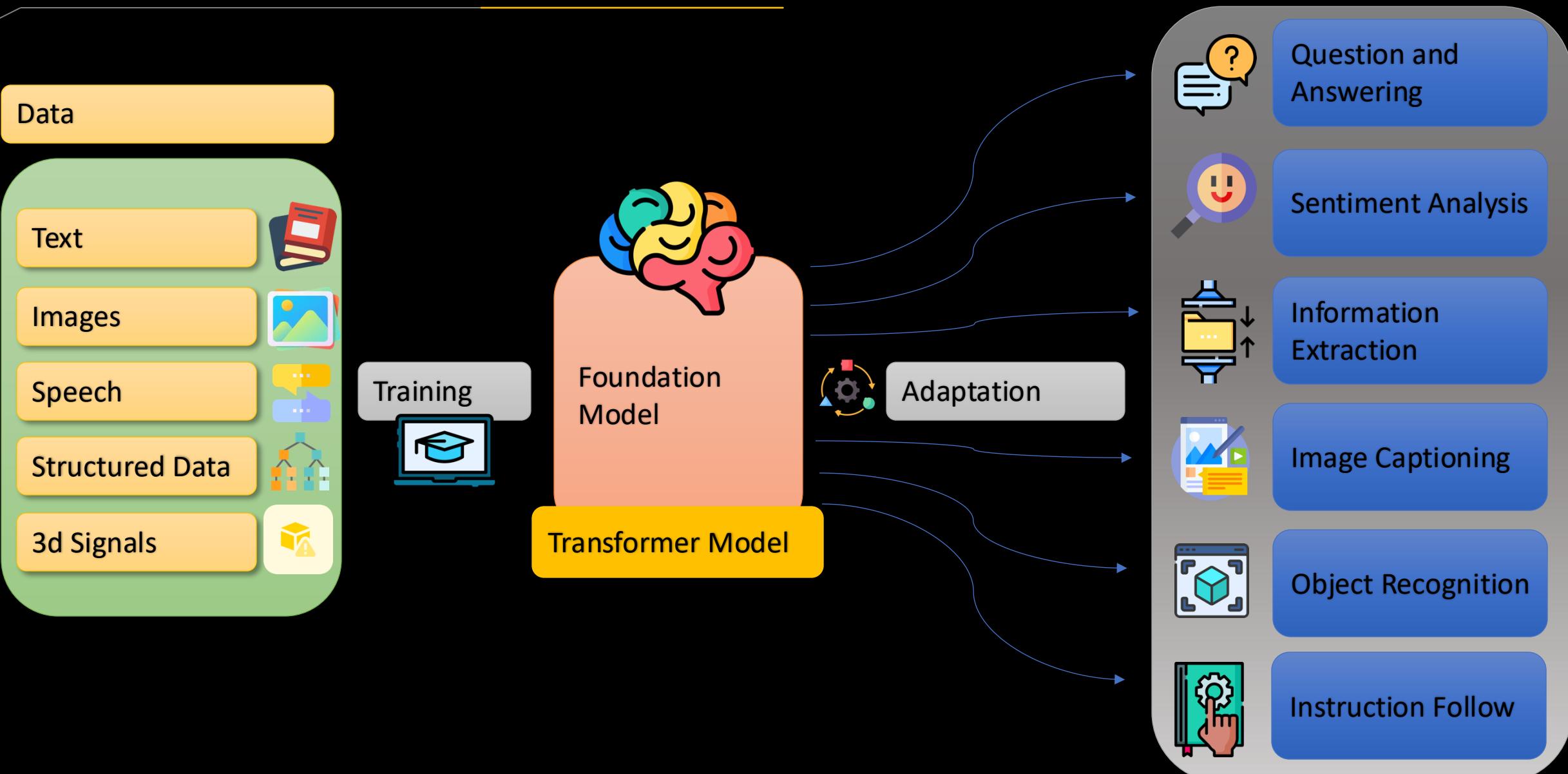


Traditional model development

High cost & slow deployment - Each service is trained disjointly



Foundation Models



Demos

Text Classification

Named Entity Recognition (Zero-Shot)

Extract the name and mailing address from this email:

Dear Kaylee,

It was great to talk to you at the seminar. I thought Peter's talk
was quite good.

Thank you for the book. Here's my address 995 Regent Dr,
Boulder, CO 80309

Best,

Marilú

Name: Marilú

Mailing Address: 995 Regent Dr, Boulder, CO 80309

Text Classification

Custom Named Entities (Few-Shot)

Extract job titles from the following sentences.

Sentence: Jane Doe has been working for IBM for 20 years as a Linux Engineer.

Job title: **Linux Engineer**

###

Sentence: Jane Doe has been working for IBM for 20 years and she loved it.

Job title: **none**

###

Sentence: Demis Hassabis| Director | Google

Job title: **Director**

###

Sentence: Juan is the CTO of Platform.sh, he was previously the CTO of Commerce Guys, a leading ecommerce provider.

Job title: **CTO**

Abstractive Summarization

Contact Center Summarization

Summarize this for a call center agent:

Agent: Thank you for calling ADM. Who am I am speaking to?

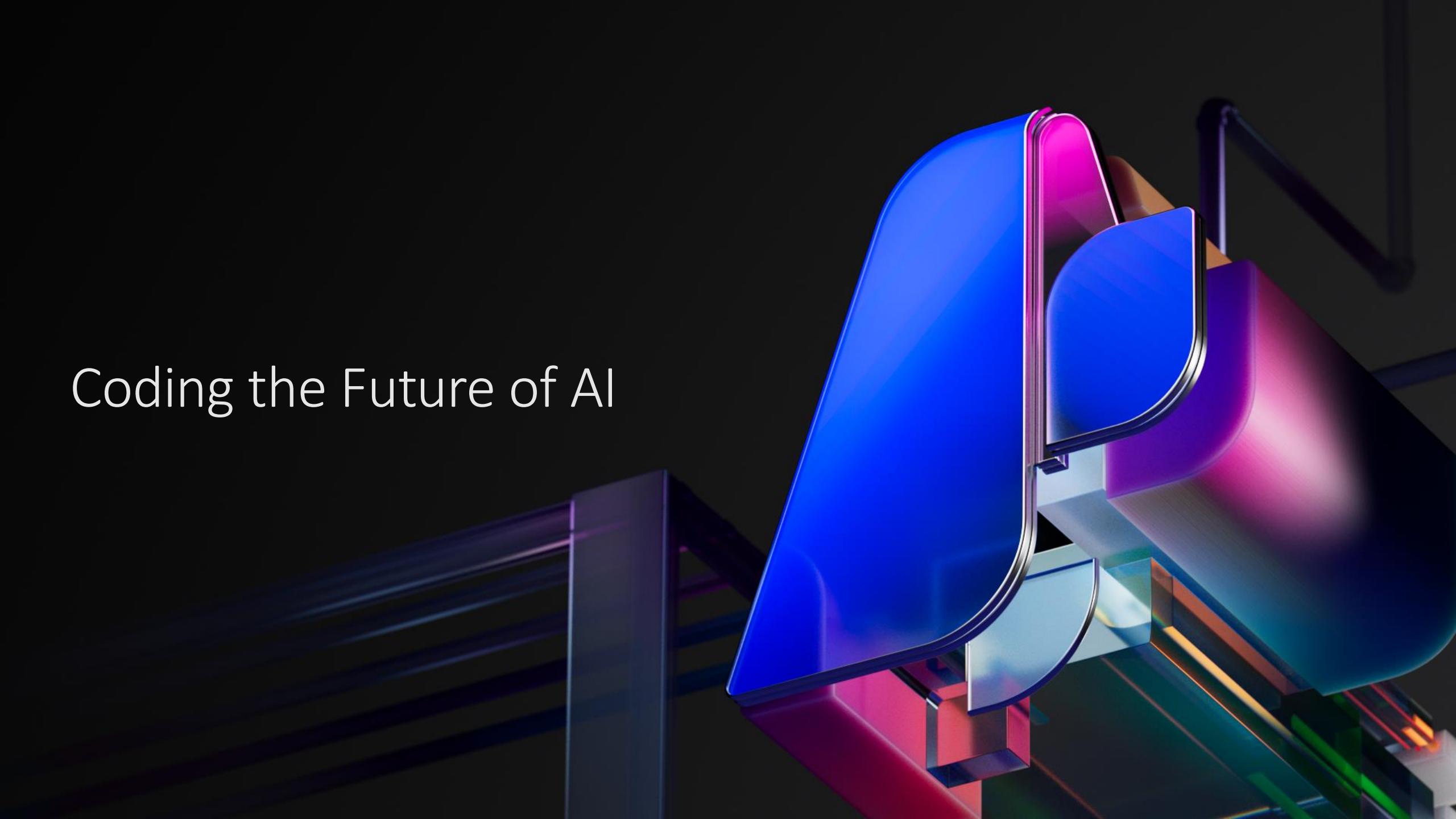
Customer: Hello, my name is Jeremiah Contreras. I own a small business and have some questions regarding payroll processing.

Agent: Good morning, Jeremiah, before we get started may I ask you a few questions so that I better answer your questions today?

Customer: Thank you that is quite helpful. Are there specific regulations that I need to follow?

Agent: Certain aspects of payroll processing are regulated by the Internal Revenue Service (IRS) and the Department of Labor (DOL)

The agent is speaking to a customer named Jeremiah Contreras. Jeremiah is asking for tips on manual payroll processing. The agent also informs Jeremiah that certain aspects of payroll processing are regulated by the Internal Revenue Service (IRS) and the Department of Labor (DOL).



Coding the Future of AI



**Every code program
will be reinvented
with AI**



**New code will be
built that weren't
possible before**



New code applications
built over the next 24-36
months



Organizations accelerating
code modernization to
bring AI into existing apps



Developers using AI in
their toolset

The reality

14%

**Conversation rate from
AI PoC to production**

74%

**Organizations in GenAI
planning/
experimentation stage**

Factors hindering GenAI production

41% Lack of developer skills and tools

38% Excessive costs outstripping ROI goals

36% Infrastructure performance and availability

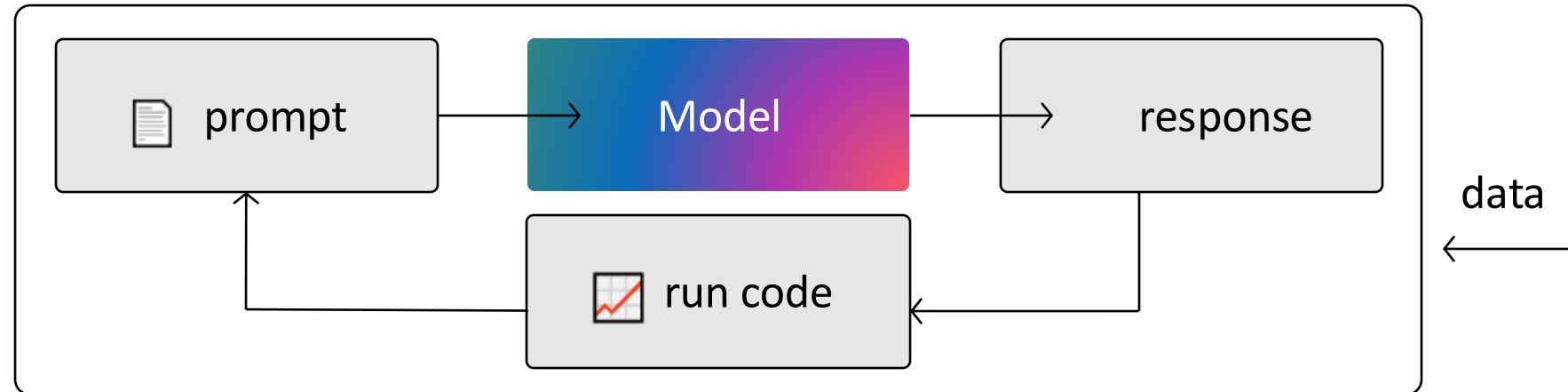
35% Poor IT and business coordination

De-mystifying AI Development

At a basic level

Providing prompt to a model

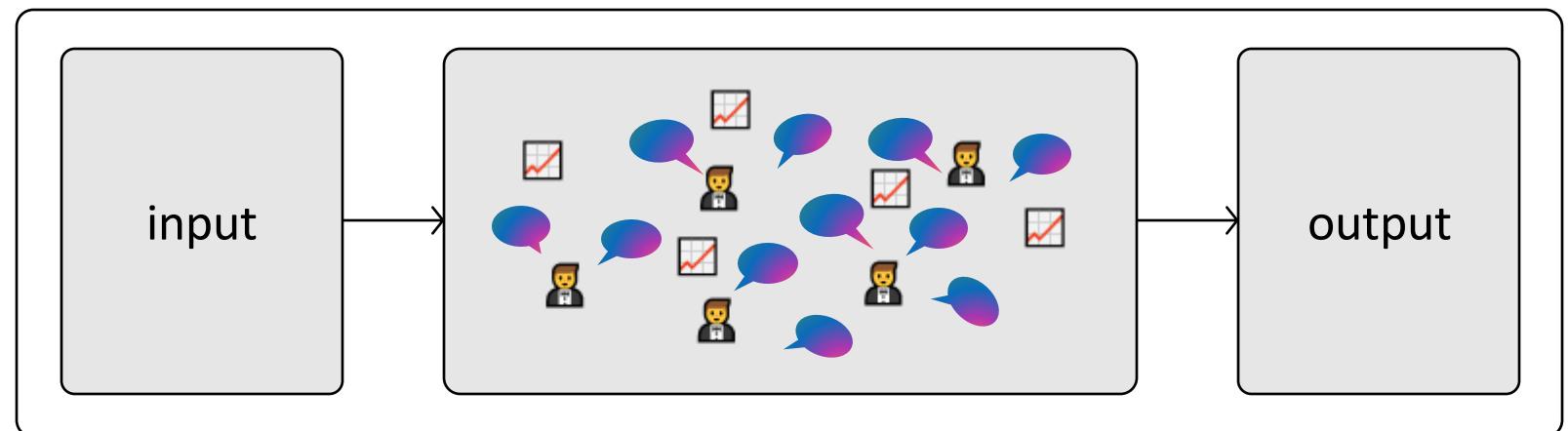
Acting on the response



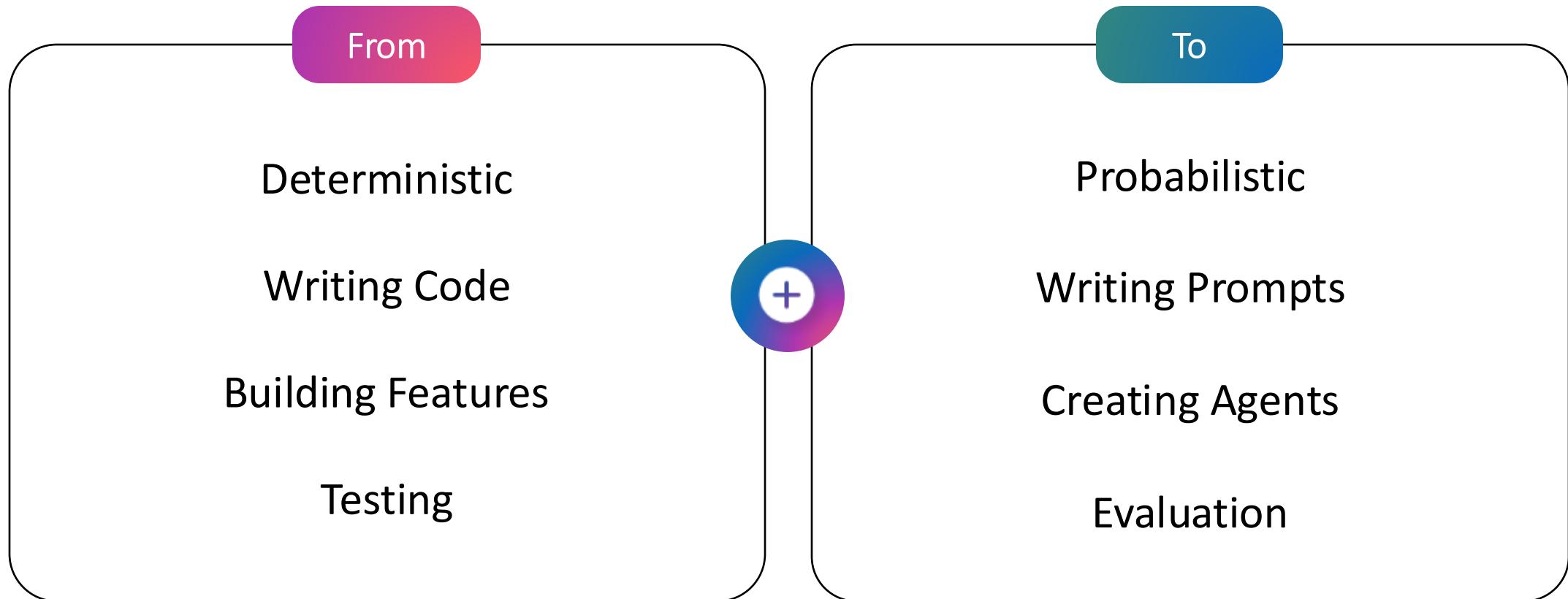
At scale

Holding many different conversations with models & agents

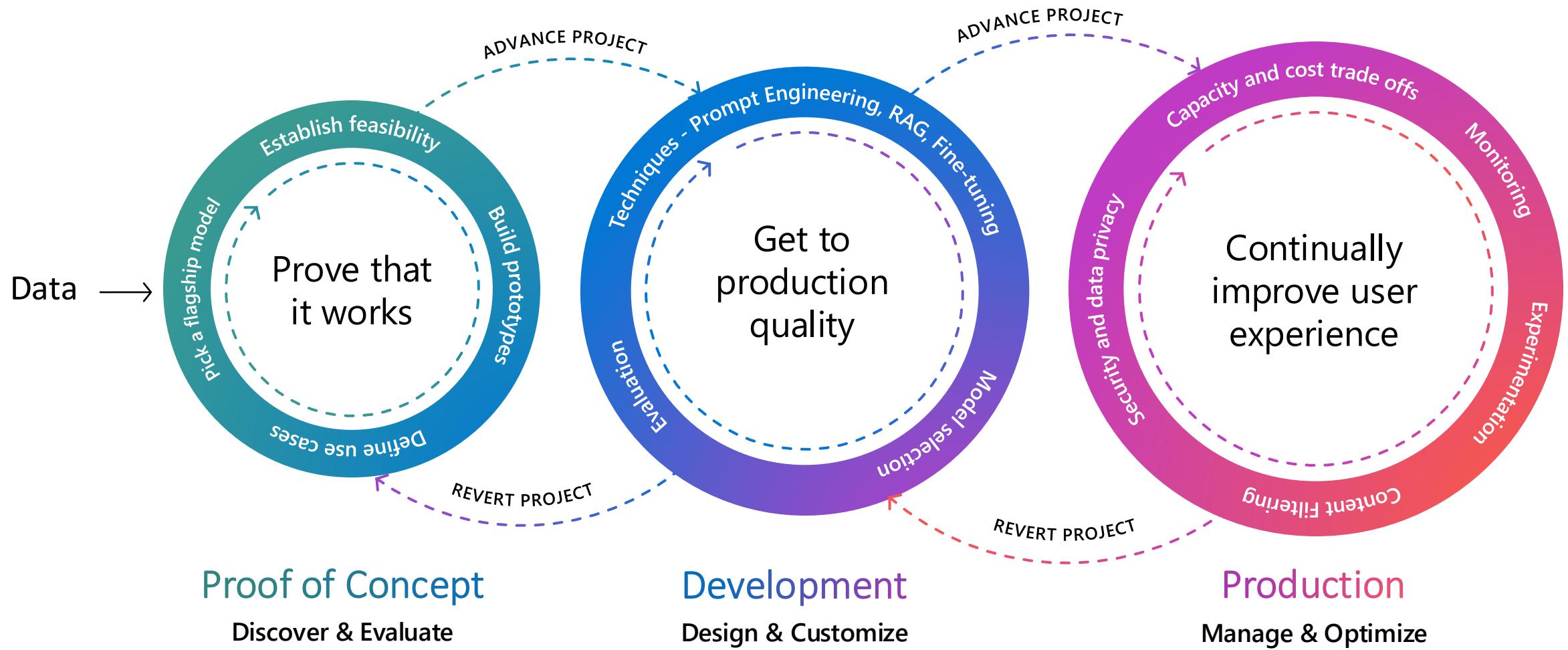
Extracting information into useful results



The way we write code is changing



Developers start small and iterate

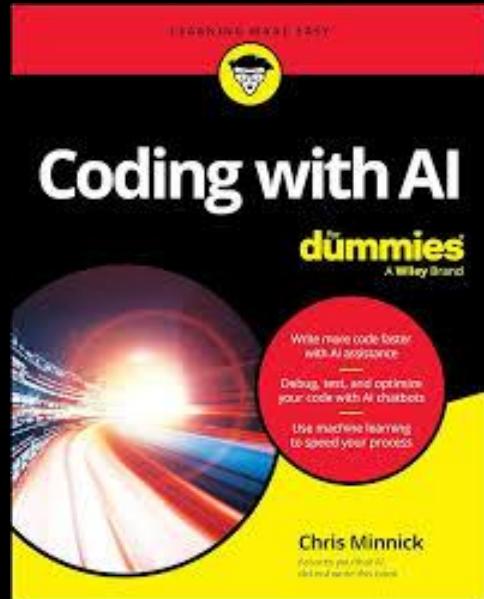


A photograph of two women in an office environment. In the foreground, a Black woman with curly hair and glasses, wearing a green cardigan over a brown t-shirt, is smiling broadly at the camera. Behind her, another woman with short dark hair, wearing a tan corduroy jacket over a yellow shirt, looks off-camera with a neutral expression. The office has a modern design with exposed wooden beams on the ceiling, large windows, and a television screen showing a video in the background.

On to production

Leveraging AI in Coding

Techniques, Tools, and Best Practices



I. Techniques & Technologies

1. Benefits
2. Parsing ML/DL
3. AI coding tools
4. Coding w chatbots

II. Using AI to write code

5. From plan to prototype
6. Formatting & improving code
7. Find, eliminate bugs
8. Translate & optimize code

III. Test, Document & Maintain code

9. Testing your code
10. Documenting your code
11. Maintaining your code

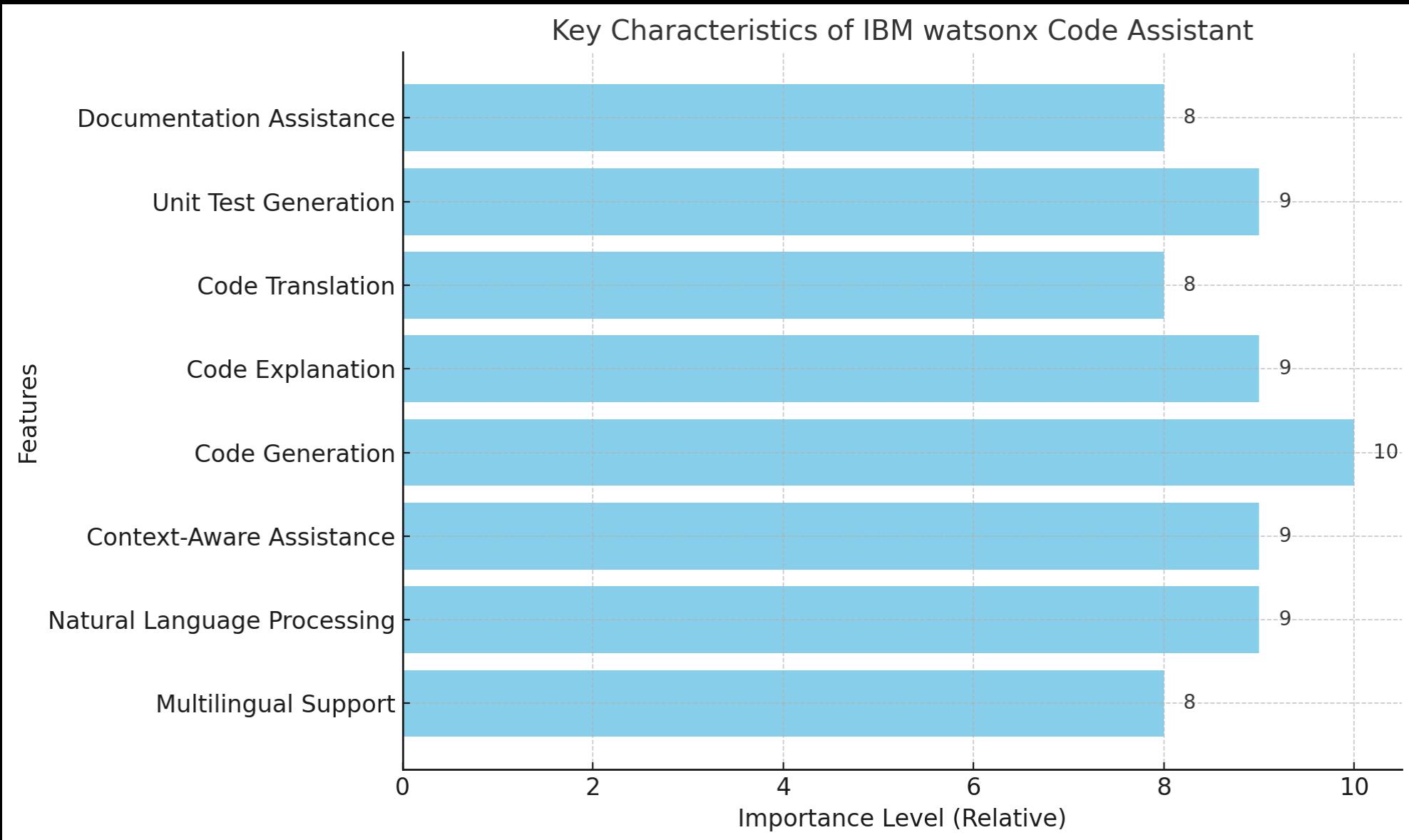
AI Mentor

55% faster coding
60% code acceptance
75% less frustrated



Aspect	IBM Watsonx Code Assistant	GitHub Copilot									
Ease of Use	Highly accessible due to natural language prompts, suitable for all skill levels.	Easy to integrate with IDEs like VS Code and JetBrains tools.									
Supported Languages	Multilingual support (Python, Java, C++, Go, TypeScript, etc.).	Similar multilingual support with strengths in Python, JavaScript, and TypeScript.									
Code Explanation	Offers detailed explanations for existing code, making it great for understanding unfamiliar code.	Limited explanation capabilities; focuses more on code generation.									
Context-Awareness	Advanced context awareness for specific tasks and workflows.	<table border="1"><thead><tr><th>Developer Level</th><th>IBM Watsonx Code Assistant</th><th>GitHub Copilot</th></tr></thead><tbody><tr><td>Junior Developers</td><td>- May feel overwhelming due to advanced features.- Enterprise focus may limit accessibility.</td><td>- Intuitive and easy to use.- Helps juniors learn coding patterns through autocompletion.</td></tr><tr><td>Senior Developers</td><td>- Tailored for advanced use cases, including cross-language projects and enterprise workflows.- Documentation and explanation aid in team collaboration.</td><td>- Limited advanced features.- Best for rapid prototyping and coding tasks rather than deep customization.</td></tr></tbody></table>	Developer Level	IBM Watsonx Code Assistant	GitHub Copilot	Junior Developers	- May feel overwhelming due to advanced features.- Enterprise focus may limit accessibility.	- Intuitive and easy to use.- Helps juniors learn coding patterns through autocompletion.	Senior Developers	- Tailored for advanced use cases, including cross-language projects and enterprise workflows.- Documentation and explanation aid in team collaboration.	- Limited advanced features.- Best for rapid prototyping and coding tasks rather than deep customization.
Developer Level	IBM Watsonx Code Assistant	GitHub Copilot									
Junior Developers	- May feel overwhelming due to advanced features.- Enterprise focus may limit accessibility.	- Intuitive and easy to use.- Helps juniors learn coding patterns through autocompletion.									
Senior Developers	- Tailored for advanced use cases, including cross-language projects and enterprise workflows.- Documentation and explanation aid in team collaboration.	- Limited advanced features.- Best for rapid prototyping and coding tasks rather than deep customization.									
Integration	May require specific tools for integration, designed for IBM's ecosystem.	Seamless integration with GitHub repositories, VS Code, and JetBrains IDEs.									
Pricing	Enterprise-focused pricing; suitable for teams and large-scale projects.	Subscription-based pricing with free tiers for students and small projects.									
Customization	Allows customization for domain-specific tasks using IBM's Granite models.	Limited customization but supports open-source models.									
Best Use Cases	Enterprise-grade solutions, complex workflows, and teams needing high context-awareness and documentation.	Individual developers or teams looking for rapid prototyping and integration with GitHub workflows.									

Code
smarter,
not
harder:
watsonx
Code
Assistant



Code generation



Hi, how can I help you today?

You

Write a Python script that connects to a MongoDB database, retrieves documents from a 'users' collection, and prints usernames and their corresponding email addresses.

:



Assisted by watsonx Code Assistant

```
import pymongo
```

Connect to MongoDB

```
client = pymongo.MongoClient("mongodb://localhost:27017/")
```

Access the 'users' collection

```
db = client["mydatabase"]
```

```
collection = db["users"]
```

Retrieve documents and print usernames and email addresses

```
for document in collection.find():
```

```
    print(f"Username: {document['username']}, Email: {document['email']}")
```



Note: This script assumes that a MongoDB database named 'mydatabase' exists on the local machine, and a 'users' collection with documents containing 'username' and 'email' fields.



Code completion

primes.py 1 ● main.py 2 Settings

Users > jportilla > Downloads > primes.py > ...

```
1 #calculate primes less 100
2 #172.27.36.41:11434
3
4 def calculatePrimes(n):
5     primes = []
    for num in range(2, n+1):
        is_prime = True
        for i in range(2, int(num**0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            primes.append(num)
    return primes
```

Code Explanation

Chat session 1

watsonx

Hi, how can I help you today?

You

main.py explain this code :

watsonx

This code is a Python script that uses the MLflow library to train a machine learning model. The script takes in several input arguments, including the path to the input data, the test-train ratio, the number of estimators, the learning rate, and the name of the registered model. The script starts by parsing the input arguments using the argparse module. It then starts a new MLflow run and enables autologging for the sklearn module. Next, the script prepares the data by reading an Excel file using the pandas library, splitting the data into training and testing sets, and extracting the label column. The script then trains a GradientBoostingClassifier model using the training data and logs the number of samples and features. It also logs the classification report for the predicted values. Finally, the script registers the model with MLflow and saves it to a file. Overall, this code is a good example of how to use MLflow to train and register a machine learning model in Python.

Upvote

src > main.py > ...

```
10 ✓ def main():
11     """Main function of the script."""
12
13     # input and output arguments
14     parser = argparse.ArgumentParser()
15     parser.add_argument("--data", type=str, help="path to input data")
16     parser.add_argument("--test_train_ratio", type=float, required=False, default=0.25)
17     parser.add_argument("--n_estimators", required=False, default=100, type=int)
18     parser.add_argument("--learning_rate", required=False, default=0.1, type=float)
19     parser.add_argument("--registered_model_name", type=str, help="model name")
20     args = parser.parse_args()
21
22     # Start Logging
23     mlflow.start_run()
24
25     # enable autologging
26     mlflow.sklearn.autolog()
27
28 #####
29 #<prepare the data>
30 #####
31 print(" ".join(f'{k}={v}' for k, v in vars(args).items()))
32
33 print("input data:", args.data)
34
35 credit_df = pd.read_excel(args.data, header=1, index_col=0)
36
37 mlflow.log_metric("num_samples", credit_df.shape[0])
38 mlflow.log_metric("num_features", credit_df.shape[1] - 1)
39
40 ✓ train_df, test_df = train_test_split(
41     credit_df,
42     test_size=args.test_train_ratio,
43 )
44 #####
45 #</prepare the data>
```

Code Translation

Chat session 1

AI

write similar function in java main.py :

watsonx

```
// Assisted by watsonx Code Assistant
//main.py
import os
import argparse
import pandas as pd
import mlflow
import mlflow.sklearn
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split

def main():
    """Main function of the script."""

    // input and output arguments
    parser = argparse.ArgumentParser()
    parser.add_argument("--data", type=str, help="path to input data")
    parser.add_argument("--test_train_ratio", type=float, required=False, default=0.25)
    parser.add_argument("--n_estimators", required=False, default=100, type=int)
    parser.add_argument("--learning_rate", required=False, default=0.1, type=float)
    parser.add_argument("--registered_model_name", type=str, help="model name")
    args = parser.parse_args()

    // Start Logging
    mlflow.start_run()

    // enable autologging
    mlflow.sklearn.autolog()

#####
#<prepare the data>
#####
print(" ".join(f"{k}={v}" for k, v in vars(args).items()))

print("input data:", args.data)

credit_df = pd.read_excel(args.data, header=1, index_col=0)

mlflow.log_metric("num_samples", credit_df.shape[0])
mlflow.log_metric("num_features", credit_df.shape[1] - 1)

train_df, test_df = train_test_split(
    credit_df,
    test_size=args.test_train_ratio,
)
#####
#</prepare the data>
#####

#####
#<train the model>
#####
# Extracting the label column
y_train = train_df.pop("default payment next month")

# convert the dataframe values to array
X_train = train_df.values

# Extracting the label column
y_test = test_df.pop("default payment next month")

# convert the dataframe values to array
X_test = test_df.values
```

Enter / for commands

Code Test

Chat session 1

Prompt library

- Develop a GoLang HTTP server that serves a simple REST API for managing a collection of books. Implement CRUD operations (Create, Read, Update, Delete) for book entities, storing data in an in-memory data structure.
- Develop JUnit test cases for a Java class that represents a simple calculator with methods for addition, subtraction, multiplication, and division.
- Write Golang test functions for a package that handles image processing. Include test cases for image resizing and format conversion.
- Write JUnit test cases for a Java class that implements a stack data structure. Include tests for push, pop, and peek operations, as well as handling edge cases such as an empty stack.
- Add comments to a C++ class representing a file handling utility. Include comments for methods handling file reading, writing, and error handling.

watsonx

Here's an example of JUnit test cases for a Java class that implements a stack data structure:

```
// Assisted by watsonx Code Assistant
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class StackTest {

    Stack<Integer> stack = new Stack<>();

    @Test
    public void testPush() {
        stack.push(1);
        assertEquals(1, stack.peek());
        stack.push(2);
        assertEquals(2, stack.peek());
    }

    @Test
    public void testPop() {
        stack.push(1);
        stack.push(2);
        assertEquals(2, stack.pop());
        assertEquals(1, stack.pop());
    }

    @Test
    public void testPeek() {
        stack.push(1);
        assertEquals(1, stack.peek());
        stack.push(2);
        assertEquals(2, stack.peek());
    }

    @Test
    public void testEmptyStack() {
        Exception exception = assertThrows(Exception.class, () -> stack.peek());
        assertEquals("Stack is empty", exception.getMessage());
    }
}
```

Code Translation

/translate from python to java main.py



watsonx

```
// Assisted by watsonx Code Assistant
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
import java.util.stream.Collectors;

public class Main {
    public static void main(String[] args) {
        String data = args[0];
        double testTrainRatio = Double.parseDouble(args[1]);
        int nEstimators = Integer.parseInt(args[2]);
        double learningRate = Double.parseDouble(args[3]);
        String registeredModelName = args[4];

        // Start Logging
        Mlflow.startRun();

        // enable autologging
        Mlflow.sklearn.autolog();

        // Prepare the data
        System.out.println("input data: " + data);
        List<String> lines = null;
        try {
            lines = Files.readAllLines(Paths.get(data));
        } catch (IOException e) {
            e.printStackTrace();
        }
        String csvData = String.join("\n", lines);
        String[] csvLines = csvData.split("\n");
        String[] header = csvLines[0].split(",");
        String[] dataLines = new String[csvLines.length - 1];
```

Code Documentation

The screenshot shows a user interface for code documentation. At the top, there are navigation links: "/document" and "/docs". On the right side, there is a "You" section with a profile picture and a "Logout" button. Below the navigation, a search bar contains the text "/document main.py".

The main content area features a profile icon for "watsonx" and the name "watsonx".

The code snippet is as follows:

```
# Assisted by watsonx Code Assistant
def main():
    """Main function of the script.

Args:
    --data (str): path to input data
    --test_train_ratio (float): test/train split ratio (default: 0.25)
    --n_estimators (int): number of estimators (default: 100)
    --learning_rate (float): learning rate (default: 0.1)
    --registered_model_name (str): model name

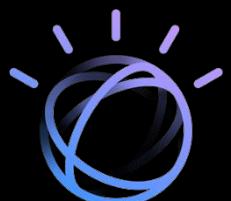
Returns:
    None
    """
```


At the bottom of the content area, there are two small icons: a thumbs up and a thumbs down.

Closing Thoughts

Industry Principles for Trust and Transparency

<https://www.ibm.com/downloads/cas/J45XZOAR>



1.The purpose of AI is to augment
human intelligence

2.Data and insights belong to
their creator

3.AI systems must be transparent
and explainable

Takeaways AI lecture

Citizen AI:

Develop critical thinking skills, have your voice heard

<https://www.accenture.com/cz-en/insight-explainable-citizen-ai>

Big Data Advantage:

When giving away your personal data, choose wisely

<https://www.slideshare.net/RobertoVII/ai-and-big-data-for-business-and-people-advantage>

Intelligent Automation:

Educate yourself on AI, thrive in your profession

https://1.dam.s81c.com/m/3de136737e51fb20/original/IBM-Automation-Whitepaper_Final.pdf

STAT-5350/4350

Applied Deep Learning

Jesus.Portilla@colorado.edu

Q&A

Ivan Portilla

Portilla@gmail.com

