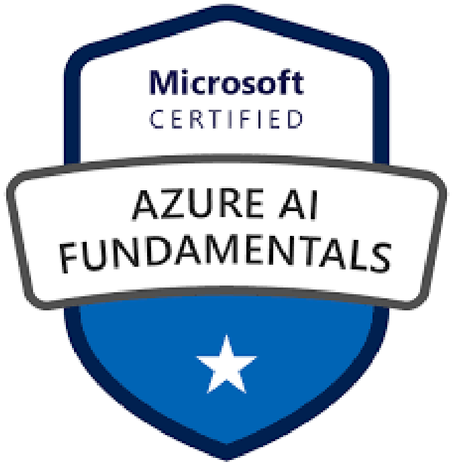


Microsoft Azure AI Fundamentals AI-900

Presented by Johannes Hayer

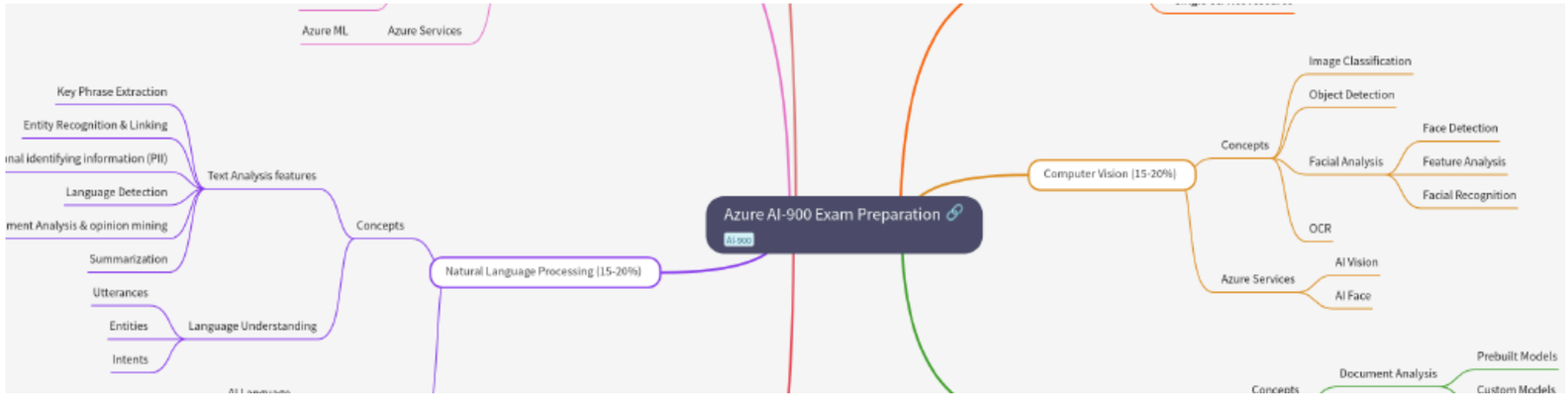
Disclaimer

- These slides are free to use and share for personal and educational purposes.
- Commercial use is not permitted.
- For maximal learning success
  - check out the complete course on Udemy - available with a huge discount to make it accessible for everyone!
  - use free prep-exam questions on quizlab.cloud to test your knowledge and get familiar with the exam style and question formats.
- Thank you for respecting these terms! Wishing you success on your exam journey.

AI-900 MindMap

- **Get a clear overview over all Topics & Sub-topics you need to know for the exam!**

- **Click here to claim your free AI-900 MindMap**



About me

- I'm Johannes!
- 3x AWS Certified and Ix Azure Certified (AI 900)
- B. Sc. in Informatics & Working as Solutions Architect for years

- You can find me here:
  - Linkedin: https://www.linkedin.com/in/johannes-hayer-b8253a294/
  - Medium: https://medium.com/@muellerjohannes93
  - Blog: https://jhayer.tech

Overview

01

- Machine Learning

02

- Computer Vision

03

- Natural Language Processing (NLP)

04

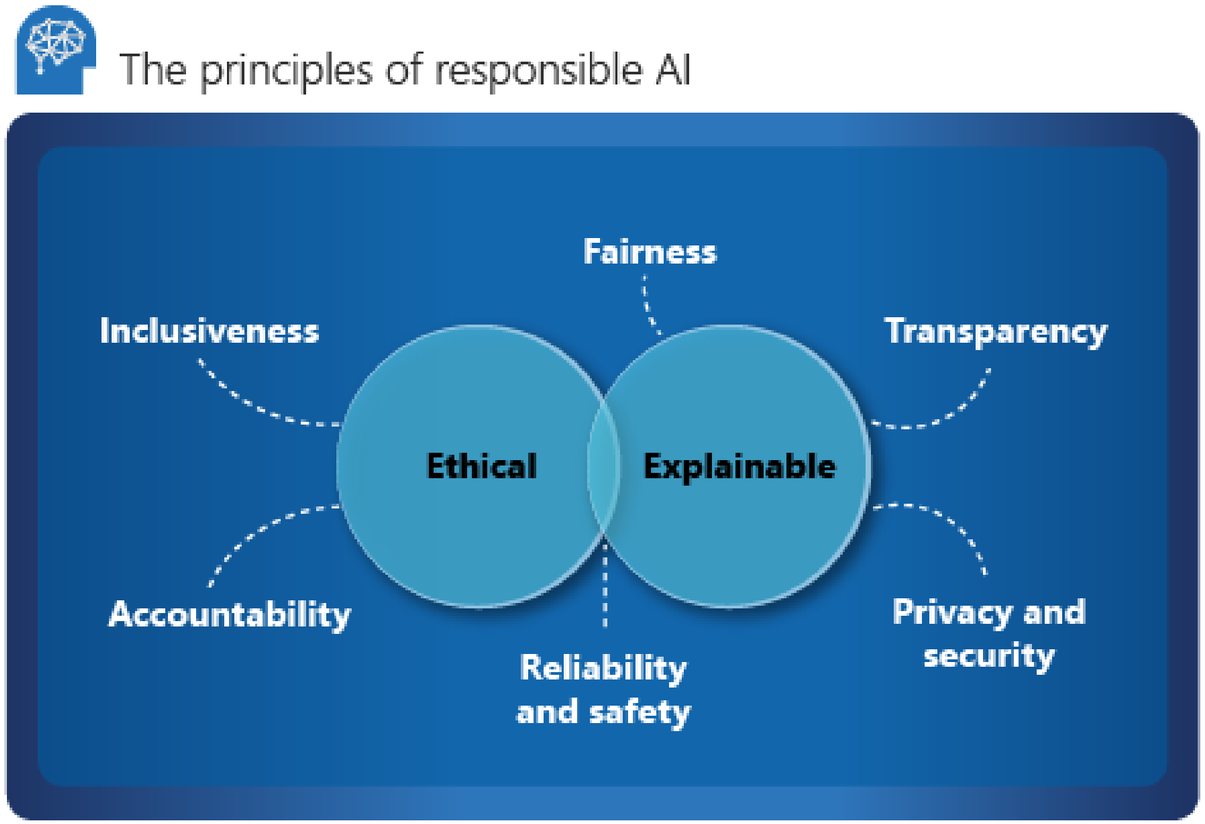
- Document Intelligence & Knowledge Mininng

05

- Generative AI

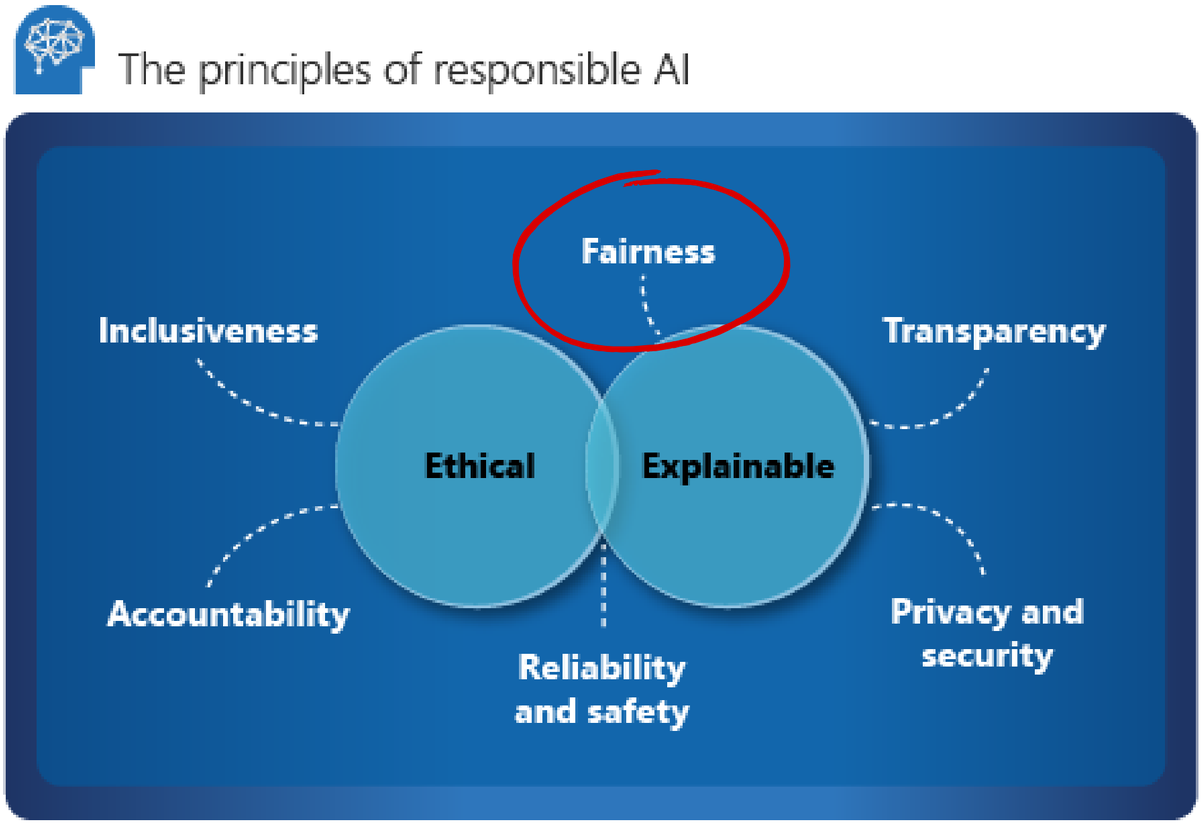
Responsible AI

- At Microsoft, AI software development is guided by a set of six principles, designed to ensure that AI applications provide amazing solutions to difficult problems without any unintended negative consequences.



Fairness

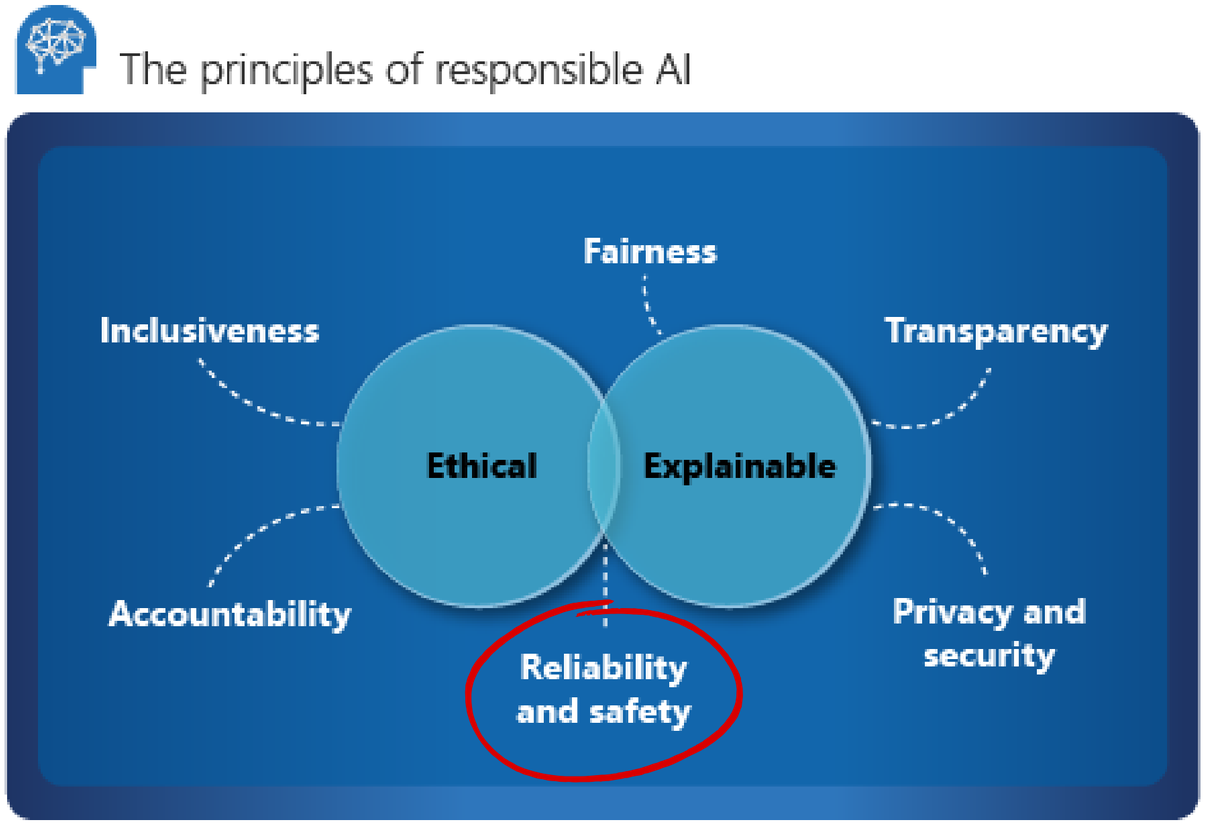
- AI systems must treat all people equally, avoiding bias based on gender, ethnicity, or other factors.



Example: A loan approval model should predict outcomes without bias. For instance, the model should not disproportionately deny loans to applicants from a particular ethnic group.

Reliability and Safety

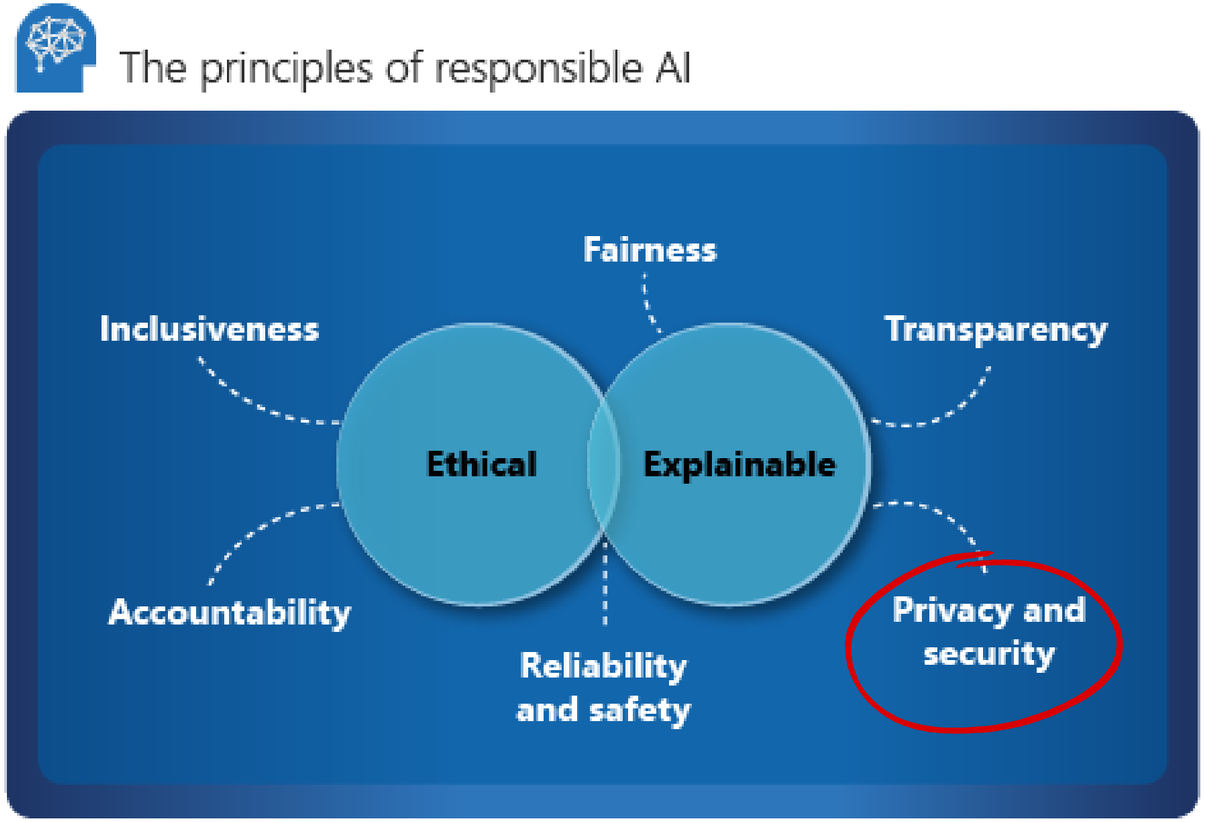
- AI systems should perform reliably and safely, especially in critical applications like autonomous vehicles or medical diagnosis.
- Development: Rigorous testing and deployment management are essential to ensure AI systems function as expected.



Example: An AI-based software system for an autonomous vehicle must reliably detect and respond to obstacles to avoid accidents.

Privacy and Security

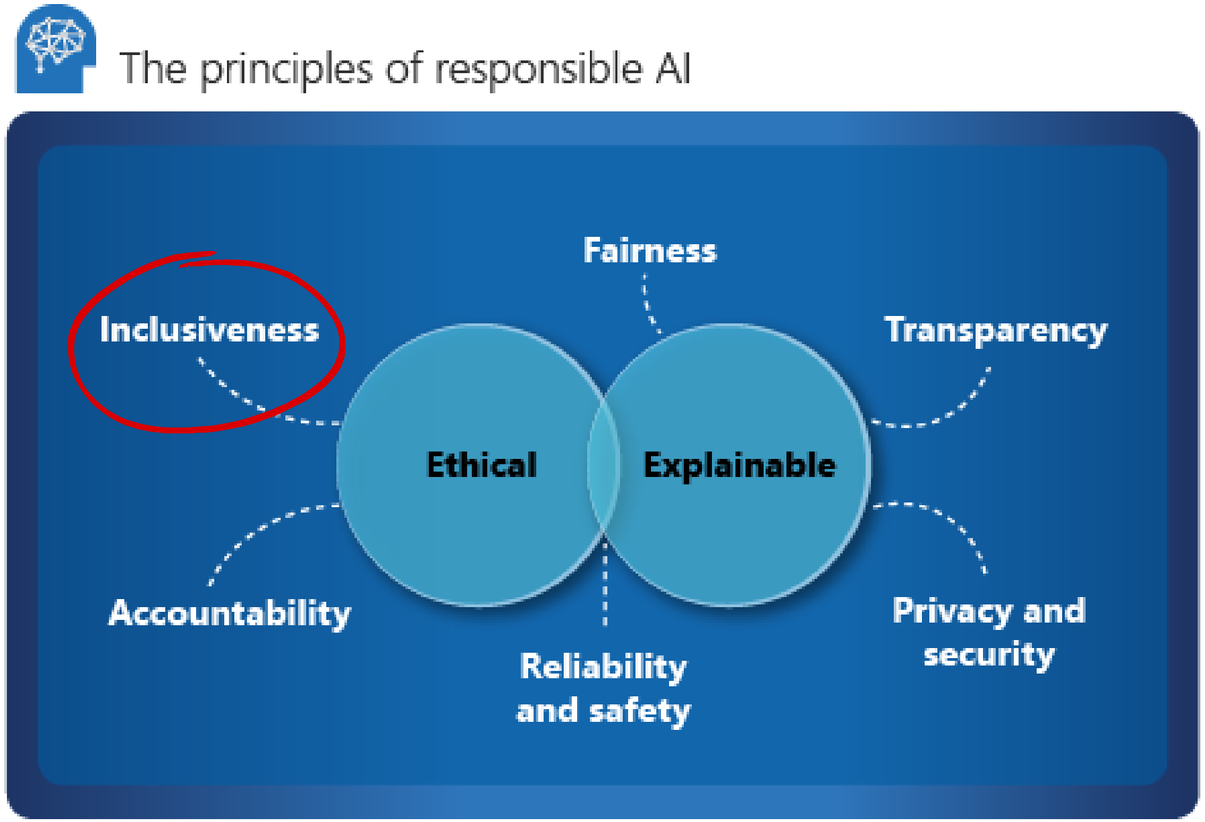
- AI systems must be secure and respect user privacy, especially when handling personal data.
- Consideration: Privacy and security concerns must be addressed during training and production phases.



Example: A healthcare AI system must ensure that patient data used to train models and make predictions is anonymized and secure from unauthorized access.

Inclusiveness

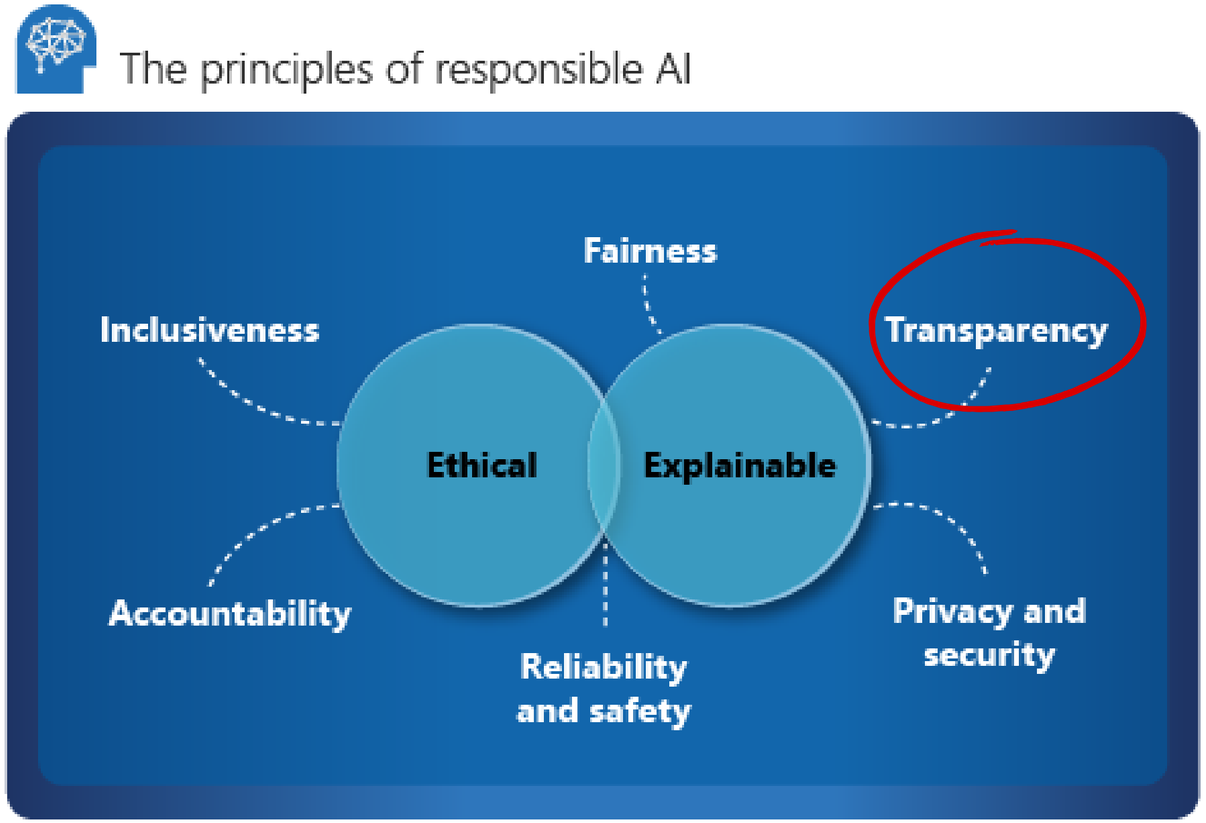
- AI should benefit all of society, regardless of physical ability, gender, sexual orientation, ethnicity, or other factors.



Example: A voice recognition system should accurately understand speakers with different accents and speech patterns.

Transparency

- AI systems should be understandable, with clear communication about their purpose, functionality, and limitations.

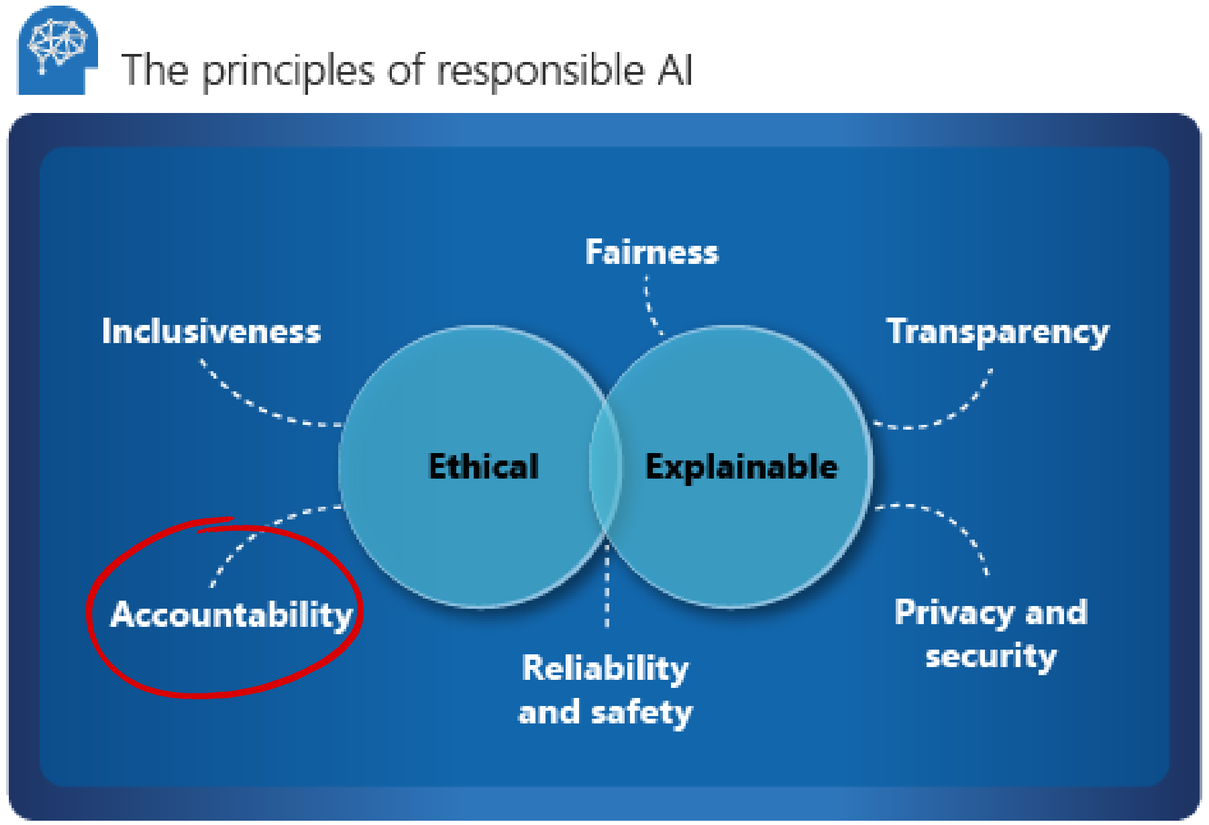


Example: A recommendation system for online shopping should explain why certain products are suggested to users.

Accountability

- Accountability is an essential pillar of responsible AI. It means people and organizations must take responsibility for the AI systems they create and deploy.

- Having clear frameworks and processes to ensure AI systems meet ethical and legal standards throughout their lifecycle.



Example: In a financial institution, an AI system used for fraud detection should have accountability measures in place, including:

- Clear documentation of who designed and approved the system
- Transparent reporting of the system's performance and any incidents
- A clear chain of responsibility from developers to healthcare providers

Machine Learning

01 Machine Learning

02 Computer Vision

03 Natural Language Processing

04 Document Intelligence & Knowledge Mininng

05 Generative AI

What is Machine Learning?

- In math, a function like f(x)=y takes an input x and produces an output y. Every time you input x, you get the same output.

- Instead of defining the function explicitly, ML algorithms learn the function from data. This process of defining the function is called training

- After the training phase you can use the model (software program that encapsulates the function) to predict new values. This part is called inferencing

Vocab's used in ML

- features = observered attribute(s)
  the data that you know: size of the house, location,..

- labels = outcome that the model is trying to predict
  the data that you dont know: price of the house

Size (sq ft)	Bedrooms	Bathrooms	Location	Age (years)	Proximity to Schools	Public Transport	Price
2000	3	2	City Center	10	Near	Yes	$300,000
1500	2	1	Suburb	5	Far	No	$200,000
2500	4	3	Downtown	15	Moderate	Yes	$400,000

Types of Machine Learning

Supervised machine learning

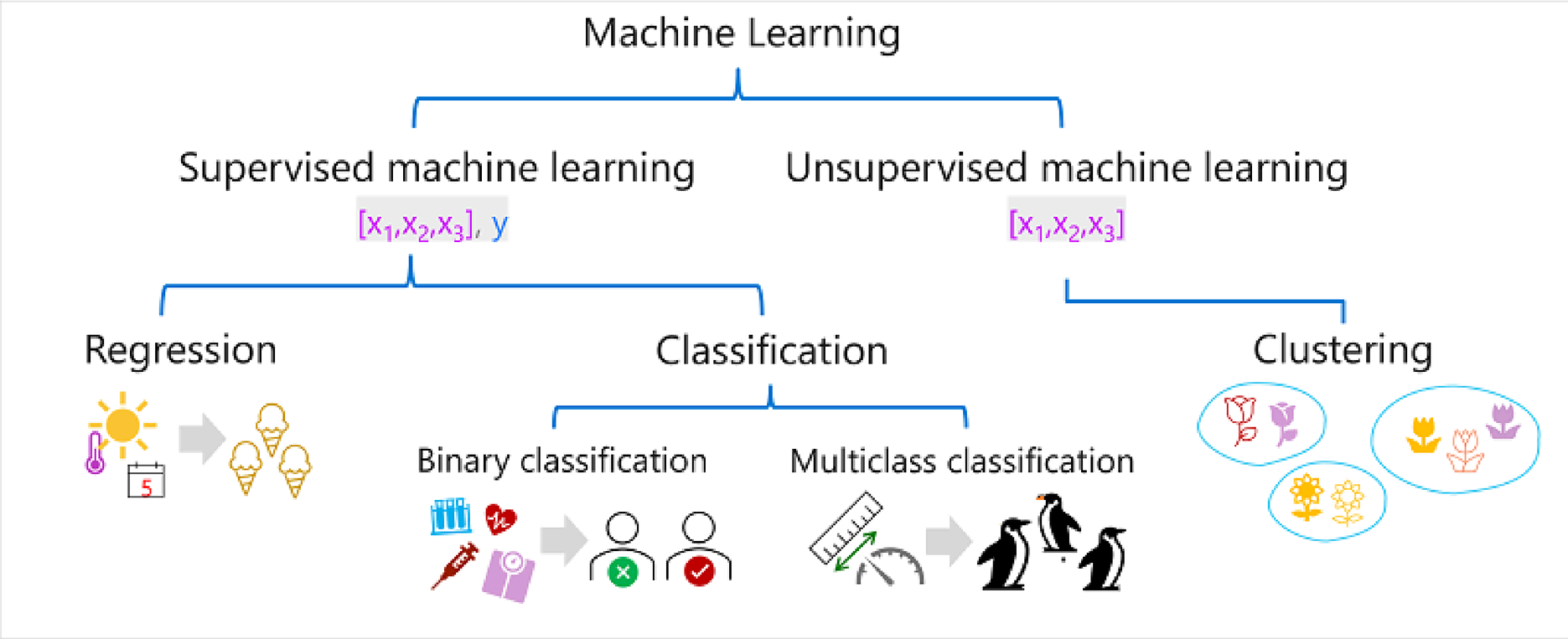
- machine learning algorithms in which the training data includes both feature values and known label values.
  [X1,X2,X3], Y

- is used to train model by determining a relationship between the features and labels in past observations

- every time we have feature(s) and label data to train a model it's a supervised type

eg: we want to predict the house price based on house location, size,.. and we have all the data like location, size and price!

Supervised machine learning



Unsupervised machine learning

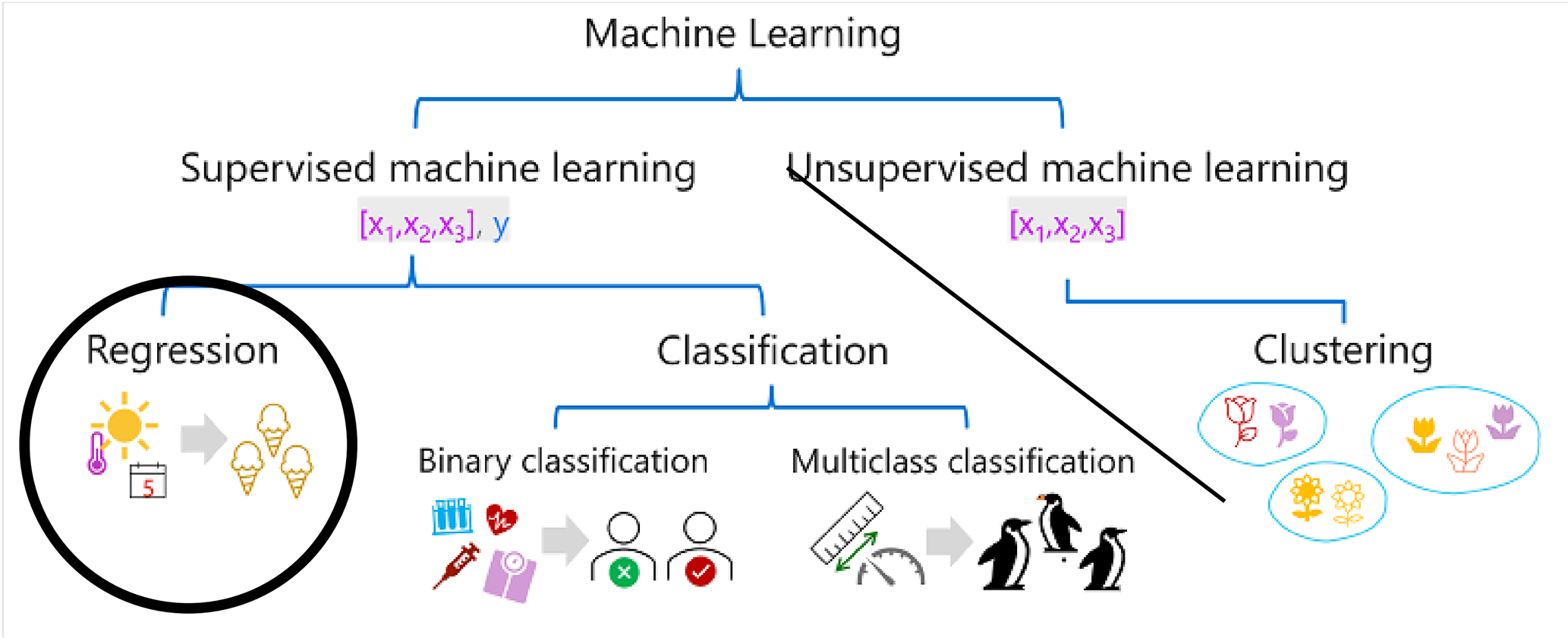
- machine learning algorithms in which the training data includes ONLY feature values
  [X1,X2,X3]

- Unsupervised machine learning algorithms determine relationships between the features of the observations in the training data.

- every time we have only the feature(s) data to train a model it's a unsupervised type

eg: Identify groups of similar customers based on demographic attributes and purchasing behavior.

Unsupervised machine learning



Regression

- Regression models are trained to predict numeric label values based on training data that includes both features and known labels.
  - The number of ice creams sold on a given day, based on the temperature, rainfall, and windspeed.
  - The selling price of a property based on its size in square feet, the number of bedrooms it contains, and socio-economic metrics for its location.
  - The fuel efficiency (in miles-per-gallon) of a car based on its engine size, weight, width, height, and length.

Regression – Training Process

Training– Process

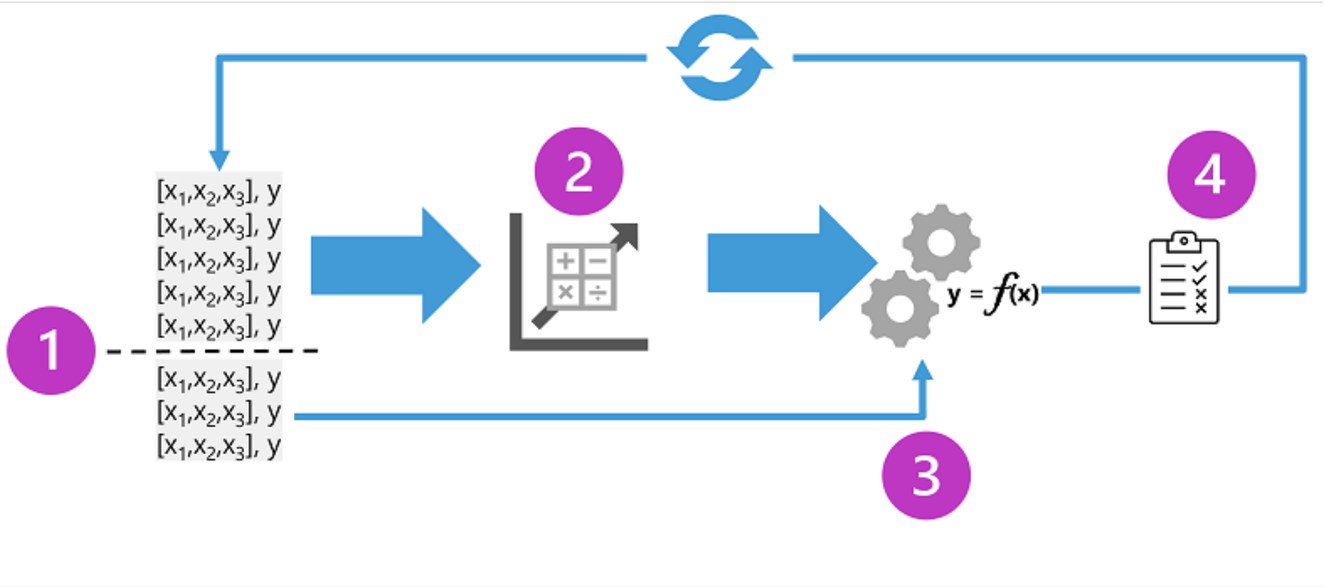
1. Split data into training and validation sets

2. Take the training data and put it into the algorithm (this case linear regression)

3. Use the validation data to test the model by predicting labels for the features.

4. Compare known actual labels (in the validation set) with predicted values. Calculate common metrics to evaluate model

= Iteration



Regression evaluation metrics

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Coefficient of determination (R2)

Mean Absolute Error (MAE)

- Measures the average magnitude of prediction errors (doesn't matter prediction was over or under the actual value)
- for each prediction error is called absolute error
- can be summarised for the whole validation set as mean absolute error (MAE)

Actual Price ($)	Predicted Price ($)	Absolute Error ($)
200,000	210,000	10,000
250,000	240,000	10,000
300,000	320,000	20,000
280,000	260,000	20,000
350,000	370,000	20,000

Mean Absolute Error (MAE): $16,000

To calculate the Mean Absolute Error, we sum up all the absolute errors and divide by the number of predictions
MAE = (10,000 + 10,000 + 20,000 + 20,000 + 20,000) = 80,000 / 5 = 16,000

Mean Squared Error (MSE)

- takes all discrepancies between predicted and actual labels into account equally
- "amplifies" larger errors by squaring the individual errors

Actual Price ($)	Predicted Price ($)	Error ($)	Squared Error
200,000	210,000	10,000	100,000,000
250,000	240,000	-10,000	100,000,000
300,000	320,000	20,000	400,000,000
280,000	260,000	-20,000	400,000,000
350,000	370,000	20,000	400,000,000

Mean Absolute Error (MAE): $16,000

Mean Squared Error (MSE): $280,000,000

We square these errors: 100,000,000, 100,000,000, 400,000,000, 400,000,000, and 400,000,000.

We calculate the mean of these squared errors:
MSE = (100,000,000 + 100,000,000 + 400,000,000 + 400,000,000 + 400,000,000) / 5 = 280,000,000

Root Mean Squared Error

- If we want to measure the error in terms of the house prices, we need to calculate the square root of the MSE; which produces a metric called, unsurprisingly, Root Mean Squared Error (RMSE)

Actual Price ($)	Predicted Price ($)	Error ($)	Squared Error
200,000	210,000	10,000	100,000,000
250,000	240,000	-10,000	100,000,000
300,000	320,000	20,000	400,000,000
280,000	260,000	-20,000	400,000,000
350,000	370,000	20,000	400,000,000

Mean Absolute Error (MAE): $16,000
Mean Squared Error (MSE): $280,000,000
Root Mean Squared Error (RMSE): $16,733.201

The MSE of $280,000,000 is a large number that's difficult to interpret in terms of house prices.

It doesn't directly tell us how far off our predictions are in dollars. This is where RMSE comes in:
RMSE = √MSE = √280,000,000 ≈ $16,733

The RMSE of $16,733 is much more interpretable. It tells us that, on average, our model's predictions deviate from the actual house prices by about $16,733.

This is in the same unit (dollars) as our original data, making it easier to understand in the context of house prices.

Coefficient of determination (R2)

- R^2 is a measure of how well our prediction model fits the data
- It's expressed as a value between 0 and 1 (or 0% to 100%)
- The closer to 1 (or 100%), the better our model predicts

Actual Price ($)	Predicted Price ($)	(Actual - Predicted)^2	(Actual - Mean Actual)^2
200,000	210,000	100,000,000	5,776,000,000
250,000	240,000	100,000,000	676,000,000
300,000	320,000	400,000,000	576,000,000
280,000	260,000	400,000,000	16,000,000
350,000	370,000	400,000,000	5,476,000,000

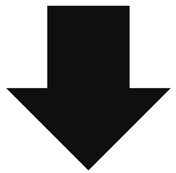
Mean Actual Price: $276,000
Sum of Squared Residuals (SSRes): 1,400,000,000
Total Sum of Squares (SSTot): 12,520,000,000
R-squared (R^2): 0.8882

- Calculate the mean of actual house prices: Mean Actual Price = $276,000
- Calculate the Sum of Squared Residuals (SSRes): This is the sum of (Actual Price - Predicted Price)^2 for each house. SSRes = 1,800,000,000
- Calculate the Total Sum of Squares (SSTot): This is the sum of (Actual Price - Mean Actual Price)^2 for each house. SSTot = 17,040,000,000
- Calculate R^2: $R^2 = 1 - (SSRes / SSTot)$ $R^2 = 1 - (1,800,000,000 / 17,040,000,000)$ $R^2 \approx 0.89$

Iterative training

What can we do to improve our training process?

- Feature selection and preparation (choosing which features to include in the model)
- Algorithm selection (We explored linear regression in the previous example, but there are many other regression algorithms)
- Algorithm parameters (numeric settings to control algorithm behavior, more accurately called hyperparameters to differentiate them from the x and y parameters).



After multiple iterations, the model that results in the best evaluation metric that's acceptable for the specific scenario is selected.

Classification

- Instead of predicting numeric values like a regression model, classification models predict probability values

Two main types:

1. Binary Classification
2. Multiclass Classification

- Binary Classification:
  - Predicts between two options (1 and 0) or (yes or no)
  - Example: Spam or not spam

- Multiclass Classification:
  - Predicts among three or more options
  - Example: Dog breed identification

Binary classification

- Is a supervised machine learning technique

- Binary classification algorithms are used to train a model that predicts one of two possible labels for a single class. Essentially, predicting true or false.

- Follows the same iterative process of training, validating and evaluating models.

Binary classification – Training

example data

Blood glucose (x)	Diabetic? (y)
67	0
103	1
114	1
72	0
116	1
65	0

Training– Process

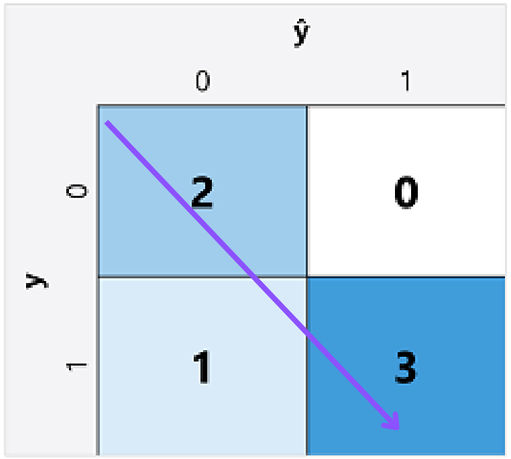
1. Split data into training and validation sets
2. Take the training data and put it into the algorithm (this case logistic regression)
   Despite its name, in machine learning logistic regression is used for classification, not regression.
3. Use the validation data to test the model by predicting labels for the features.
4. Compare known actual labels (in the validation set) with predicted values. Calculate common metrics to evaluate model

= Iteration

Same as Regression but with other algorithms and metrics!

Binary classification Evaluation Metrics

The first step in calculating evaluation metrics for a binary classification model is usually to create a matrix of the number of correct and incorrect predictions for each possible class label ---> confusion matrix



- ŷ=0 and y=0: True negatives (TN) ✓ This is when the model correctly predicts no diabetes (0) for a patient who actually doesn't have diabetes (0).
- ŷ=1 and y=0: False positives (FP) ✗ This is when the model incorrectly predicts diabetes (1) for a patient who actually doesn't have diabetes (0).
- ŷ=0 and y=1: False negatives (FN) ✗ This is when the model incorrectly predicts no diabetes (0) for a patient who actually has diabetes (1).
- ŷ=1 and y=1: True positives (TP) ✓ This is when the model correctly predicts diabetes (1) for a patient who actually has diabetes (1).

- true predictions are shown in a diagonal line from top-left to bottom-right
- Color-intensity indicates number of predictions in each cell

(model that predicts well should reveal a deeply shaded diagonal trend)

Evaluation Metrics – Accuracy

The simplest metric you can calculate from the confusion matrix is accuracy - the proportion of predictions that the model got right.

- $\hat{y}$=0 and y=0: True negatives (TN)
- $\hat{y}$=1 and y=0: False positives (FP)
- $\hat{y}$=0 and y=1: False negatives (FN)
- $\hat{y}$=1 and y=1: True positives (TP)

- Accuracy is calculated as:

(TN+TP) ÷ (TN+FN+FP+TP)

- In the case of our diabetes example, the calculation is:

(2+3) ÷ (2+1+0+3)

= 5 ÷ 6

= 0.83

Our diabetes classification model produced correct predictions 83% of the time.

Evaluation Metrics – Recall

Recall is a metric that measures the proportion of positive cases that the model identified correctly

- $\hat{y}$=0 and y=0: True negatives (TN)
- $\hat{y}$=1 and y=0: False positives (FP)
- $\hat{y}$=0 and y=1: False negatives (FN)
- $\hat{y}$=1 and y=1: True positives (TP)

- Recall is calculated as:
  TP ÷ (TP+FN)

- In the case of our diabetes example, the calculation is:

3 ÷ (3+1) = 3 ÷ 4

= 0.75

Our diabetes classification model identified 75% of patients who have diabetes as having diabetes.

Evaluation Metrics – Precision

Precision is a similar metric to recall, but measures the proportion of predicted positive cases where the true label is actually positive

- $\hat{y}$=0 and y=0: True negatives (TN)
- $\hat{y}$=1 and y=0: False positives (FP)
- $\hat{y}$=0 and y=1: False negatives (FN)
- $\hat{y}$=1 and y=1: True positives (TP)

- Precision is calculated as:
  TP ÷ (TP+FP)

- In the case of our diabetes example, the calculation is:

3 ÷ (3+0) = 3 ÷ 3

= 1.0

So 100% of the patients predicted by our model to have diabetes do in fact have diabetes.

Recall = quantity
Precision = quality

Evaluation Metrics F1-score

F1-score is an overall metric that combined recall and precision

- F1 is calculated as:
  (2 x Precision x Recall) ÷ (Precision + Recall)

- In the case of our diabetes example, the calculation is:

  (2 x 1.0 x 0.75) ÷ (1.0 + 0.75)

  = 1.5 ÷ 1.75

  = 0.86

- High F1 score means your model is doing well at both finding all positive cases and being accurate in its positive predictions.

Multiclass classification

- Is a supervised machine learning technique
- Multiclass classification is used to predict to which of multiple possible classes an observation belongs
- Follows the same iterative process of training, validating and evaluating models.

Multiclass classification – Training

example data

Flipper length (x)	Species (y)
167	0
172	0
225	2
197	1
189	1
232	2
158	0

Training– Process

1. Split data into training and validation sets
2. Take the training data and put it into the algorithm (OvR or Multinomial algorithms)
3. Use the validation data to test the model by predicting labels for the features.
4. Compare known actual labels (in the validation set) with predicted values. Calculate common metrics to evaluate model

= Iteration

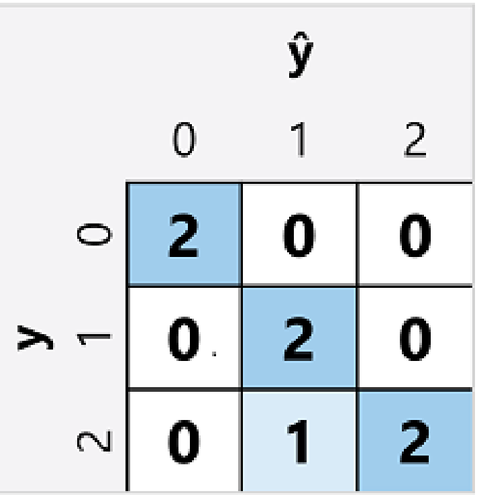
Same as Regression, Binary Classification but with other algorithms !

Multiclass classification – Evaluating

- You can evaluate a multiclass classifier by calculating binary classification metrics for each individual class.

Flipper length (x)	Actual species (y)	Predicted species (ŷ)
165	0	0
171	0	0
205	2	1
195	1	1
183	1	1
221	2	2
214	2	2

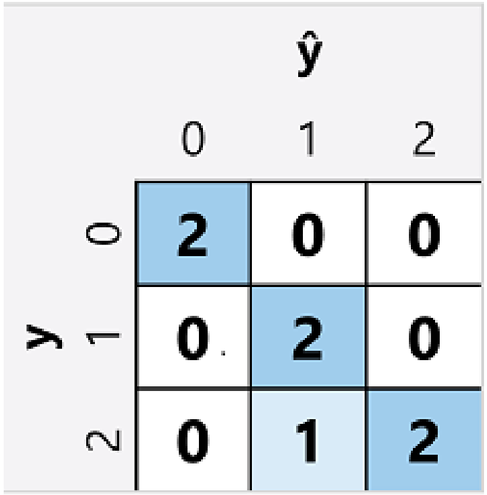
- Class 0: Adelie (short flippers)
- Class 1: Chinstrap (medium flippers)
- Class 2: Gentoo (long flippers)



confusion matrix for a multiclass classifier is similar to that of a binary classifier, except that it shows the number of predictions for each combination of predicted (ŷ) and actual class labels (y):

Multiclass classification – Evaluating

- From this confusion matrix, we can determine the metrics for each individual class as follows



Class	TP	TN	FP	FN	Accuracy	Recall	Precision	F1-Score
0	2	5	0	0	1.0	1.0	1.0	1.0
1	2	4	1	0	0.86	1.0	0.67	0.8
2	2	4	0	1	0.86	0.67	1.0	0.8

To calculate the overall accuracy, recall, and precision metrics, you use the total of the TP, TN, FP, and FN metrics:

- Overall accuracy = (13+6)÷(13+6+1+1) = 0.90
- Overall recall = 6÷(6+1) = 0.86
- Overall precision = 6÷(6+1) = 0.86
- Overall F1-score = (2x0.86x0.86)÷(0.86+0.86) = 0.86

Clustering

- Clustering is a form of unsupervised machine learning in which observations are grouped into clusters based on similarities in their data values, or features

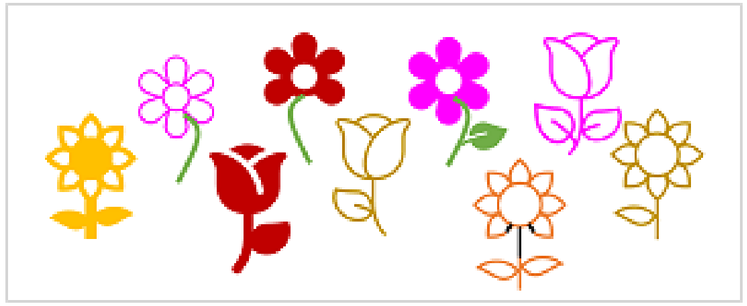
Examples

- Anomaly detection: Unsupervised clustering can process large datasets and discover data points that are atypical in a dataset.
- Recommendation engines: Using association rules, unsupervised machine learning can help explore transactional data to discover patterns or trends that can be used to drive personalized recommendations for online retailers.
- Natural language processing (NLP): Unsupervised learning is commonly used for various NLP applications, such as categorizing articles in news sections, text translation and classification, or speech recognition in conversational interfaces.

Clustering Example

- Suppose a botanist observes a sample of flowers and records the number of leaves and petals on each flower:

- There are no known labels in the dataset, just two features. The goal is not to identify the different types (species) of flower; just to group similar flowers together based on the number of leaves and petals.

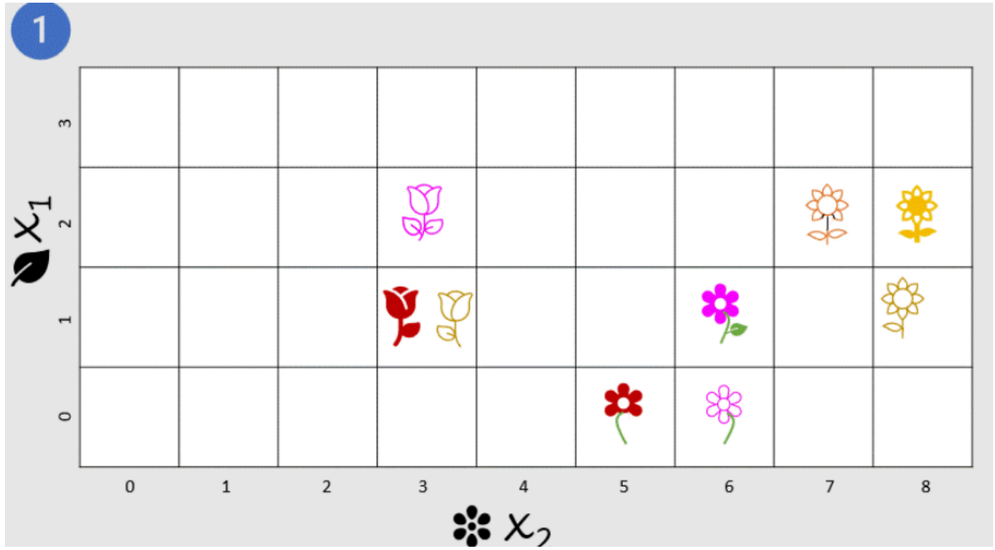


Leaves (x1)	Petals (x2)
0	5
0	6
1	3
1	3
1	6
1	8
2	3
2	7
2	8

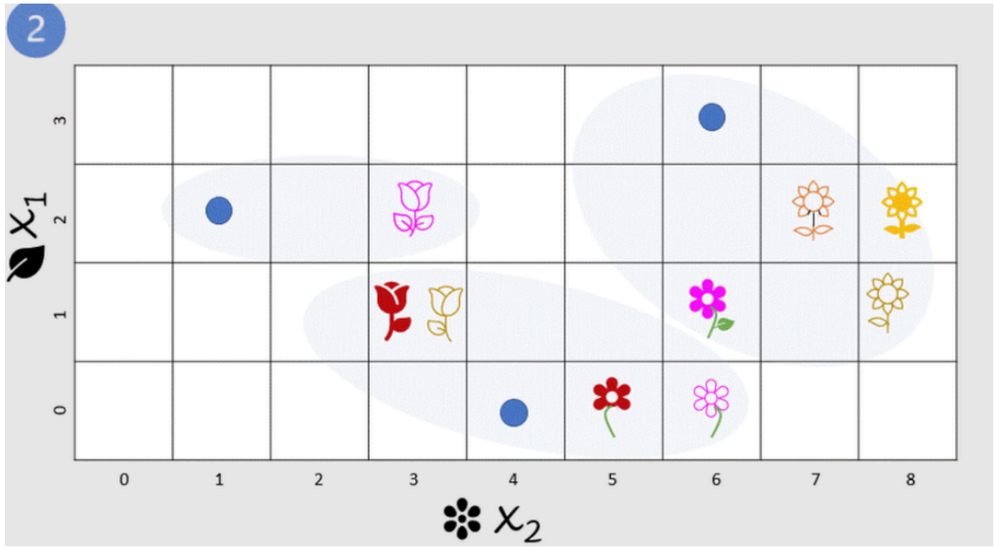
Clustering – Training

Training– Process

1. The feature (x) values are vectorized to define n-dimensional coordinates (where n is the number of features)



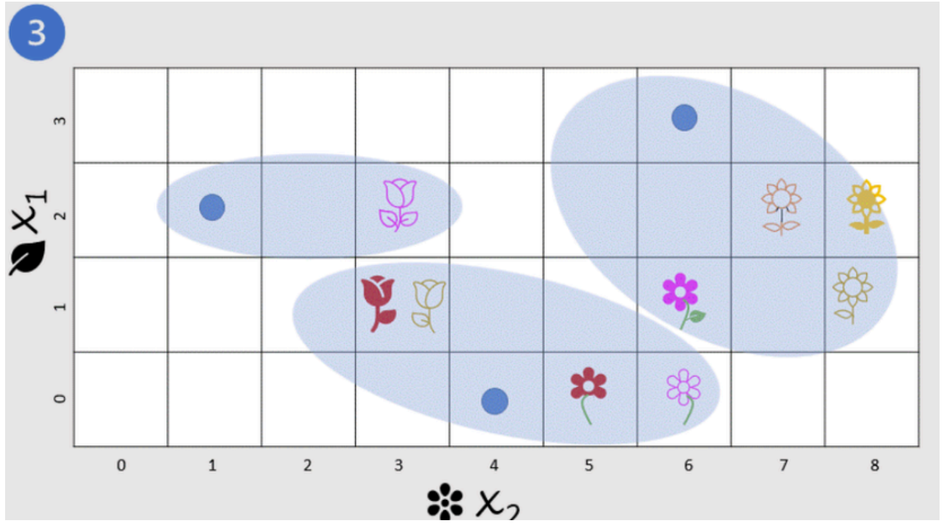
2. You decide how many clusters you want to use to group the flowers - call this value k. For example, to create three clusters, you would use a k value of 3. Then k points are plotted at random coordinates. These points become the center points for each cluster, so they're called centroids.



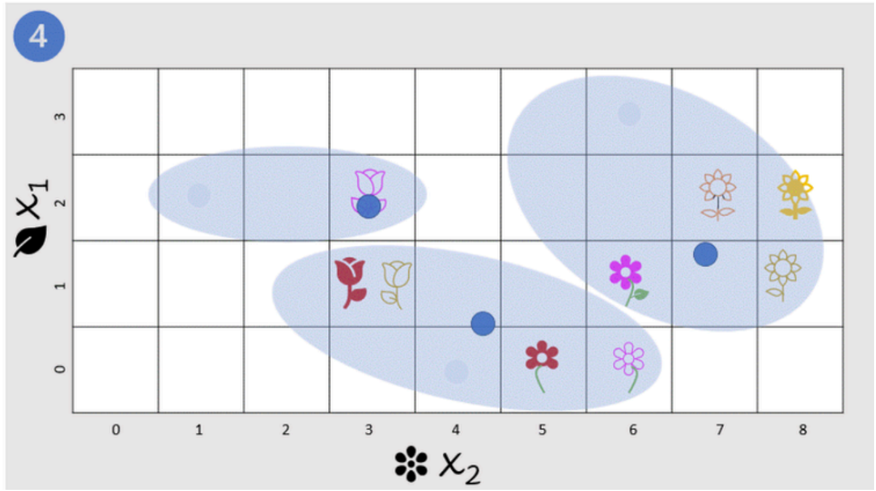
Clustering – Training

Training– Process

3. Each data point (in this case a flower) is assigned to its nearest centroid.



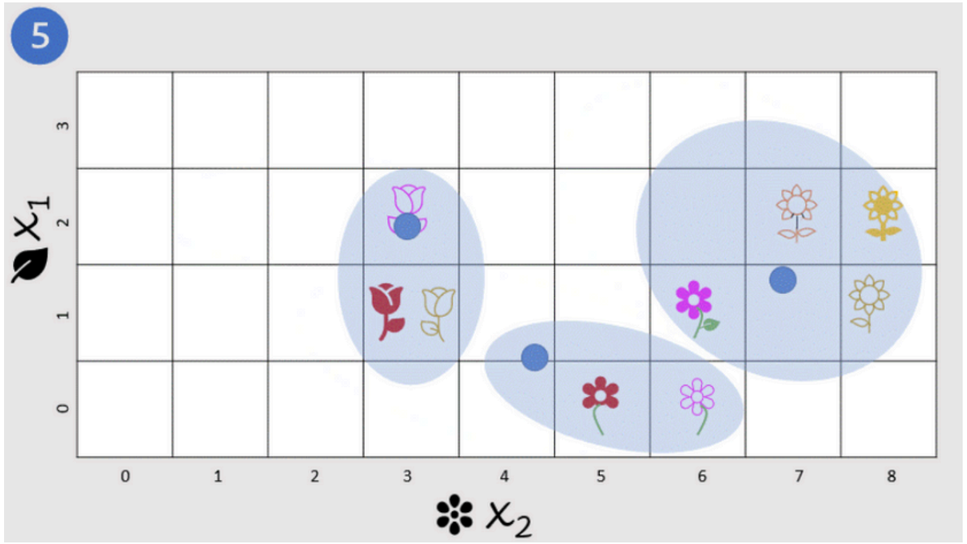
4. Each centroid is moved to the center of the data points assigned to it based on the mean distance between the points.



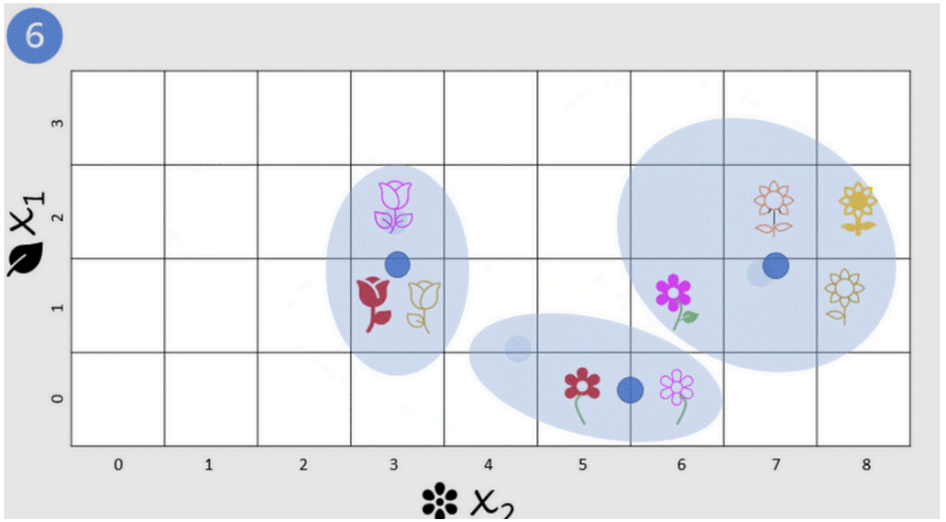
Clustering – Training

Training– Process

5. After the centroid is moved, the data points may now be closer to a different centroid, so the data points are reassigned to clusters based on the new closest centroid.



6. The centroid movement and cluster reallocation steps are repeated until the clusters become stable or a predetermined maximum number of iterations is reached.



Clustering – Evaluating

- Average distance to cluster center: How close, on average, each point in the cluster is to the centroid of the cluster.

- Average distance to other center: How close, on average, each point in the cluster is to the centroid of all other clusters.

- Maximum distance to cluster center: The furthest distance between a point in the cluster and its centroid.

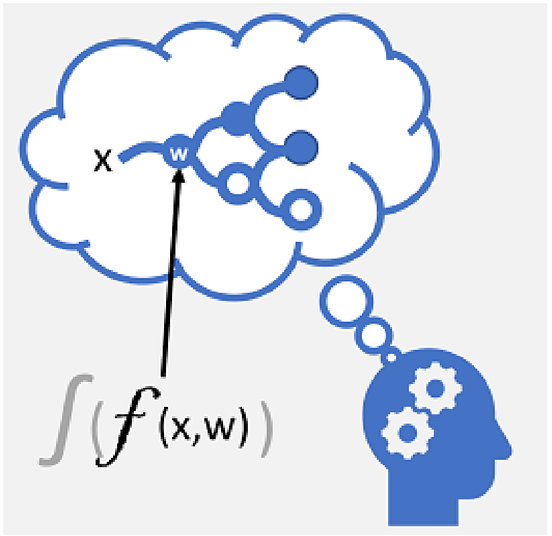
- Silhouette: A value between -I and I that summarizes the ratio of distance between points in the same cluster and points in different clusters (The closer to I, the better the cluster separation).

What is Deep Learning?

- Deep learning is an advanced form of machine learning that tries to emulate the way the human brain learns.

- Uses artificial neural networks that simulates electrochemical activity in biological neurons by using mathematical functions

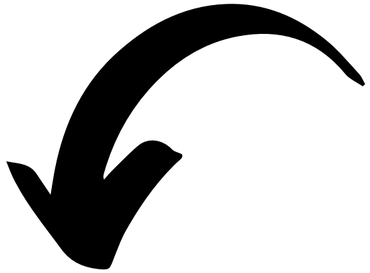
Biological neural network



Artificial neural network

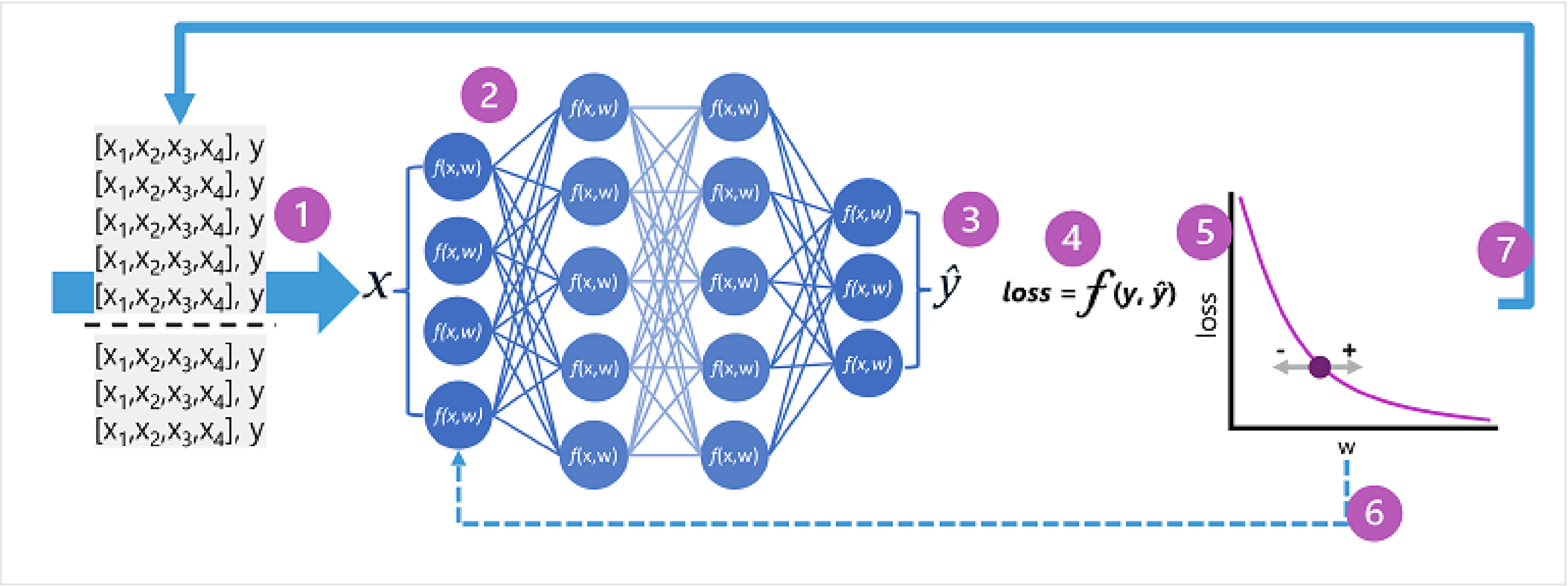
The “Deep”

- We have many layers - hundreds or even thousands!
- Each layer processes the information and passes it to the next



- Creates a “deep” network = deep neural networks (DNNs)

How Does It Learn?



How Does It Learn?

I. Data Preparation
Split data into training and validation sets • Feed training features into the input layer

2.Forward Pass
Apply weights to inputs (initially random) • Pass data through network layers

3.Prediction
Output layer produces probabilities
Example: [0.3, 0.1, 0.6] for penguin species

4.Loss Calculation
Compare predictions to actual values
Aggregate differences into single 'loss' value

5. Optimization
Use calculus to determine weight adjustments,Goal: Reduce overall loss

6.Backpropagation
Update weights throughout the network • Start from output layer, move backwards

7.Iteration
Repeat process over multiple 'epochs'
Continue until loss is minimized and predictions are accurate

Deep Learning vs Supervised

- Deep Learning:
  - Uses many layers of artificial neurons
  - Can learn complex patterns on its own
  - Good for tasks like image recognition or language processing
  - Needs lots of data and computing power
  - Can be harder to understand how it makes decisions

- Supervised Machine Learning:
  - Uses simpler models with fewer layers
  - Needs humans to pick important features
  - Good for structured data with clear rules
  - Works well with less data
  - Easier to understand and explain its decisions

Azure Machine Learning

- is a cloud service for training, deploying, and managing machine learning models
    - Exploring data and preparing it for modeling.
    - Training and evaluating machine learning models.
    - Registering and managing trained models.
    - Deploying trained models for use by applications and services.
    - Reviewing and applying responsible AI principles and practices.

- Features and capabilities of Azure Machine Learning
    - Centralized storage and management of datasets for model training and evaluation.
    - On-demand compute resources on which you can run machine learning jobs, such as training a model.
    - Automated machine learning (AutoML), which makes it easy to run multiple training jobs with different algorithms and parameters to find the best model for your data.
    - Visual tools to define orchestrated pipelines for processes such as model training or inferencing.

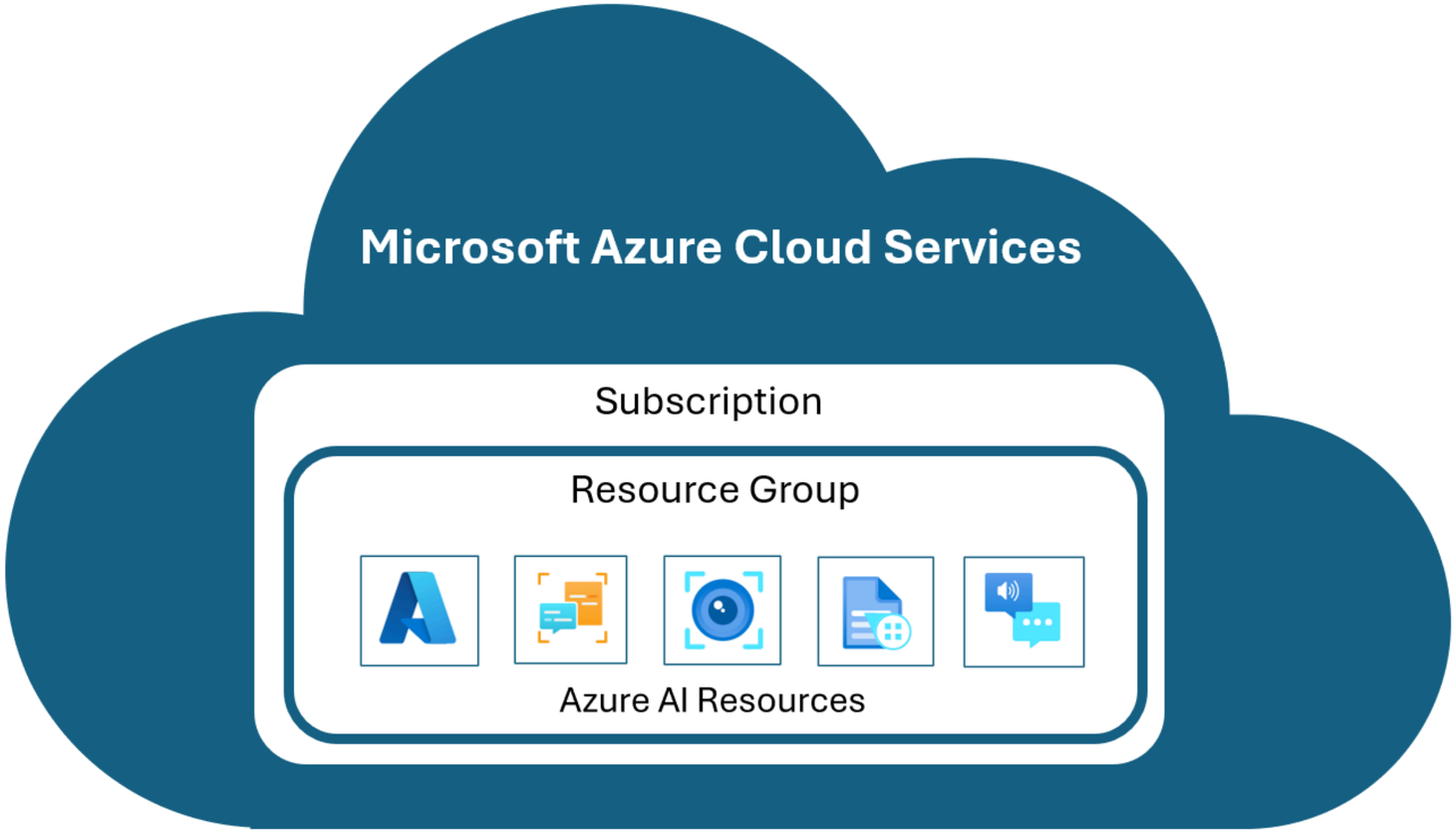
What are Azure AI Services?

- Azure AI services are pre-built AI capabilities that can be easily integrated into web or mobile applications. These services cover a wide range of AI functionalities, including:
  - Image recognition
  - Natural language processing
  - Speech-to-text and text-to-speech
  - AI-powered search
  - And many more

Three principles of AI Services

- Azure AI services are based on three principles that dramatically improve speed-to-market:
  - Prebuilt and ready to use
  - Acced through APIs
  - Available on Azure

Create Azure AI service resources



Create Azure AI service resources

- There are two types of AI service resources: multi-service or single-service.

- Multi-service resource

a resource created in the Azure portal that provides access to multiple Azure AI services with a single key and endpoint. Use the resource Azure AI services when you need several AI services or are exploring AI capabilities.

all AI services are billed together

- Single-service resources

a resource created in the Azure portal that provides access to a single Azure AI service, such as Speech, Vision, Language, etc. Each Azure AI service has a unique key and endpoint.

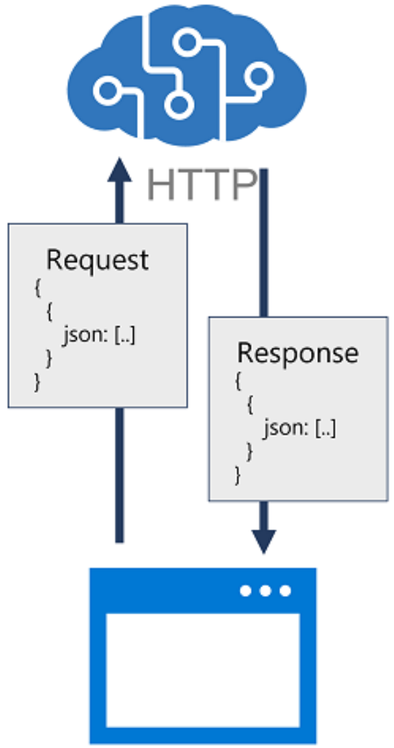
all AI services are billed separately

Use Azure AI services

- We can use Azure AI Services with:
  - REST API
  - Software Development Kit (SDK)
  - Visual Studio interfaces

Use Azure AI services

- We can use Azure AI Services with:
  - REST API
  - In most cases, service functions can be called by submitting data in JSON format over an HTTP request, which may be a POST, PUT, or GET request depending on the specific function being called




```
curl -X POST "https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=fr" \
-H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY" \
-H "Ocp-Apim-Subscription-Region: YOUR_REGION" \
-H "Content-Type: application/json" \
-d "[{'Text':'Hello, how are you?'}]"
```

Use Azure AI services

- **The endpoint URI. This is the HTTP address at which the REST interface for the service can be accessed.**

```
curl -X POST "https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=fr" \
-H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY" \
-H "Ocp-Apim-Subscription-Region: YOUR_REGION" \
-H "Content-Type: application/json" \
-d "[{'Text':'Hello, how are you?'}]"
```

Use Azure AI services

- A subscription key. Access to the endpoint is restricted based on a subscription key. Client applications must provide a valid key to consume the service.

```
curl -X POST "https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=fr" \
-H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY" \
-H "Ocp-Apim-Subscription-Region: YOUR_REGION" \
-H "Content-Type: application/json" \
-d "[{'Text':'Hello, how are you?'}]"
```

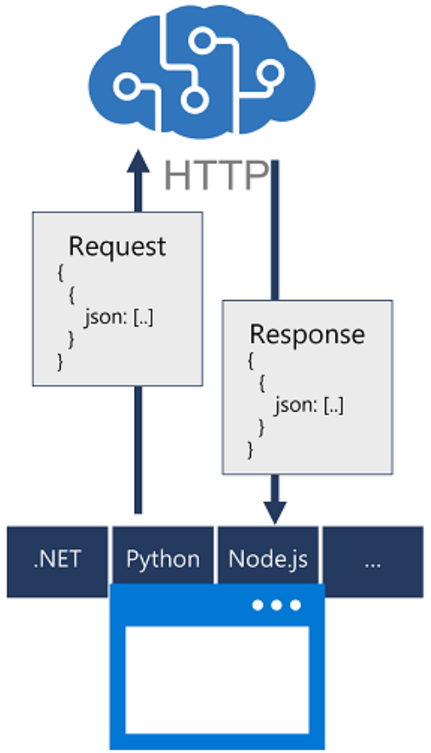
Use Azure AI services

- The resource location. When you provision a resource in Azure, you generally assign it to a location, which determines the Azure data center in which the resource is defined.

```
curl -X POST "https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=fr" \
-H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY" \
-H "Ocp-Apim-Subscription-Region: YOUR_REGION" \
-H "Content-Type: application/json" \
-d "[{'Text':'Hello, how are you?'}]"
```

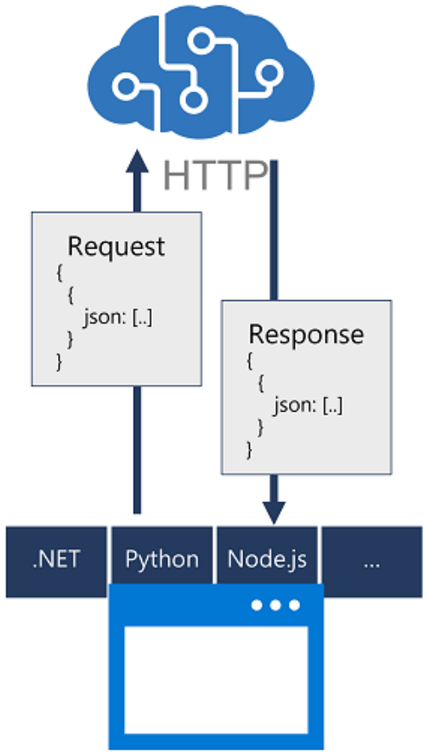
Use Azure AI services

- We can use Azure AI Services with:
  - SDK
  - Software development kits (SDKs) for common programming languages abstract the REST interfaces for most AI services.
  - SDK availability varies by individual AI services, but for most services there's an SDK for languages such as:
    - Microsoft C# (.NET Core)
    - Python
    - JavaScript (Node.js)
    - Go
    - Java



Use Azure AI services

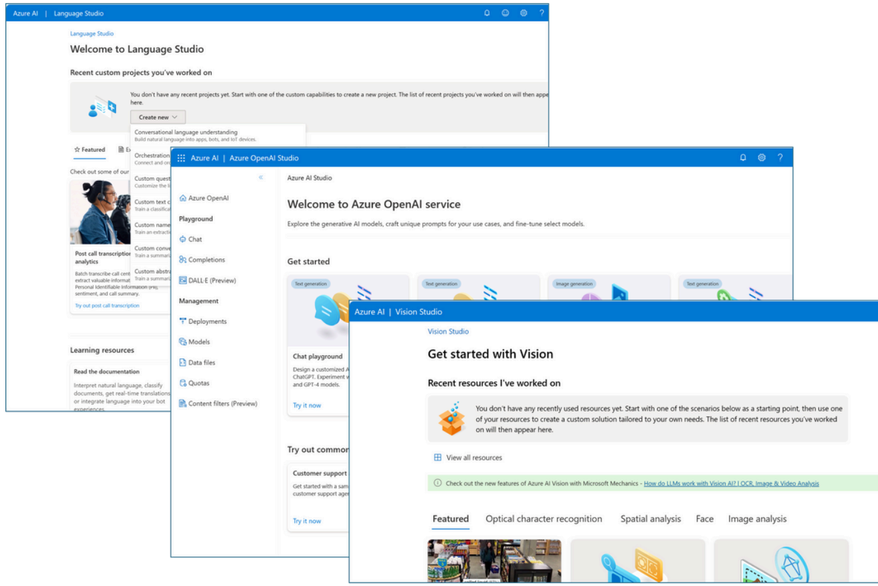
- We can use Azure AI Services with:
  - SDK




```python
from azure.core.credentials import AzureKeyCredential
from azure.ai.translation.text import TextTranslationClient

# Set your Translator service credentials
subscription_key = "YOUR_SUBSCRIPTION_KEY"
endpoint = "https://api.cognitive.microsofttranslator.com"

# Initialize the TextTranslationClient with credentials and endpoint
credential = AzureKeyCredential(subscription_key)
client = TextTranslationClient(endpoint, credential)

# Define the text to translate and the target language
text_to_translate = "Hello, how are you?"
target_language = "fr"  # French

# Call the translate method
response = client.translate(content=[{"Text": text_to_translate}], to=[target_language])

# Process the response
for translation in response:
    for translation_text in translation.translations:
        print(f"Translated text: {translation_text.text}")
```

Use Azure AI services

- We can use Azure AI Services with:
  - service studio interfaces
  - Studio interfaces provide a friendly user interface to explore Azure AI services
  - There are different studios for different Azure AI services, such as Vision Studio, Language Studio, Speech Studio, and the Content Safety Studio



BEFORE YOU CAN USE AN AI SERVICE RESOURCE, YOU MUST ASSOCIATE IT WITH THE STUDIO YOU WANT TO USE ON THE SETTINGS PAGE

Computer Vision

01 Machine Learning

02 Computer Vision

03 Natural Language Processing

04 Document Intelligence & Knowledge Mininng

05 Generative AI

Check your knowledge on machine learning part

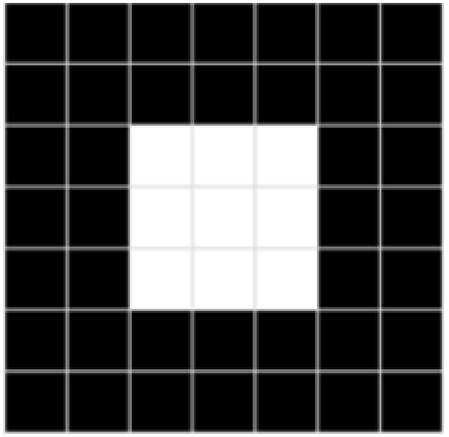
Fundamentals of Computer Vision

- Computer vision is one of the core areas of artificial intelligence (AI)
- enables AI to “see” the world and make sense of it
- it’s the ability of processing images and videos and interpret them

Images and image processing

- An image is an array of numeric pixel values

```
0   0   0   0   0   0   0
0   0   0   0   0   0   0
0   0 255 255 255   0   0
0   0 255 255 255   0   0
0   0 255 255 255   0   0
0   0   0   0   0   0   0
0   0   0   0   0   0   0
```



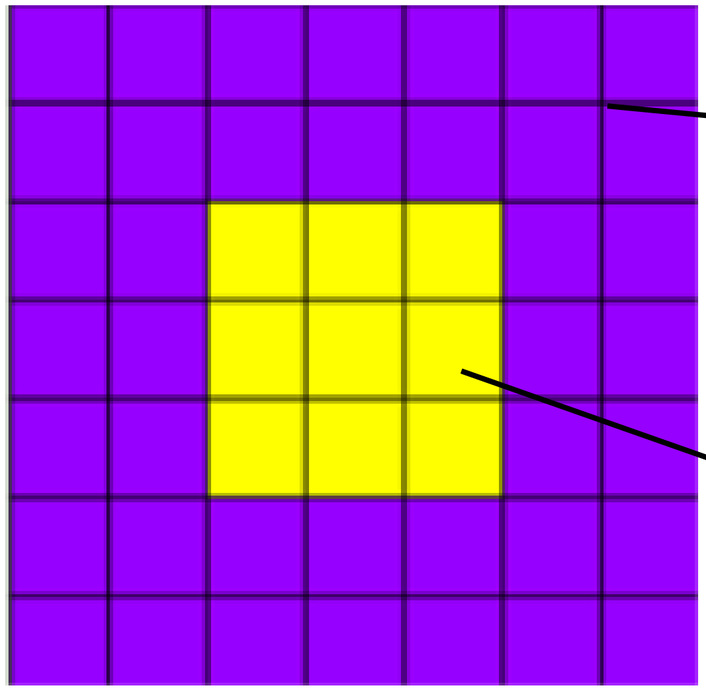
- The array consists of seven rows and seven columns, representing the pixel values for a 7x7 pixel image (which is known as the image's resolution)
- Each pixel has a value between 0 (black) and 255 (white); with values between these bounds representing shades of gray
- The array of pixel values for this image is two-dimensional (representing rows and columns, or x and y coordinates) and defines a single rectangle of pixel values

Images and image processing

- In reality, most digital images are multidimensional and consist of three layers (known as channels) that represent red, green, and blue (RGB) color hues

```
Red:
 150  150  150  150  150  150  150
 150  150  150  150  150  150  150
 150  150  255  255  255  150  150
 150  150  255  255  255  150  150
 150  150  255  255  255  150  150
 150  150  150  150  150  150  150
 150  150  150  150  150  150  150

Green:
 0    0    0    0    0    0    0
 0    0    0    0    0    0    0
 0    0   255  255  255   0    0
 0    0   255  255  255   0    0
 0    0   255  255  255   0    0
 0    0    0    0    0    0    0
 0    0    0    0    0    0    0

Blue:
 255  255  255  255  255  255  255
 255  255  255  255  255  255  255
 255  255   0    0    0   255  255
 255  255   0    0    0   255  255
 255  255   0    0    0   255  255
 255  255  255  255  255  255  255
 255  255  255  255  255  255  255
```

The purple squares are represented by the combination:



Red: 150
Green: 0
Blue: 255

The yellow squares are represented by the combination:

Red: 255
Green: 255
Blue: 0

Filters to process images

- A common way to perform image processing tasks is to apply filters that modify the pixel values of the image to create a visual effect.

- A filter is defined by one or more arrays of pixel values, called filter kernels

- Filter with a 3x3 kernel:

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

- The kernel is then convolved across the image, calculating a weighted sum for each 3x3 patch of pixels and assigning the result to a new image.

Example image filters

START

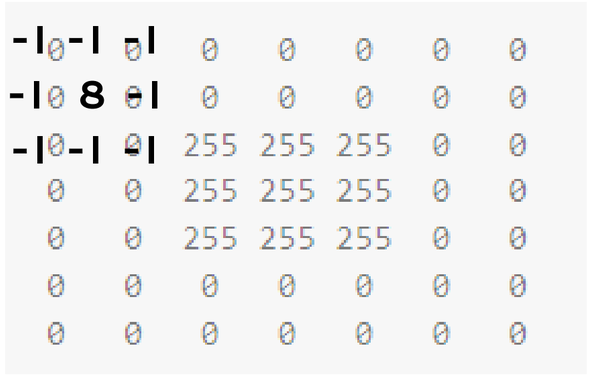
- Filter with a 3x3 kernel:

```
-1 -1 -1
-1  8 -1
-1 -1 -1
```

- Image:

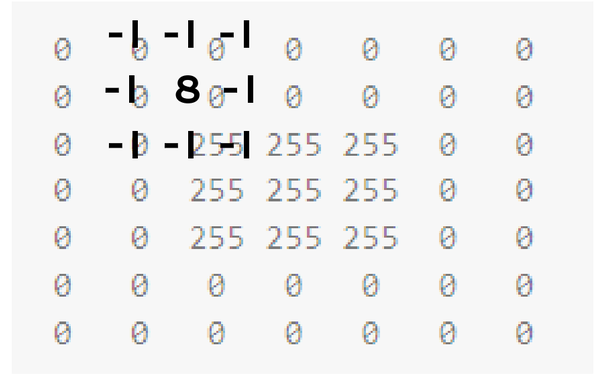
```
0   0   0   0   0   0   0
0   0   0   0   0   0   0
0   0 255 255 255   0   0
0   0 255 255 255   0   0
0   0 255 255 255   0   0
0   0   0   0   0   0   0
0   0   0   0   0   0   0
```

- First, we apply the filter kernel to the top left patch of the image. Multiplying each pixel value by the corresponding weight value in the kernel and adding the reuslts



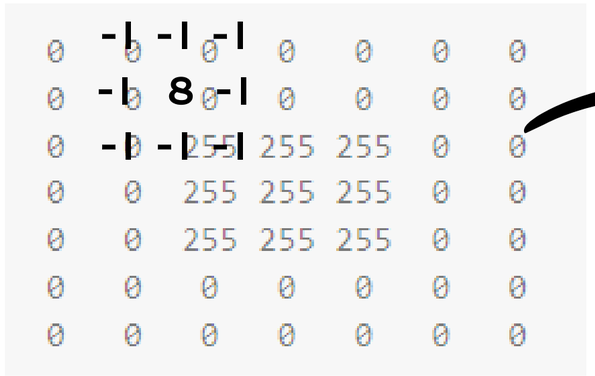

```
(0 x -1) + (0 x -1) + (0 x -1) +
(0 x -1) + (0 x 8) + (0 x -1) +
(0 x -1) + (0 x -1) + (255 x -1) = -255
```

The result (-255) becomes the first value in a new array



Then we move the filter kernel along one pixel to the right and repeat the operation

Example image filters




```
(0 x -1) + (0 x -1) + (0 x -1) +
(0 x -1) + (0 x 8) + (0 x -1) +
(0 x -1) + (255 x -1) + (255 x -1) = -510
```

New Image -255 -510 .. ...

... ...

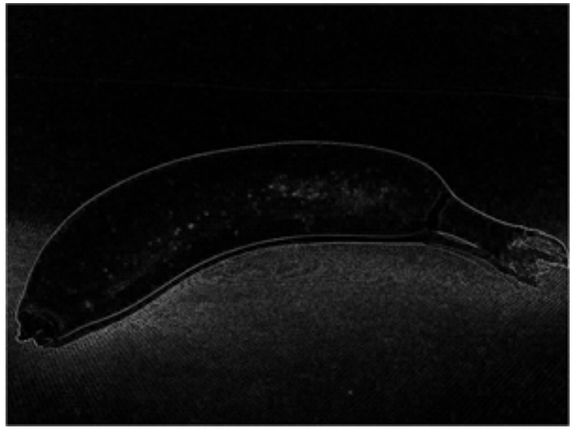
Then we move the filter kernel along one pixel to the right and repeat the operation

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

-255	-510	-765	-510	-255
-510	1275	765	1275	-510
-765	765	0	765	-765
-510	1275	765	1275	-510
-255	-510	-765	-510	-255

Example image filters

- This kind of image manipulation is often referred to as convolutional filtering.



- The filter used in this example is a particular type of filter (called a laplace filter) that highlights the edges on objects in an image

- There are many other kinds of filter that you can use to create blurring, sharpening, color inversion, and other effects.

Machine learning for computer vision

- The goal of computer vision is often to extract meaning, or at least actionable insights, from images

- Requires the creation of machine learning models that are trained to recognize features based on large volumes of existing images.

- Most common machine learning model architectures for computer vision is a convolutional neural network (CNN)

Convolutional neural networks (CNNs)

- CNNs use filters to extract numeric feature maps from images, and then feed the feature values into a deep learning model to generate a label prediction.

EXAMPLE

- You might train a CNN model with images of different kinds of fruit (such as apple, banana and organe)

- During the training process for a CNN, filter kernels are initially defined using randomly generated weight values

- Then, as the training process progresses, the models predictions are evaluated against known label values, and the filter weights are adjusted to improve accuracy.

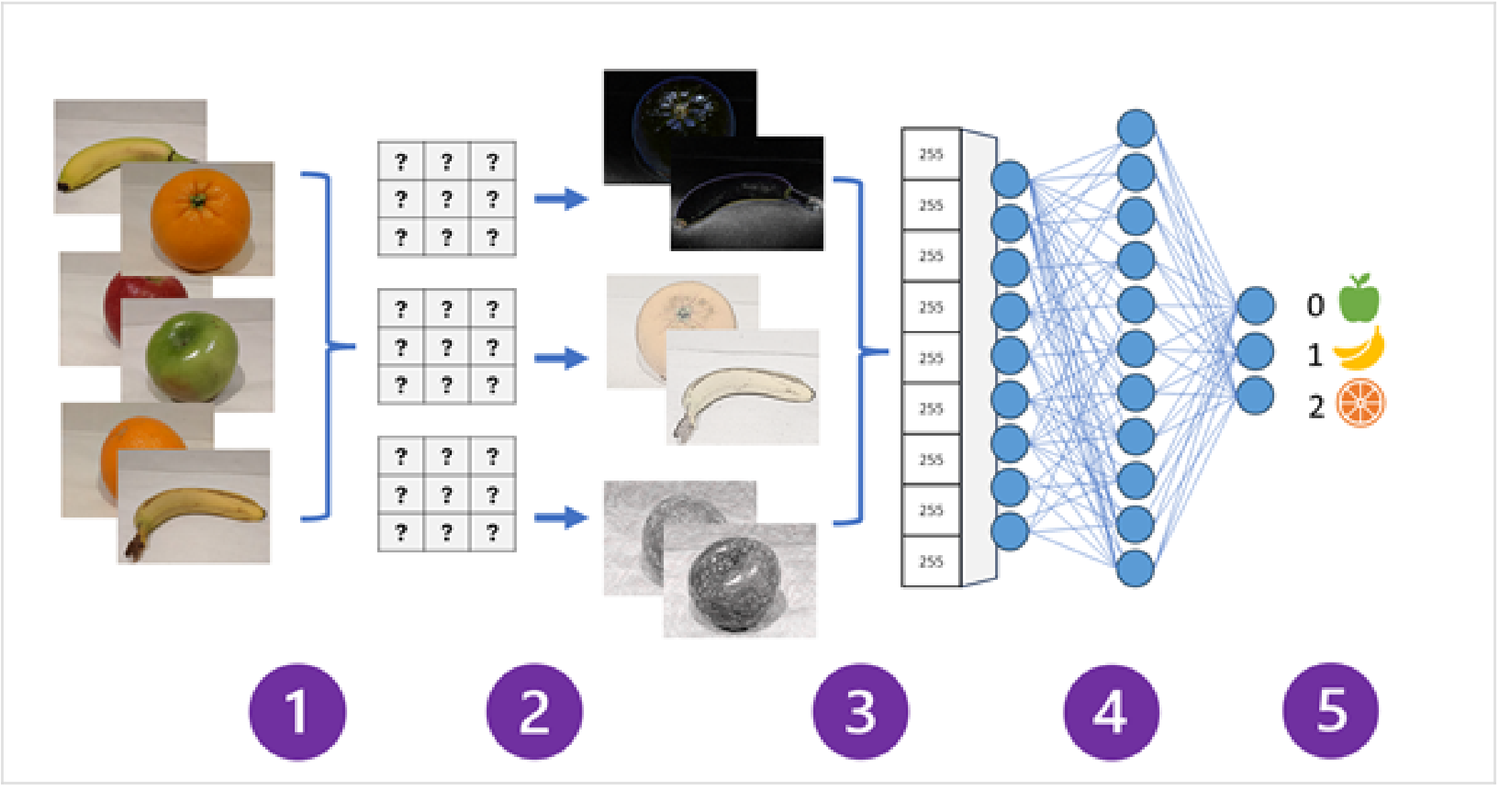
CNN for Image Classification

1) Images with known labels (for example, 0: apple, 1: banana, or 2: orange) are fed into the network to train the model.

2) One or more layers of filters is used to extract features from each image as it is fed through the network. The filter kernels start with randomly assigned weights and generate arrays of numeric values called feature maps.

3) The feature maps are flattened into a single dimensional array of feature values.

4) The feature values are fed into a fully connected neural network.



5) The output layer of the neural network uses a softmax or similar function to produce a result that contains a probability value for each possible class, for example [0.2, 0.5, 0.3].

Introducing Transformers

- Many years, Convolutional Neural Networks (CNNs) have been the backbone of computer vision

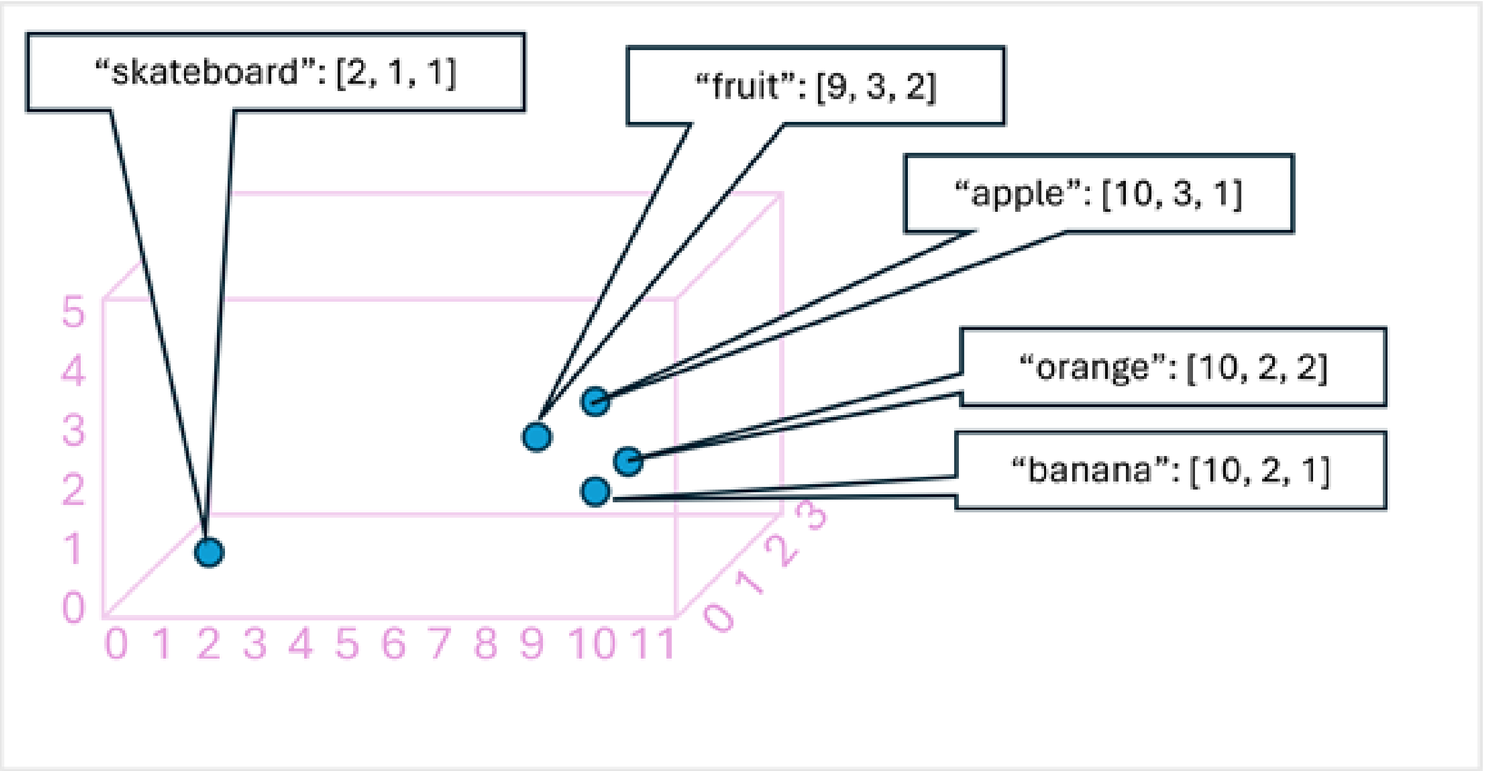
- They have been used for various tasks, from simple image classification to more complex object detection

- But the world of AI is constantly evolving and a new player has entered the game:
  TRANSFORMERS

- Transformers, originally developed for natural language processing, have shown remarkable potential in computer vision. But how do they work?

Transformers

- Imagine words as points in a three-dimensional space.



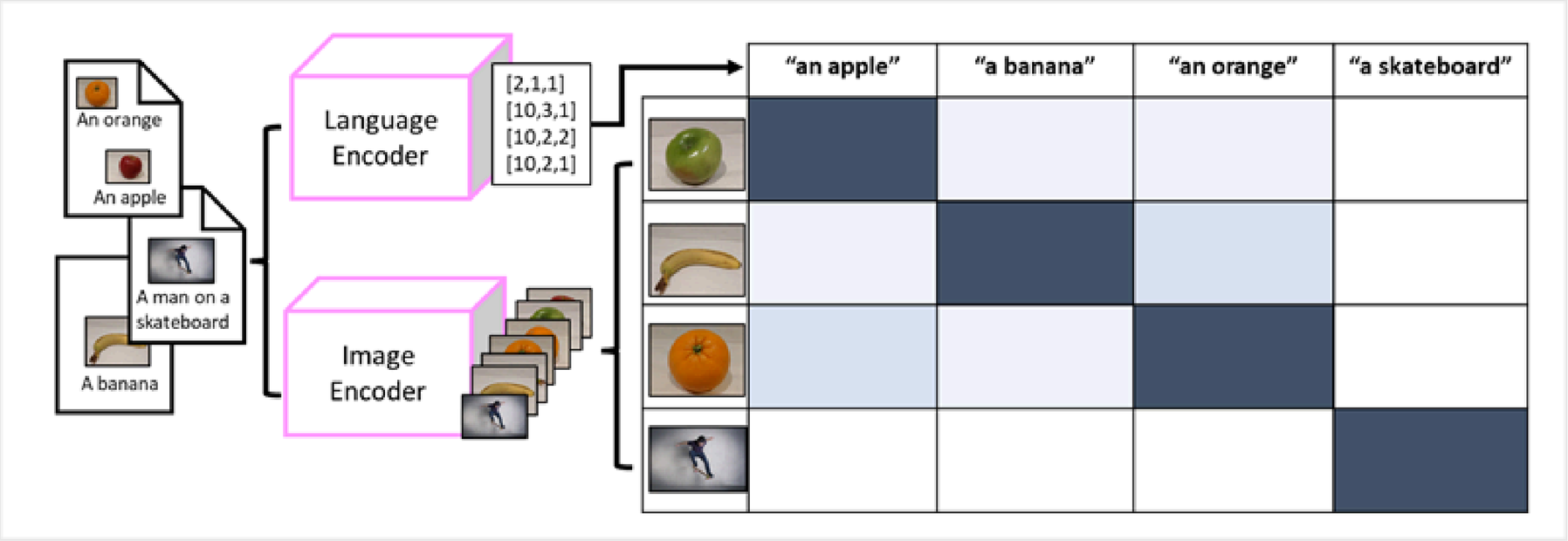
- semantically similar words are positioned close to each

- Transformers process vast amounts of data, encoding words or phrases as numeric arrays called embeddings.

Multi-Modal Models

- These models are trained on a large volume of captioned images, without fixed labels.

- Uses an image encoder to extract features from images based on pixel values, and combine these with text embeddings from a language encoder.



Microsoft Florence model

- Trained with huge volumes of captioned images from the Internet, it includes both a language encoder and an image encoder

- Florence is an example of a foundation model. In other words, a pre-trained general model on which you can build multiple adaptive models for specialist tasks

- You can use Florence as a foundation model for adaptive models that perform:
  - Image classification: Identifying to which category an image belongs.
  - Object detection: Locating individual objects within an image.
  - Captioning: Generating appropriate descriptions of images.
  - Tagging: Compiling a list of relevant text tags for an image.

Azure AI Vision

- The architecture for computer vision models can be complex (require significant volumes of training images and compute power to perform the training process)

- Microsoft's Azure AI Vision service provides prebuilt and customizable computer vision models (based on Florence foundation model)

- With Azure AI Vision, you can create sophisticated computer vision solutions quickly and easily; taking advantage of "off-the-shelf" functionality for many common computer vision scenarios, while retaining the ability to create custom models using your own images.

Azure AI Vision

- Azure AI Vision supports multiple image analysis capabilities, including:
  - Optical character recognition (OCR) - extracting text from images.
  - Generating captions and descriptions of images.
  - Detection of thousands of common objects in images.
  - Tagging visual features in images

- These tasks, and more, can be performed in Azure AI Vision Studio, with REST API or SDK.

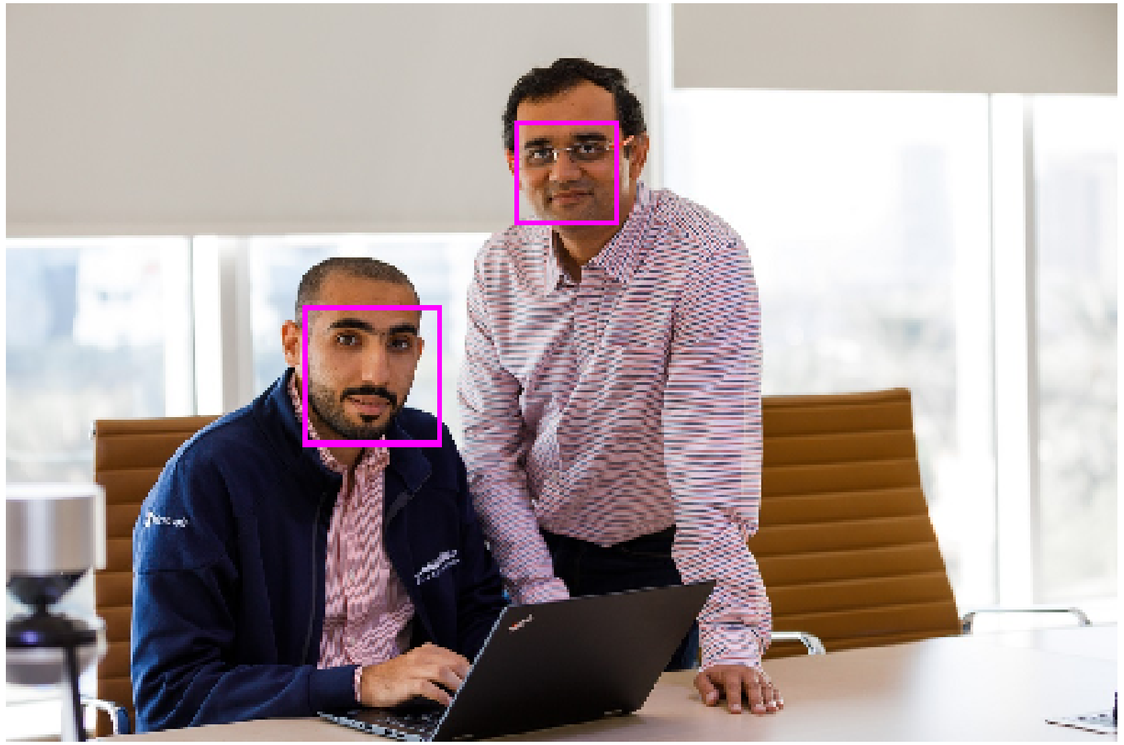
Facial Recognition

- Face detection and analysis is an area of artificial intelligence (AI) which uses algorithms to locate and analyze human faces in images or video content.

- There are many applications for face detection, analysis, and recognition.
  - Security - facial recognition can be used in building security applications, and increasingly it is used in smart phones operating systems for unlocking devices
  - Social media - facial recognition can be used to automatically tag known friends in photographs.
  - Intelligent monitoring - for example, an automobile might include a system that monitors the driver's face to determine if the driver is looking at the road, looking at a mobile device, or shows signs of tiredness.

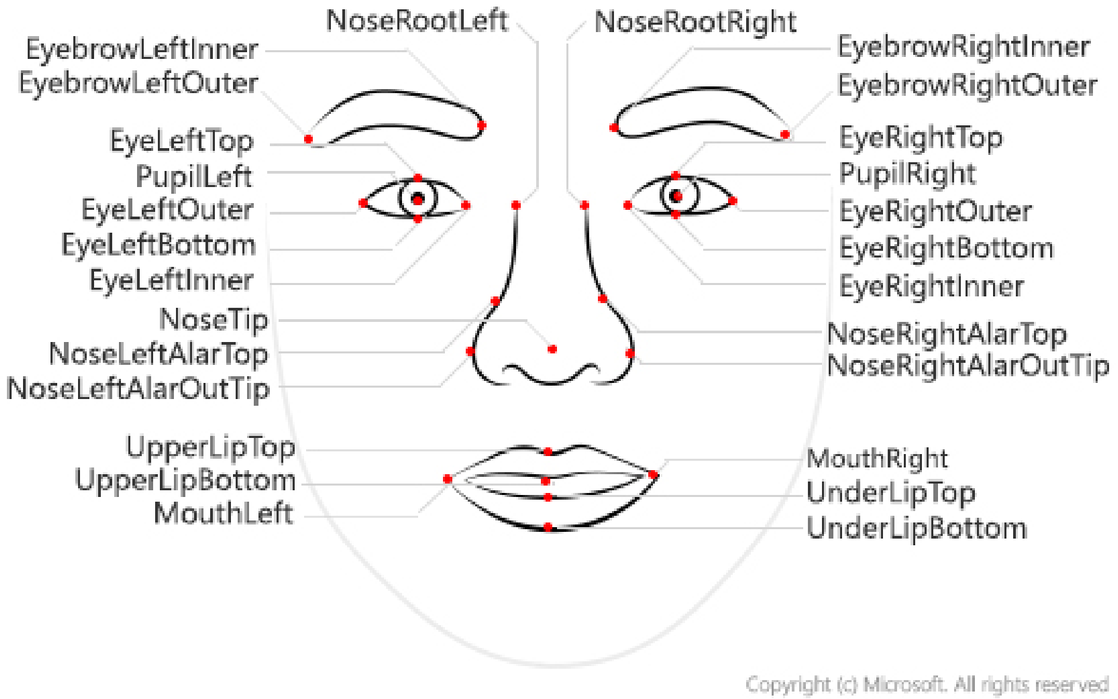
Understand Face analysis

- Face detection involves identifying regions of an image that contain a human face, typically by returning bounding box coordinates that form a rectangle around the face, like this:



Understand Face analysis

- With Face analysis, facial features can be used to train machine learning models to return other information, such as facial features such as nose, eyes, eyebrows, lips, and others.



Understand Face analysis

- A further application of facial analysis is to train a machine learning model to identify known individuals from their facial features.

- This is known as facial recognition, and uses multiple images of an individual to train the model. This trains the model so that it can detect those individuals in new images on which it wasn't trained.

Face Analysis Recap

1) Face Detection
- Identifies faces in images
- Returns bounding box coordinates

2) Feature Analysis
- Extracts information about facial features (eyes, nose, lips, etc.)
- Used to train machine learning models

3)Facial Recognition
- Identifies specific individuals
- Requires multiple training images per person
- Can detect known individuals in new images

Face analysis on Azure

- Microsoft Azure provides multiple Azure AI services that you can use to detect and analyze faces, including:
    - Azure AI Vision, which offers face detection and some basic face analysis, such as returning the bounding box coordinates around an image.
    - Azure AI Video Indexer, which you can use to detect and identify faces in a video.
    - Azure AI Face, which offers pre-built algorithms that can detect, recognize, and analyze faces.

- AI Face offers the widest range of facial analysis capabilities.

Azure AI Face service

- The Azure Face service can return the rectangle coordinates for any human faces that are found in an image, as well as a series of attributes related to those face such as:

Accessories
indicates whether the given face has accessories. This attribute returns possible accessories including headwear, glasses, and mask, with confidence score between zero and one for each accessory.

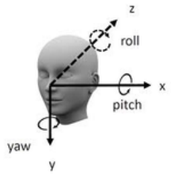
Blur
how blurred the face is, which can be an indication of how likely the face is to be the main focus of the image.

Exposure
such as whether the image is underexposed or over exposed. This applies to the face in the image and not the overall image exposure.

Azure AI Face service

- The Azure Face service can return the rectangle coordinates for any human faces that are found in an image, as well as a series of attributes related to those face such as:

Glasses
whether or not the person is wearing glasses.



Head pose
the face's orientation in a 3D space.

Mask
indicates whether the face is wearing a mask.

Responsible AI use

- Anyone can use the Face service to:
  - Detect the location of faces in an image.
  - Determine if a person is wearing glasses.
  - Determine if there's occlusion, blur, noise, or over/under exposure for any of the faces.
  - Return the head pose coordinates for each face in an image.

- The Limited Access policy requires customers to submit an intake form to access additional Azure AI Face service capabilities including:
  - The ability to compare faces for similarity.
  - The ability to identify named individuals in an image.

Azure resources for Face

- To use the Face service, you must create one of the following types of resource in your Azure subscription:
  - Face: Use this specific resource type if you don't intend to use any other Azure AI services, or if you want to track utilization and costs for Face separately.
  - Azure AI services: A general resource that includes Azure AI Face along with many other Azure AI services such as Azure AI Content Safety, Azure AI Language, and others. Use this resource type if you plan to use multiple Azure AI services and want to simplify administration and development.

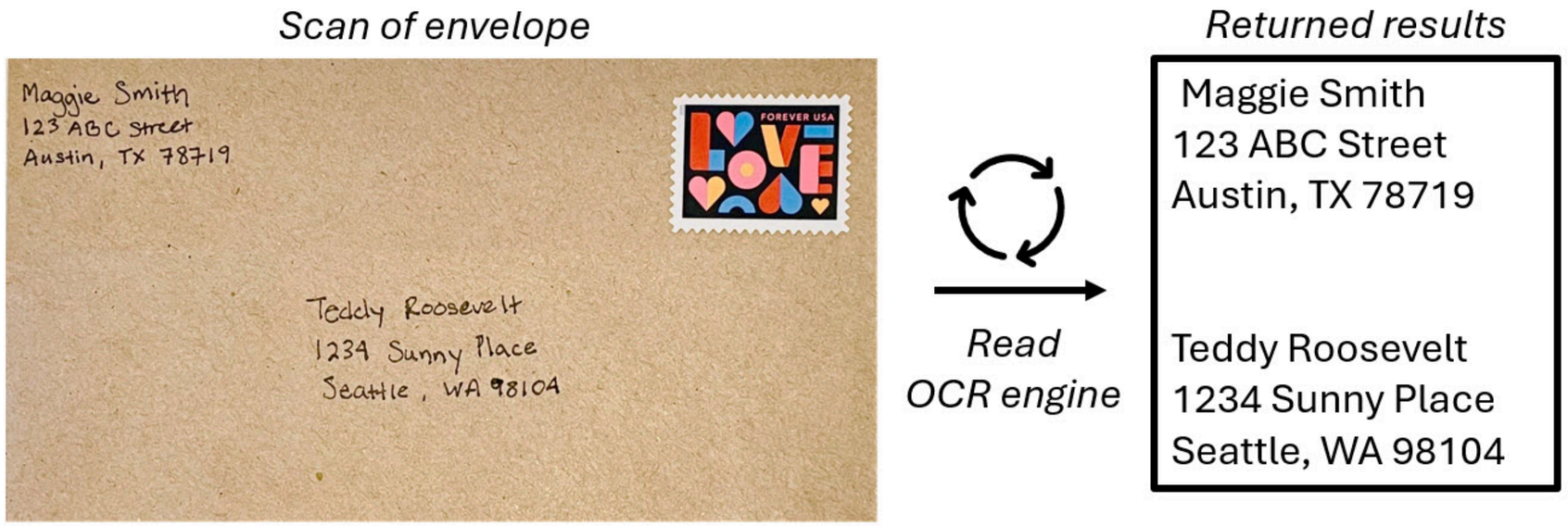
Introduction of Opticcal Character Recognition

- The capability for artificial intelligence (AI) to process words in images into machine-readable text.

- It's the intersection of computer vision with natural language processing
  - Vision capabilities are needed to "read" the text
  - Natural language processing capabilities make sense of it

Introduction of Opticcal Character Recognition

- OCR is the foundation of processing text in images and uses machine learning models that are trained to recognize individual shapes as letters, numerals, punctuation, or other elements of text.

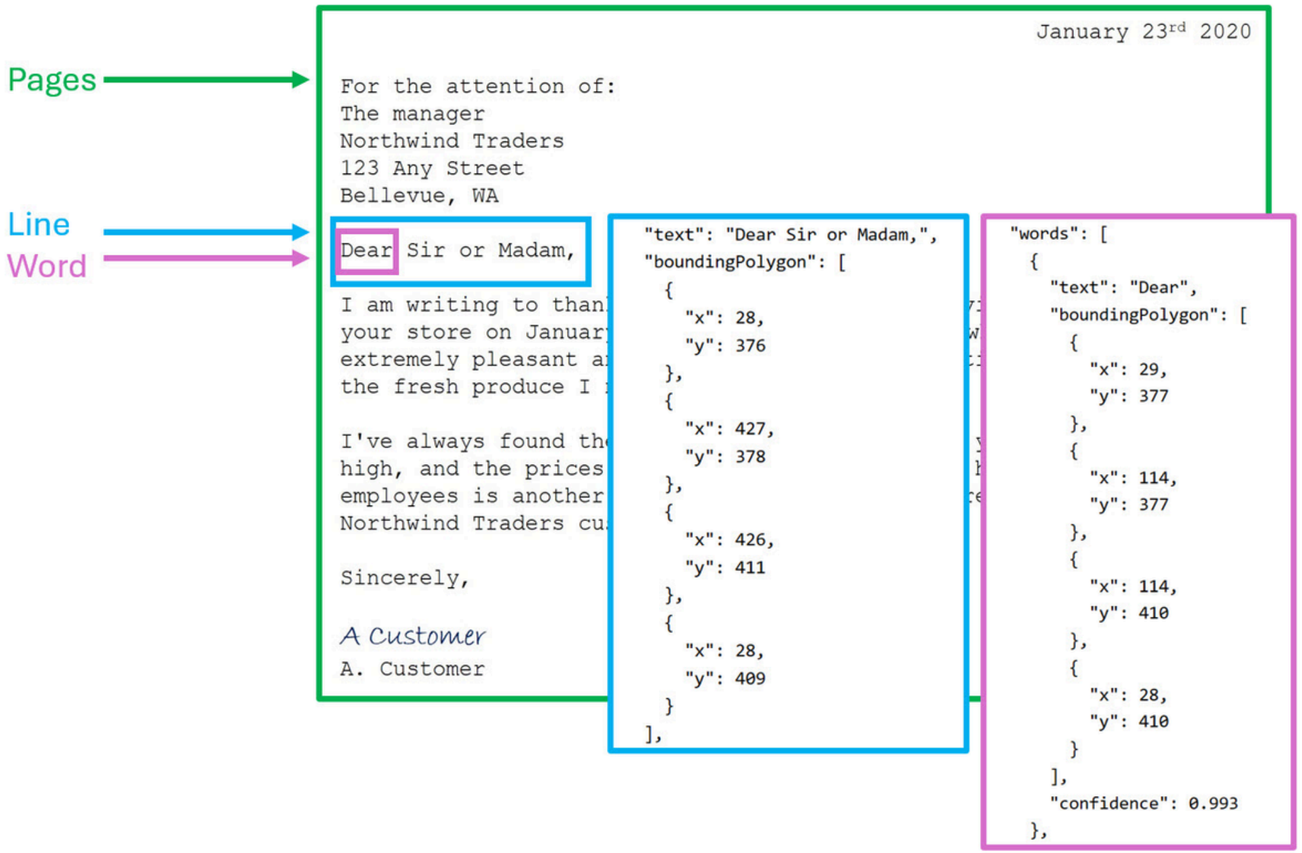


Azure AI Vision’s OCR Engine

- Azure AI Vision service has the ability to extract machine-readable text from image
- Azure AI Vision's Read API is the OCR engine that powers text extraction from images, PDFs, and TIFF files.
- The Read API, otherwise known as Read OCR engine, uses the latest recognition models and is optimized for images that have a significant amount of text or have considerable visual noise.
- The OCR engine takes in an image file and identifies bounding boxes, or coordinates, where items are located within an image. In OCR, the model identifies bounding boxes around anything that appears to be text in the image.

Azure AI Vision's OCR Engine

- Calling the Read API returns results arranged into the following hierarchy:
  - Pages - One for each page of text, including information about the page size and orientation.
  - Lines - The lines of text on a page.
  - Words - The words in a line of text, including the bounding box coordinates and text itself.



- Each line and word includes bounding box coordinates indicating its position on the page.

Use Vision's OCR Engine

- To use the AI Vision service, you must create one of the following types of resource in your Azure subscription:
  - Azure AI Vision: A specific resource for vision services. Use this resource type if you don't intend to use any other AI services, or if you want to track utilization and costs for your AI Vision resource separately.
  - Azure AI services: A general resource that includes Azure AI Vision along with many other Azure AI services
- There are several ways to use Azure AI Vision's Read API:
  - Vision Studio
  - REST API
  - Software Development Kits (SDKs): Python, C#, JavaScript

Natural Language Processing – NLP

01 Machine Learning

02 Computer Vision

03 Natural Language Processing

04 Document Intelligence & Knowledge Mininng

05 Generative AI

Check your knowledge on computer vision part

NLP Introduction

- In order for computer systems to interpret the subject of a text in a similar way humans do, they use natural language processing (NLP)

- Natural language processing might be used to create:
  - A social media feed analyzer that detects sentiment for a product marketing campaign.
  - A document search application that summarizes documents in a catalog.
  - An application that extracts brands and company names from text.

Tokenization

- Some of the earliest techniques used to analyze text with computers involve statistical analysis of a body of text (a corpus) to infer some kind of semantic meaning

- If you can determine the most commonly used words in a given document, you can often get a good idea of what the document is about

- The first step in analyzing a corpus is to break it down into tokens.

- Tokens can be generated for partial words, or combinations of words and punctuation.

Tokenization

- "we choose to go to the moon"

- The phrase can be broken down into the following tokens, with numeric identifiers:
  - we
  - choose
  - to
  - go
  - the
  - moon

Notice that "to" (token number 3) is used twice in the corpus.

The phrase "we choose to go to the moon" can be represented by the tokens [1,2,3,4,3,5,6].

Frequency analysis

- After tokenizing the words, you can perform some analysis to count the number of occurrences of each token. The most commonly used words (other than stop words such as "a", "the", and so on) can often provide a clue as to the main subject of a text corpus

- the most common words in the entire text of the "go to the moon" --> include "new", "go", "space", and "moon".

- If we were to tokenize the text as bi-grams (word pairs), the most common bi-gram in the speech is "the moon".

Machine learning for text classification

- Another useful text analysis technique is to use a classification algorithm, such as logistic regression, to train a machine learning model that classifies text based on a known set of categorizations.

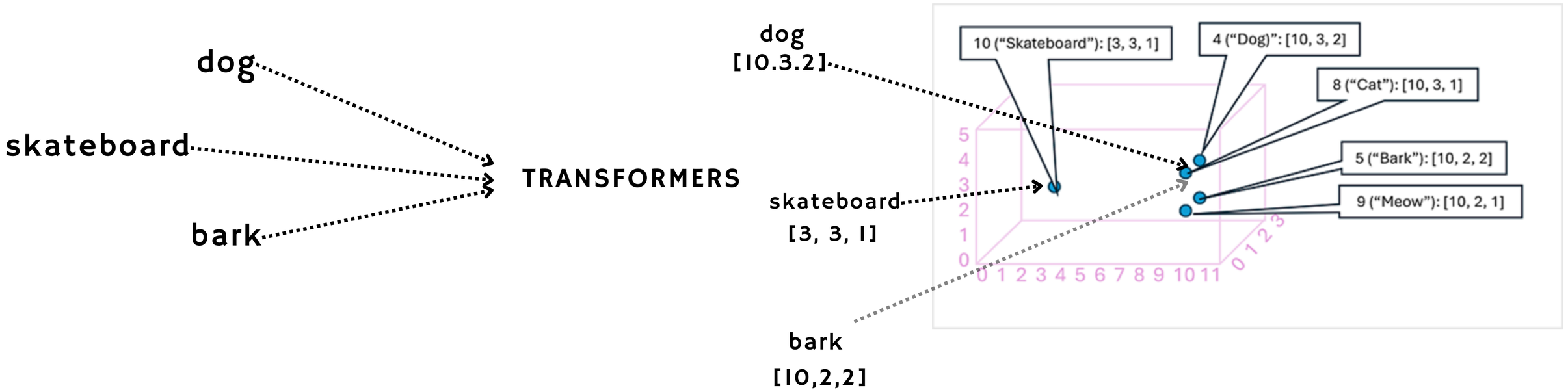
- For example, consider the following restaurant reviews, which are already labeled as 0 (negative) or 1 (positive):
  - The food and service were both great: 1
  - A really terrible experience: 0
  - Mmm! tasty food and a fun vibe: 1
  - Slow service and substandard food: 0

- Reviews with tokens for words like "great", "tasty", or "fun" are more likely to return a sentiment of 1 (positive), while reviews with words like "terrible", "slow", and "substandard" are more likely to return 0 (negative).

Semantic language models

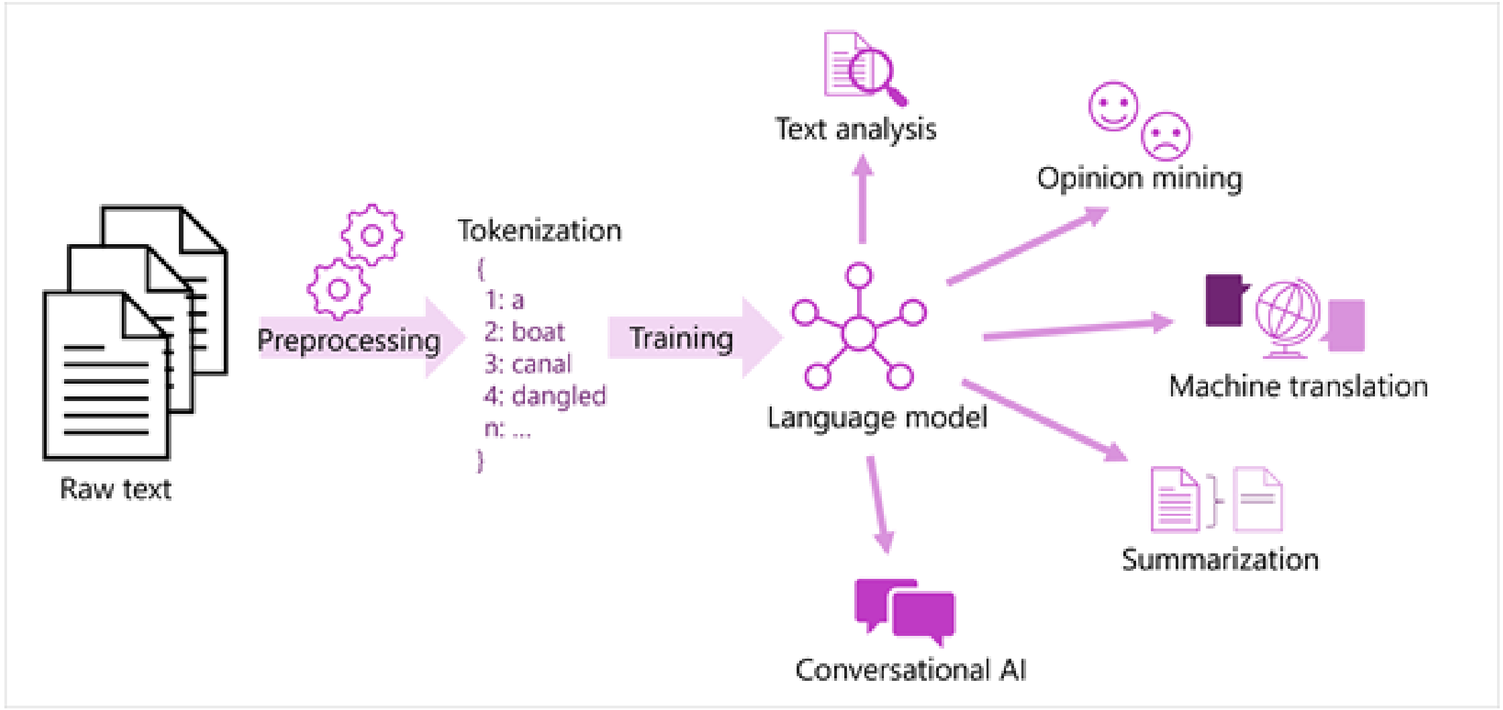
- Semantic language models are advanced NLP models that capture relationships between tokens.
- Core concept: Encoding language tokens as vectors called embeddings.
- Embeddings represent tokens as coordinates in multidimensional space.
- Semantically related tokens are grouped closer together in this space.

Semantic language models



- Words that have similar semantic meaning are closer to each other in the vector space

Semantic language models



Semantic language models

- Common NLP tasks supported by language models include:
  - Text analysis, such as extracting key terms or identifying named entities in text.
  - Sentiment analysis and opinion mining to categorize text as positive or negative.
  - Machine translation, in which text is automatically translated from one language to another.
  - Summarization, in which the main points of a large body of text are summarized.
  - Conversational AI solutions such as bots or digital assistants in which the language model can interpret natural language input and return an appropriate response.

Semantic language models vs LLM's

- Architecture:
  - Semantic language models: These are typically focused on understanding and representing the meaning of words and sentences. Examples include BERT and Word2Vec.
  - ChatGPT: It's based on a large language model (LLM) architecture, specifically the GPT (Generative Pre-trained Transformer) family.

- Purpose and capabilities:
  - Semantic models: Primarily designed for tasks like understanding context, semantic similarity, and word relationships.
  - ChatGPT: Designed for more general-purpose language understanding and generation, including conversation, content creation, and complex reasoning.

Azure AI Language

- Azure AI Language is a part of the Azure AI services offerings that can perform advanced natural language processing over unstructured text. Azure AI Language's text analysis features include:
  - Named entity recognition identifies people, places, events, and more. This feature can also be customized to extract custom categories.
  - Entity linking identifies known entities together with a link to Wikipedia.
  - Personal identifying information (PII) detection identifies personally sensitive information, including personal health information (PHI).
  - Language detection identifies the language of the text and returns a language code such as "en" for English.
  - Sentiment analysis and opinion mining identifies whether text is positive or negative.
  - Summarization summarizes text by identifying the most important information.
  - Key phrase extraction lists the main concepts from unstructured text

Entity recognition and linking

- You can provide Azure AI Language with unstructured text and it will return a list of entities in the text that it recognizes. An entity is an item of a particular type or a category

Category	Description	Examples
Person	Names of people	John Smith, Marie Curie
Location	Names of places, landmarks, geographic features	Paris, Mount Everest, Amazon River
Organization	Names of companies, institutions, agencies	Microsoft, United Nations, NASA
DateTime	Date and time expressions	January 1st 2024, next Monday, 3:30 PM
Quantity	Numeric expressions, including percentages	42, 3.14, 75%
Event	Names of events or happenings	Olympic Games, World War II
Address	Physical addresses	123 Main St, New York, NY 10001
Email	Email addresses	example@email.com
URL	Web addresses	https://www.example.com
IP Address	Internet Protocol addresses	192.168.0.1
Phone Number	Telephone numbers	+1 (555) 123-4567
Product	Names of products or services	iPhone, Microsoft Office
Skill	Abilities or areas of expertise	programming, cooking, project management
Job Title	Professional positions or roles	CEO, Software Engineer, Teacher

Entity recognition and linking

"I ate at the restaurant in Seattle last week with Bill Gates."

Entity	Type	SubType	Wikipedia URL
Seattle	Location		https://en.wikipedia.org/wiki/Seattle
last week	DateTime	DateRange	
Bill Gates	Person		https://en.wikipedia.org/wiki/Bill_Gates

Language detection

- You can use language detection capability of Azure AI Language to identify the language in which text is written.

- For each document submitted the service will detect:
  - The language name (for example "English").
  - The ISO 639-1 language code (for example, "en").
  - A score indicating a level of confidence in the language detection.

- Review 1: "A fantastic place for lunch. The soup was delicious."
- Review 2: "Comida maravillosa y gran servicio."
- Review 3: "The croque monsieur avec frites was terrific. Bon appetit!"

Document	Language Name	ISO 6391 Code	Score
Review 1	English	en	1.0
Review 2	Spanish	es	1.0
Review 3	English	en	0.9

Sentiment analysis and opinion mining

- Azure AI Language can evaluate text and return sentiment scores and labels for each sentence. This capability is useful for detecting positive and negative sentiment in social media, customer reviews, discussion forums and more.

- Azure AI Language uses a prebuilt machine learning classification model to evaluate the text

- The service returns sentiment scores in three categories: positive, neutral, and negative

- In each of the categories, a score between 0 and 1 is provided. Scores indicate how likely the provided text is a particular sentiment.

Sentiment analysis and opinion mining

- Review I: "We had dinner at this restaurant last night and the first thing I noticed was how courteous the staff was. We were greeted in a friendly manner and taken to our table right away. The table was clean, the chairs were comfortable, and the food was amazing."

Document sentiment:

- Positive Positive score: .90
- Neutral score: .I0
- Negative score: .00

Sentiment analysis and opinion mining

- Review 2: "Our dining experience at this restaurant was one of the worst I've ever had. The service was slow, and the food was awful. I'll never eat at this establishment again."

Document sentiment:

- Positive score: .00
- Neutral score: .00
- Negative score: .99

Key phrase extraction

- Key phrase extraction identifies the main points from text

- "We had dinner here for a birthday celebration and had a fantastic experience. We were greeted by a friendly hostess and taken to our table right away. The ambiance was relaxed, the food was amazing, and service was terrific. If you like great food and attentive service, you should try this place."

- Key phrase extraction can provide some context to this review by extracting the following phrases:
  - birthday celebration
  - fantastic experience
  - friendly hostess
  - great food
  - attentive service
  - dinner
  - table
  - ambiance

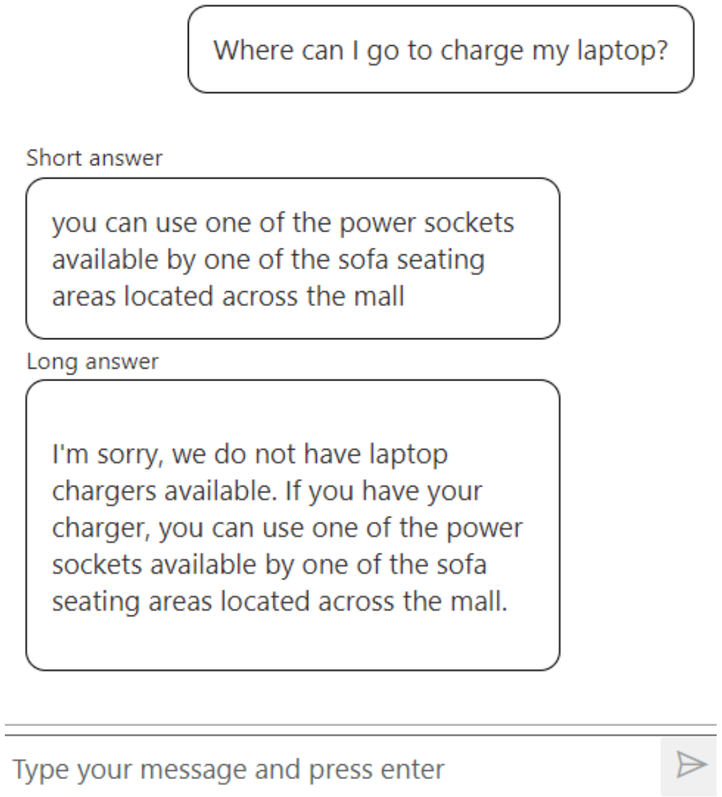
Create a resource for Azure AI Language

- To use Azure AI Language in an application, you must provision an appropriate resource in your Azure subscription. You can choose either of the following types of resource:
    - A Language resource - choose this resource type if you only plan to use Azure AI Language services, or if you want to manage access and billing for the resource separately from other services.
    - An Azure AI services resource - choose this resource type if you plan to use Azure AI Language in combination with other Azure AI services, and you want to manage access and billing for these services together.

Question answering

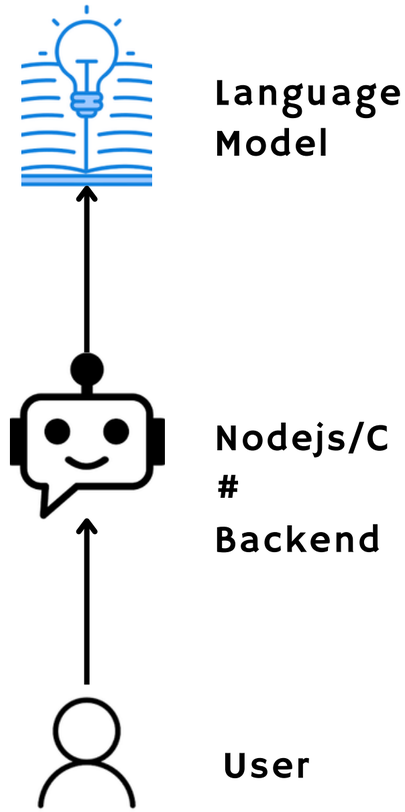
- Question answering is a custom feature of Azure AI Language, that finds the most appropriate answer for inputs from your users

- is commonly used to build conversational client applications, such as social media applications, chat bots, and speech-enabled desktop applications.



Create your own Chatbot

- You can easily create a user support bot solution on Microsoft Azure using a combination of two core services:
  - Azure AI Language: includes a custom question answering feature that enables you to create a knowledge base of question and answer pairs that can be queried using natural language input.
  - Azure AI Bot Service: provides a framework for developing, publishing, and managing bots on Azure.



Creating a custom question answering knowledge base

- You can use Azure AI Language Studio to create, train, publish, and manage question answering projects.
  - (To create a project, you must first provision a Language resource in your Azure subscription.)

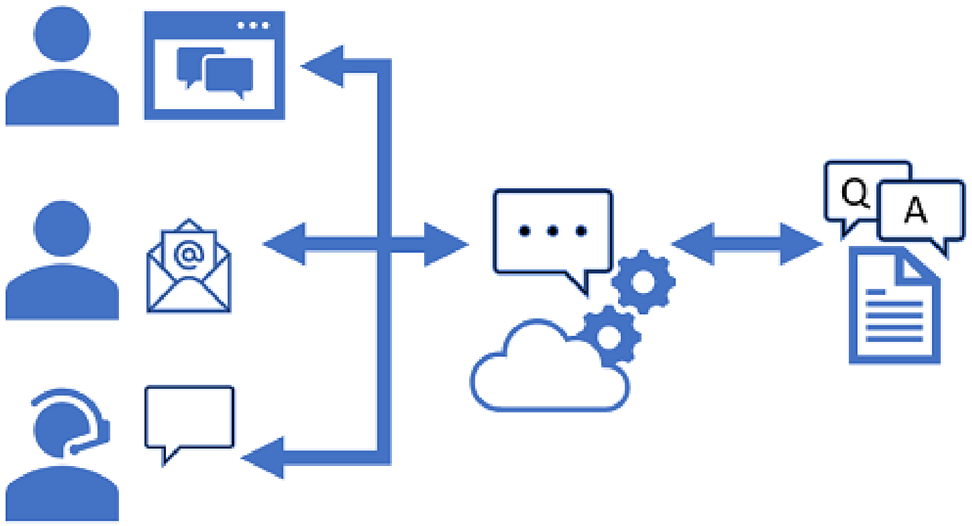
- Get questions from existing FAQ documents
- Write your own questions and answers
- Do both: use existing FAQs and add more
- Add different ways to ask the same question
  - This helps match similar questions to the right answer

Chosing the right bot

Solution	Target Audience	Key Features
(Power Virtual Agents) Copilot Studio	Fusion teams, citizen developers	• No coding required • 600+ prebuilt data connectors• Integrates with Microsoft 365 and Dynamics 365
Health Bot	Healthcare organizations	• Built-in medical database • Triage protocols • Compliant AI-powered assistants
Bot Framework SDK	Developers	• Modular and extensible SDK • Tools for end-to-end development • Integration with Azure services

Connect channels

- When your bot is ready to be delivered to users, you can connect it to multiple channels; making it possible for users to interact with it through web chat, email, Microsoft Teams, and other common communication media.



Fundamentals of conversational language understanding

- Alan Turing came up with a game (Imitation Game) in 1950 to see if computers could talk like humans

- Computers need not only to be able to accept language as input (either in text or audio format), but also to be able to interpret the semantic meaning of the input - in other words, understand what is being said.

- Common scenarios for this kind of solution include customer support applications, reservation systems, and home automation among others.

} Conversational AI

Conversational language understanding

- To work with conversational language understanding, you need to take into account three core concepts: utterances, entities, and intents.

Utterances: "What Users Say"

- Utterances
  - An utterance is an example of something a user might say, and which your application must interpret. For example, when using a home automation system, a user might use the following utterances:
    - "Switch the fan on."
    - "Turn on the light."

- Utterances are used to train the model to identify the most likely intent and the entities to which it should be applied based on a given input.

Entities: "The Key Objects"

- Entities
  - An entity is an item to which an utterance refers. For example, fan and light in the following utterances:

    - "Switch the fan on."
    - "Turn on the light."

Intents: "The User's Goal"

- Intents
  - An intent represents the purpose, or goal, expressed in a user's utterance.
  - For example, for both of the previously considered utterances, the intent is to turn a device on; so in your conversational language understanding application, you might define a TurnOn intent that is related to these utterances.

Overview

Intent	Related Utterances	Entities
Greeting	"Hello"	
	"Hi"	
	"Hey"	
	"Good morning"	
TurnOn	"Switch the fan on"	fan (device)
	"Turn the light on"	light (device)
	"Turn on the light"	light (device)
TurnOff	"Switch the fan off"	fan (device)
	"Turn the light off"	light (device)
	"Turn off the light"	light (device)

None Intent: "The Catchall"

Intent	Related Utterances	Entities
None	"What is the meaning of life?"	
	"Is this thing on?"	

- The intent should be a concise way of grouping the utterance tasks
- Of special interest is the None intent. You should consider always using the None intent to help handle utterances that do not map any of the utterances you have entered

Azure resources for conversational language understanding

To use conversational language capabilities in Azure, you need a resource in your Azure subscription. You can use the following types of resource:

- Azure AI Language: A resource that enables you to build apps with industry-leading natural language understanding capabilities without machine learning expertise. You can use a language resource for authoring and prediction.

- Azure AI services: A general resource that includes conversational language understanding along with many other Azure AI services. You can only use this type of resource for prediction.

The Lifecycle of Azure Conversational Language Understanding

Authoring

- Create authoring resource
- Define entities and intents
- Add sample utterances
- Use prebuilt domains or create custom
- Use Language Studio for easy authoring

creating the conversational AI model

Training the model

- Feed model with defined data
- Use utterances to teach model
- Match expressions to intents/entities
- Test model with sample text
- Iterate: train, test, update, repeat

teaching the conversational AI model

Predicting

- Publish trained model
- Connect client apps to endpoint
- Submit user input for prediction
- Get predicted intents and entities
- Take action based on predictions

putting the conversational AI model it into use

Understand speech recognition and synthesis

- AI speech capabilities enable us to manage home and auto systems with voice instructions, get answers from computers for spoken questions, generate captions from audio, and much more.

- To enable this kind of interaction, the AI system must support two capabilities:

    - Speech recognition - the ability to detect and interpret spoken input
    - Speech synthesis - the ability to generate spoken output

Azure AI Speech

- Azure AI Speech provides speech to text and text to speech capabilities through speech recognition and synthesis.

- You can use prebuilt and custom Speech service models for a variety of tasks
  - transcribing audio to text
  - identifying speakers in conversations
  - creating custom voices

Speech recognition

- Speech recognition takes the spoken word and converts it into data that can be processed - often by transcribing it into text.
- The spoken words can be in the form of a recorded voice in an audio file, or live audio from a microphone
- Speech patterns are analyzed in the audio to determine recognizable patterns that are mapped to words
- To accomplish this, the software typically uses multiple models, including:
  - An acoustic model that converts the audio signal into phonemes (smallest unit of sound in a language).
  - A language model that maps phonemes to words, usually using a statistical algorithm that predicts the most probable sequence of words based on the phonemes.

Speech recognition

- The recognized words are typically converted to text, which you can use for various purposes, such as:
  - Providing closed captions for recorded or live videos
  - Creating a transcript of a phone call or meeting
  - Automated note dictation
  - Determining intended user input for further processing

Speech synthesis

- Speech synthesis is concerned with vocalizing data, usually by converting text to speech. A speech synthesis solution typically requires the following information:
    - The text to be spoken
    - The voice to be used to vocalize the speech

- To synthesize speech, the system typically tokenizes the text to break it down into individual words, and assigns phonetic sounds to each word.
- It then breaks the phonetic transcription into prosodic units (such as phrases, clauses, or sentences) to create phonemes that will be converted to audio format.
- These phonemes are then synthesized as audio and can be assigned a particular voice, speaking rate, pitch, and volume.

Speech synthesis

- You can use the output of speech synthesis for many purposes, including:
    - Generating spoken responses to user input
    - Creating voice menus for telephone systems
    - Reading email or text messages aloud in hands-free scenarios
    - Broadcasting announcements in public locations, such as railway stations or airports

Speech on Azure

- Microsoft Azure offers both speech recognition and speech synthesis capabilities through Azure AI Speech service, which includes the following application programming interfaces (APIs):
    - The Speech to text API (Speech recognition)
    - The Text to speech API (Speech synthesis)

- To use Azure AI Speech in an application, you must create an appropriate resource in your Azure subscription. You can choose to create either of the following types of resource:
    - A Speech resource - use it of you only plan to use Azure AI Speech
    - An Azure AI services resource - choose this resource type if you plan to use Azure AI Speech in combination with other Azure AI services

The Speech to text API

- You can use Azure AI Speech to text API to perform real-time or batch transcription of audio into a text format
- The audio source for transcription can be a real-time audio stream from a microphone or an audio file.
- The model that is used by the Speech to text API, is based on the Universal Language Model that was trained by Microsoft. The Model is optimized for two scenarios
    - conversational
    - dictation
- You can also create and train your own custom models including acoustics, language, and pronunciation if the pre-built models from Microsoft do not provide what you need.

Real-time transcription

- Real-time speech to text allows you to transcribe text in audio streams. You can use real-time transcription for presentations, demos, or any other scenario where a person is speaking.

- In order for real-time transcription to work, your application will need to be listening for incoming audio from a microphone, or other audio input source such as an audio file. Your application code streams the audio to the service, which returns the transcribed text.

Batch transcription

- Batch transcription should be run in an asynchronous manner
- Batch jobs are scheduled on a best-effort basis.
- Normally a job will start executing within minutes of the request but there is no estimate for when a job changes into the running state.
- You can point to audio files with a shared access signature (SAS) URI and asynchronously receive transcription results.
  - A Shared Access Signature (SAS) URI is a secure way to share access to resources in Azure storage accounts without sharing the account keys.

The text to speech API

- The text to speech API enables you to convert text input to audible speech, which can either be played directly through a computer speaker or written to an audio file.
- You can specify the voice to be used to vocalize the text.
- The service includes multiple pre-defined voices with support for multiple languages and regional pronunciation, including neural voices that leverage neural networks
- You can also develop custom voices and use them with the text to speech API

Understand translation in Azure

- Microsoft provides Azure AI services that support translation. Specifically, you can use the following services:
  - The Azure AI Translator service, which supports text-to-text translation.
  - The Azure AI Speech service, which enables speech to text and speech-to-speech translation.

Understand translation concepts

- Early attempts at machine translation applied literal translations
  - A literal translation is where each word is translated to the corresponding word in the target language

- This approach presents some issues.
  - For one case, there may not be an equivalent word in the target language
  - Another case is where literal translation can change the meaning of the phrase or not get the context correct.

- Artificial intelligence systems must be able to understand, not only the words, but also the semantic context in which they're used.

Azure AI Translator

- Azure AI Translator is easy to integrate in your applications, websites, tools, and solutions

  - Uses Neural Machine Translation (NMT) for accurate translations
  - Supports text-to-text translation between 130+ languages
  - Requires specifying 'from' and 'to' languages using ISO 639-1 codes
  - Can translate to multiple languages simultaneously

- When using Azure AI Translator, you can specify one from language with multiple to languages, enabling you to simultaneously translate a source document into multiple languages.

- You can use Azure AI Translator with a programming language of your choice or the REST API. You can use some of its features with Language Studio.

Using Azure AI Translator

- Azure AI Translator includes the following capabilities:
  - Text translation - used for quick and accurate text translation in real time across all supported languages.
  - Document translation - used to translate multiple documents across all supported languages while preserving original document structure.
  - Custom translation - used to enable enterprises, app developers, and language service providers to build customized neural machine translation (NMT) systems.

- Azure AI Translator's application programming interface (API) offers some optional configuration to help you fine-tune the results that are returned, including:
  - Profanity filtering. Profanity levels are typically culture-specific but you can control profanity translation by either marking the translated text as profane or by omitting it in the results.
  - Selective translation. You can tag content so that it isn't translated. For example, you may want to tag code, a brand name, or a word/phrase that doesn't make sense when localized.

Azure AI Speech

- You can use Azure AI Speech to translate spoken audio from a streaming source, such as a microphone or audio file, and return the translation as text or an audio stream.
  - Enables real-time captioning and two-way translation
  - Supports translation from one source to 90+ target languages
  - Source language must use extended code format (e.g., 'es-US')
  - Target languages use two-letter codes

- You can use Azure AI Speech with Speech Studio or a programming language of your choice or the REST API.

- Azure AI Speech includes the following capabilities:
  - Speech to text - used to transcribe speech from an audio source to text format.
  - Text to speech - used to generate spoken audio from a text source.
  - Speech Translation - used to translate speech in one language to text or speech in another.

Document Intelligence & Knowledge Mining

01 Machine Learning

02 Computer Vision

03 Natural Language Processing

04 Document Intelligence & Knowledge Mininng

05 Generative AI

Check your knowledge on Natural Language Processing part

Introduction to Document Intelligence

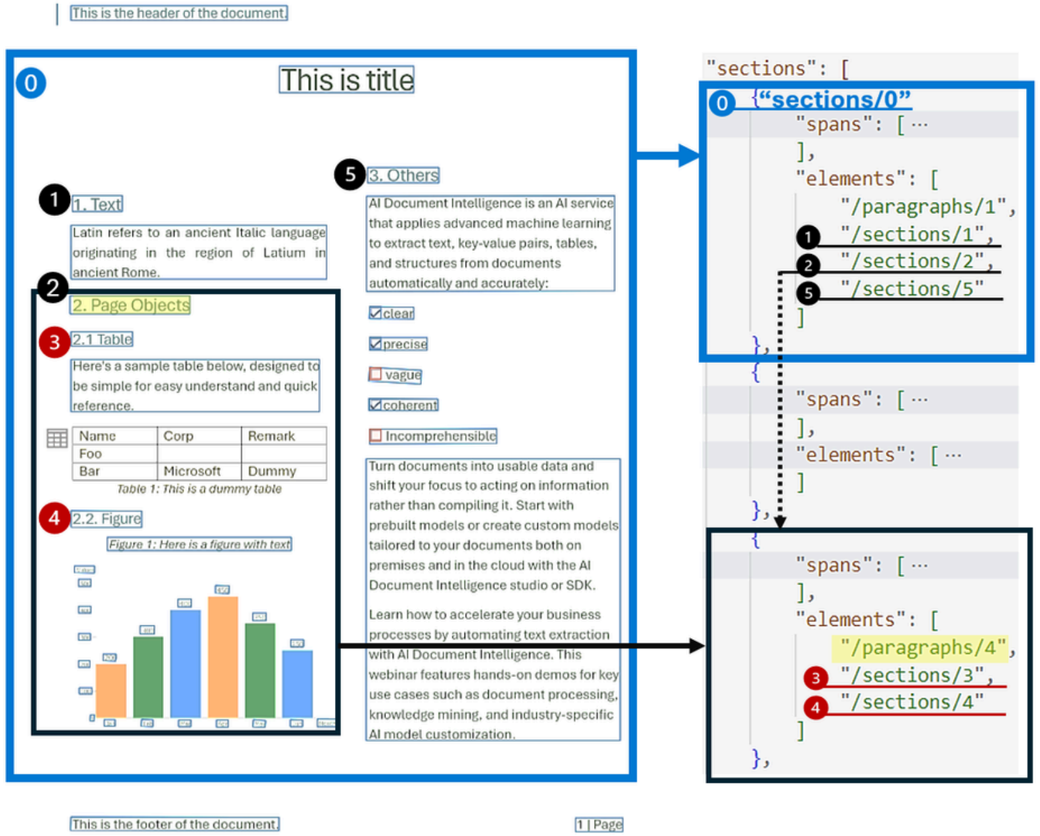
- AI capability for processing and extracting information from text-based documents

- Key capabilities:
  - Digitizes text from images (OCR foundation)
  - Comprehends document content and context
  - Automatically identifies and extracts specific data points

- Benefits for organizations:
  - Streamlines document-heavy processes
  - Reduces manual data entry
  - Improves efficiency
  - Reduces costs

Capabilities of document intelligence

- Document Intelligence relies on machine learning
- The ability to extract text, layout, and key-value pairs is known as document analysis
- Document analysis provides locations of text on a page identified by bounding box coordinates.
- Machine learning models can interpret the data in a document or form because they're trained to recognize patterns in bounding box coordinate locations and text.



Challenges of document intelligence

- Diverse document formats: Documents come in a wide variety of layouts, structures, and content types, making it difficult to create a one-size-fits-all solution.
- Handwritten or low-quality text: OCR can struggle with handwritten text or low-resolution scans, leading to errors in text extraction.
- Complex document structures: Extracting information from tables, forms, and other structured document elements requires advanced AI models.
- Contextual understanding: Interpreting the meaning and relationships within a document requires deep language understanding, beyond just recognizing individual words.

Prebuilt Document Intelligence Models

- To address the challenges of document intelligence, many AI platforms offer prebuilt models that are trained on common document types. These prebuilt models can:
  - Recognize and extract data from invoices, receipts, business cards, and other common document formats
  - Identify key fields like customer/vendor names, dates, totals, and more
  - Understand the structure and layout of documents to locate relevant information
  - Provide confidence scores on the accuracy of extracted data

Custom Document Intelligence Models

- For documents that don't fit the mold of prebuilt models, organizations can create custom AI models tailored to their specific requirements. Custom models allow for:
    - Training on unique document formats and layouts
    - Extracting specialized data fields not covered by prebuilt models
    - Incorporating domain-specific knowledge and business rules
    - Ongoing fine-tuning and retraining as document requirements evolve

Azure AI Document Intelligence

- Provides access to prebuilt and custom document intelligence models
- Leverages Azure Cognitive Services for text extraction and analysis
- Offers a user-friendly interface for testing and deploying models
- Integrates with other Azure services for end-to-end document workflows

Introduction to Knowledge Mining

- Knowledge mining is the process of extracting valuable insights and information from large volumes of data, often unstructured or semi-structured.
- It involves using advanced technologies like natural language processing, machine learning, and artificial intelligence to analyze and make sense of complex data.
- Knowledge mining can help organizations quickly find relevant information, uncover hidden patterns, and gain new perspectives from their data.

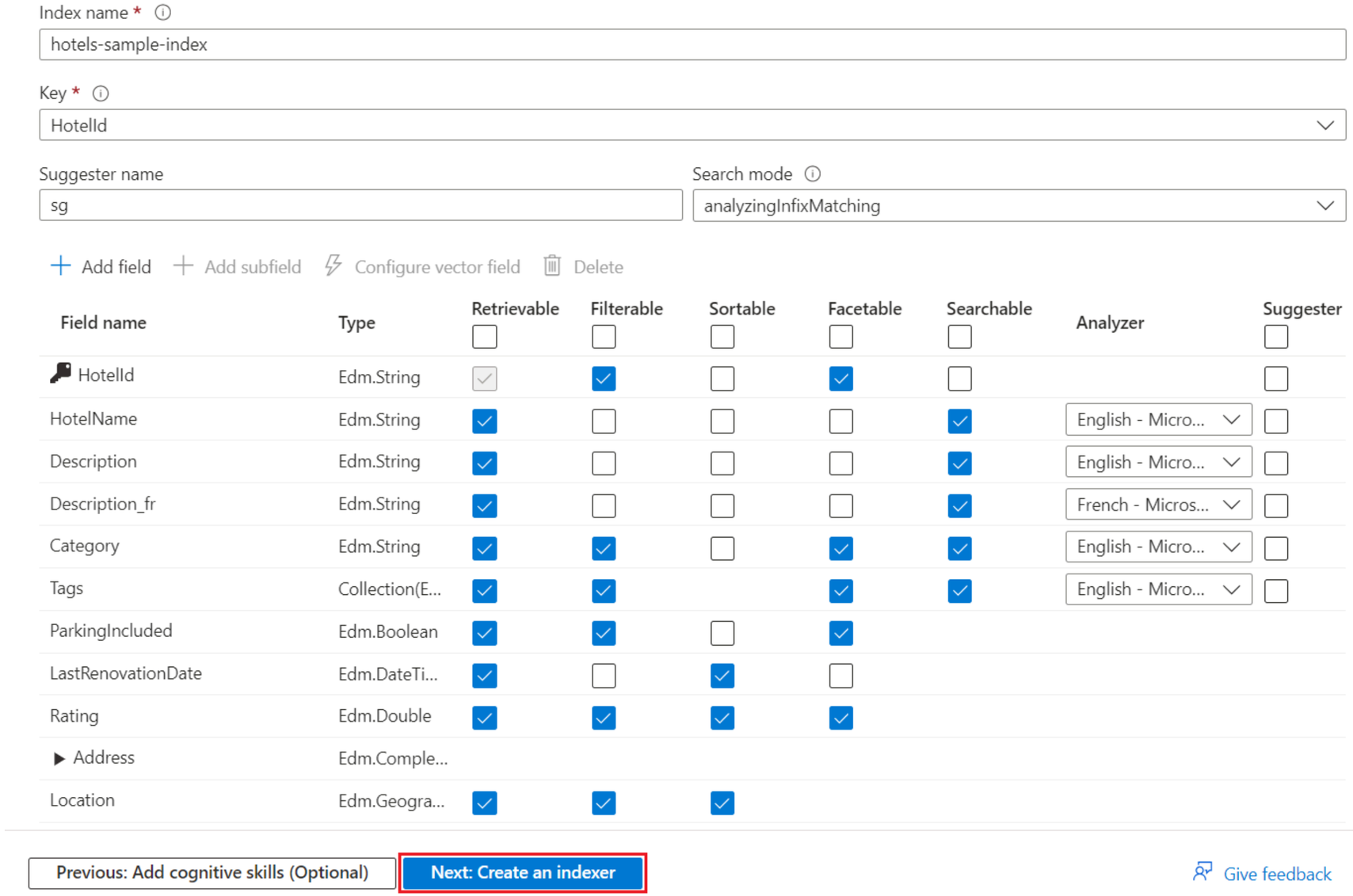
What is Azure AI Search?

- Azure AI Search is a cloud-based search service that provides the infrastructure and tools to create powerful search solutions.
- It can index and search a wide variety of data sources, including structured, semi-structured, and unstructured content.
- Azure AI Search leverages the built-in capabilities of Azure AI services, such as image processing, content extraction, and natural language processing, to perform advanced knowledge mining.
- This allows organizations to index and search previously unsearchable documents and extract valuable insights from large amounts of data.

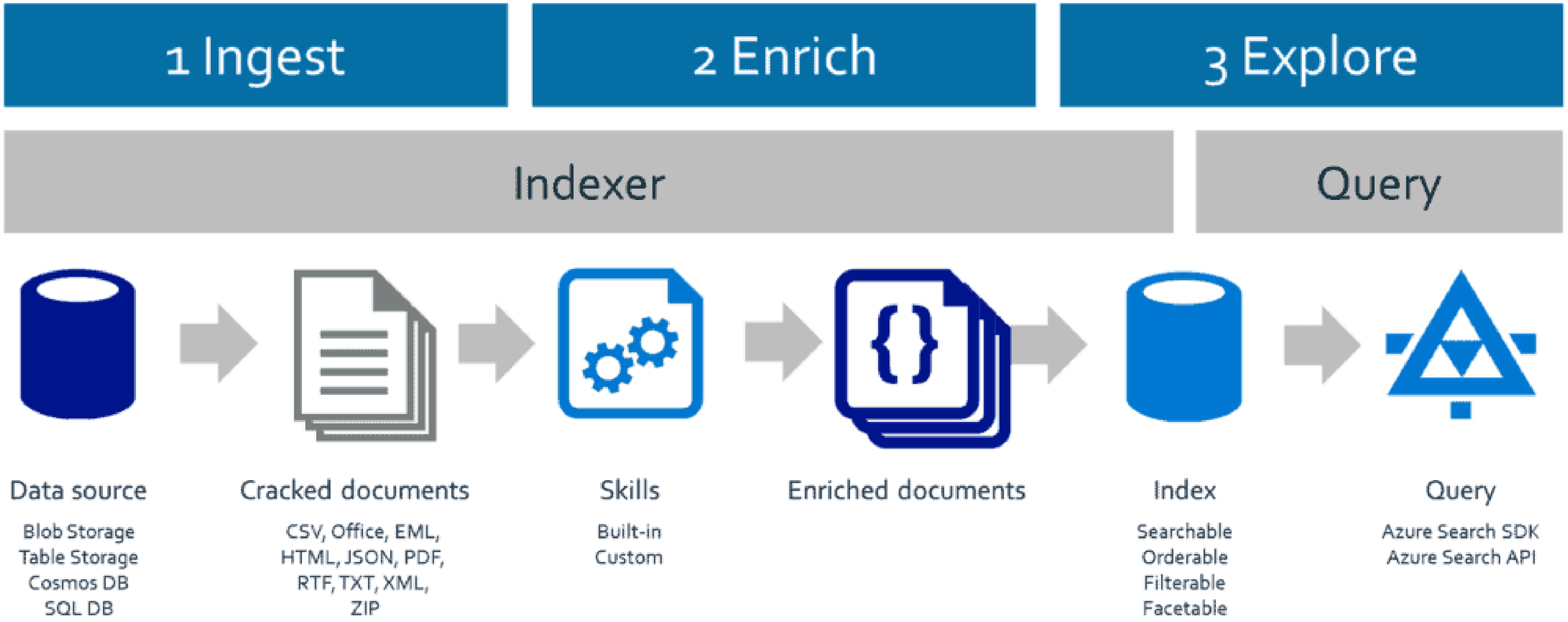
Search Index

- A search index is the core component of an Azure AI Search solution
- It is a physical data structure that stores your searchable content
- The index has a schema that defines the structure and properties of the data
- The schema includes:
  - Fields to store the content (e.g. title, description, images)
  - Data types for each field (e.g. string, number, date)
  - Attributes that control how the fields can be used (e.g. searchable, filterable, sortable)

Search Index



The Azure AI Search Indexing Pipeline



The Azure AI Search Indexing Pipeline

- Data Source: The original location of your data, such as Azure Blob Storage or a database
- Indexer: Automates the process of moving data from the data source into the search index
- Document Cracking: The indexer extracts content from various file formats (e.g., PDFs, Office documents)
- Enrichment: The indexer can apply AI-powered enrichment to the data, such as image analysis or language translation
- Push to Index: The processed data is then added to the search index

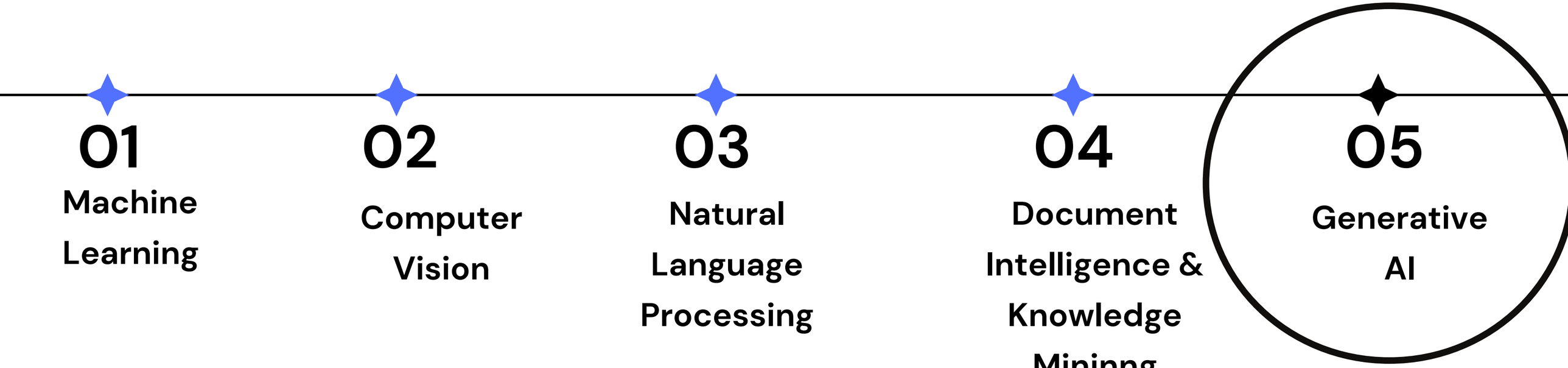
Conclusion

- Knowledge mining is the process of extracting valuable insights from large volumes of data
- Azure AI Search is a cloud-based search service that provides the tools and infrastructure to create powerful search solutions
- Azure AI Search Index: A structured collection of searchable documents, where each document is made up of fields that can be queried, filtered, and retrieved based on specified criteria.
- Indexing Pipeline: The automated process of importing, transforming, and loading data into a search index

Knowledge Mining Hands-On

- Fourth Coffee, a national coffee chain, wants to search for insights about costumer experiences
- The customer insights are in .doc format
- You decide to build an Azure AI Search index using data extracted from customer reviews.
  - Create an Azure AI Search resource (for searching)
  - Create an Azure AI services resource (for AI enrichment like nlp)
  - Azure Storage (for storing the costumer experiences)

Generative AI



Check your knowledge on document intelligence part

What is Generative AI?

- Generative AI is a type of artificial intelligence that can create original content, such as text, images, or code, based on input data.

- It takes user input (prompts) and generates:
  - Text (stories, emails, code)
  - Images (art, designs)
  - Audio (music, speech)

- Works by:
  - Learning patterns from massive amounts of training data
  - Understanding context and relationships
  - Generating new content based on learned patterns

What are language models?

- Generative AI applications are powered by language models, which are a specialized type of machine learning model that you can use to perform natural language processing (NLP) tasks, including:
    - Determining sentiment or otherwise classifying natural language text.
    - Summarizing text.
    - Comparing multiple text sources for semantic similarity.
    - Generating new natural language.

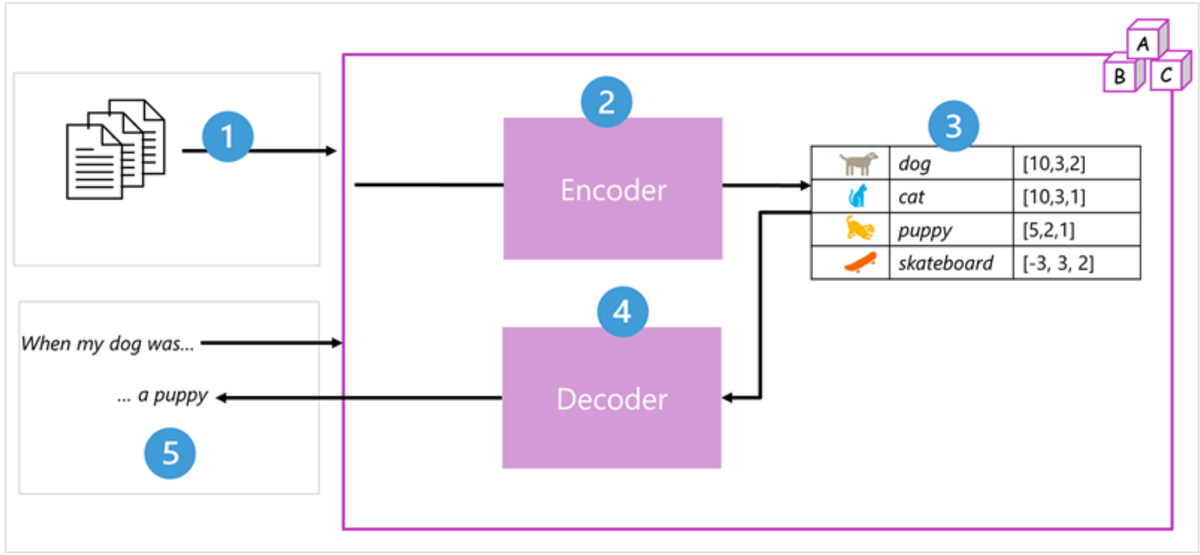
What are Transformer Models?

- Today's cutting-edge large language models are based on the transformer architecture

- Transformer models are trained with large volumes of text, enabling them to represent the semantic relationships between words and use those relationships to determine probable sequences of text that make sense.

Transformer Architecture Overview

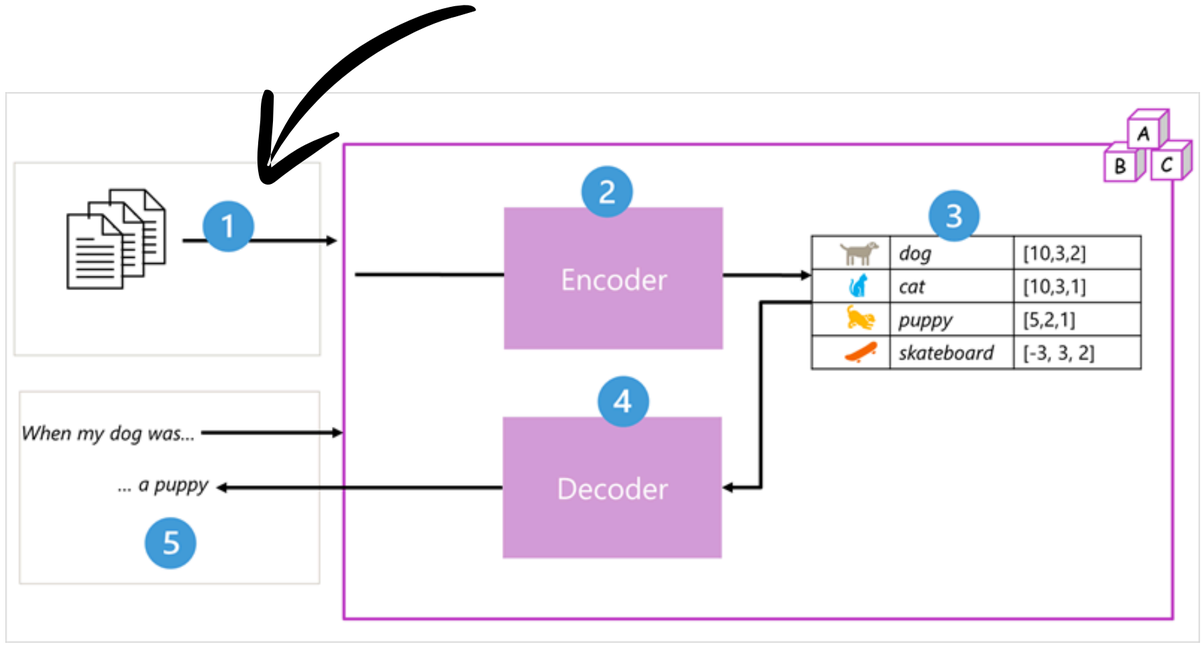
- Transformer models consist of two main components:
  - Encoder block: Processes the input sequence and generates semantic representations (embeddings)
  - Decoder block: Generates new language sequences based on the encoder's output



Tokenization

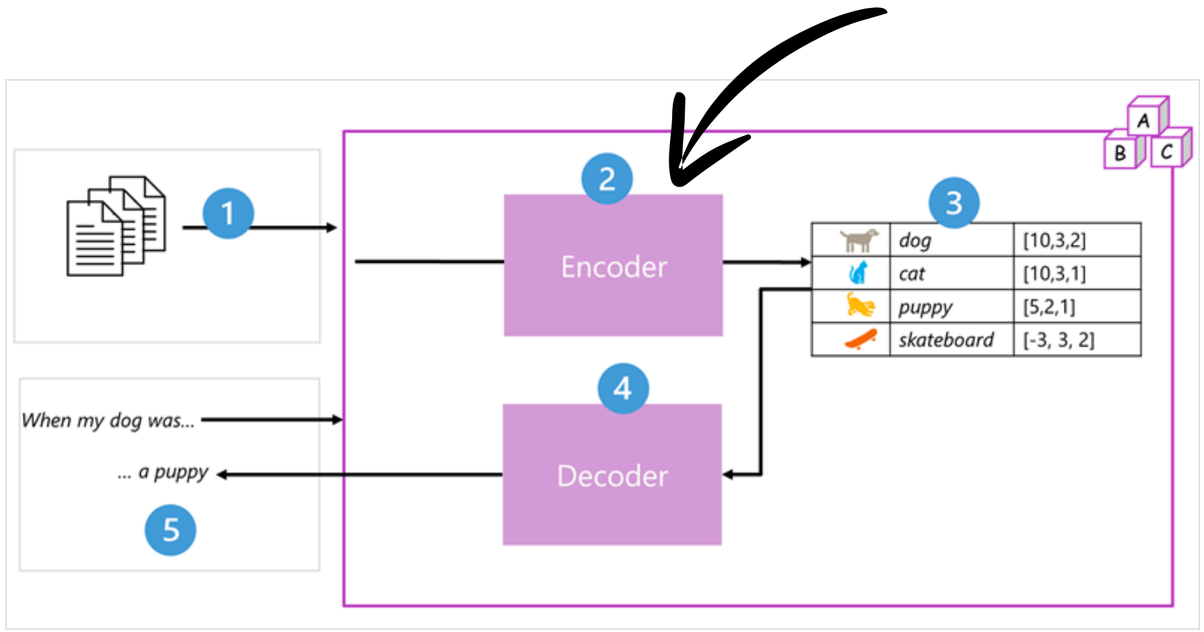
- The first step in training a transformer model is to break down the input text into discrete tokens, which represent individual words or sub-word units.

- (we already learned what and how it works, have a look at the NLP chapter)



Encoder

- The encoder block processes these token sequences using a technique called attention
- Attention mechanism focus on the most relevant parts of the input when generating new text.

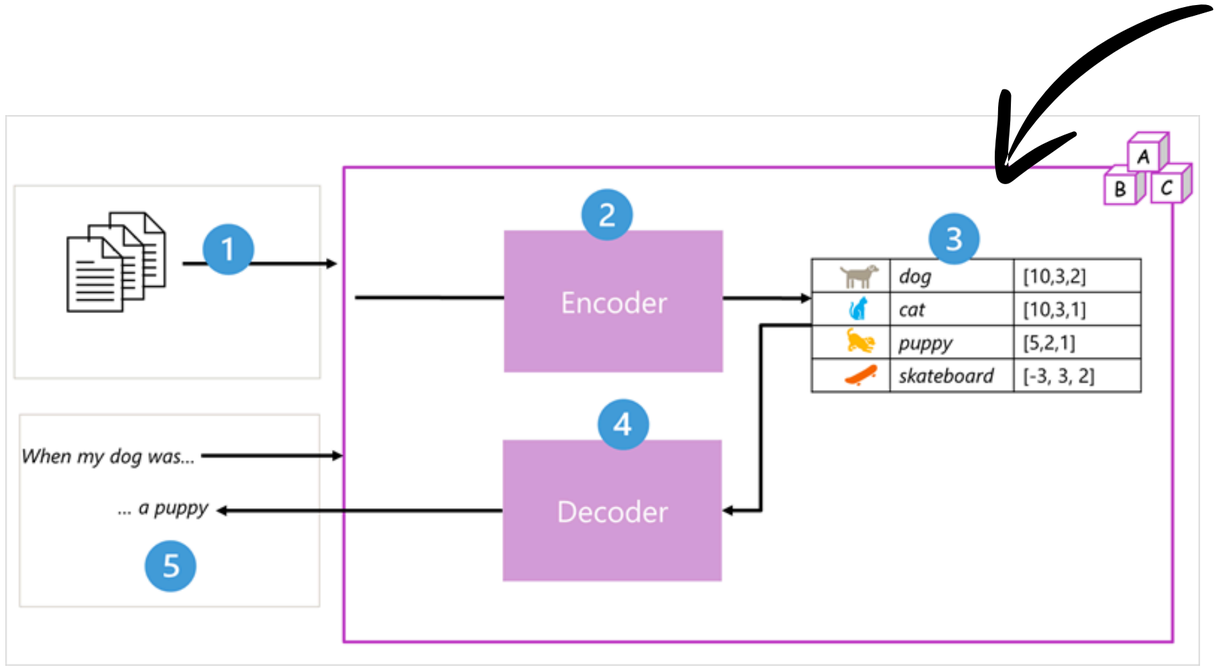


- Attention calculates the importance of each token in the input sequence
- This attention information is used to generate the next token in the output sequence
- Multi-head attention uses multiple attention calculations to capture different types of relationships

Embeddings

- The output from the encoder is a collection of vectors (multi-valued numeric arrays)

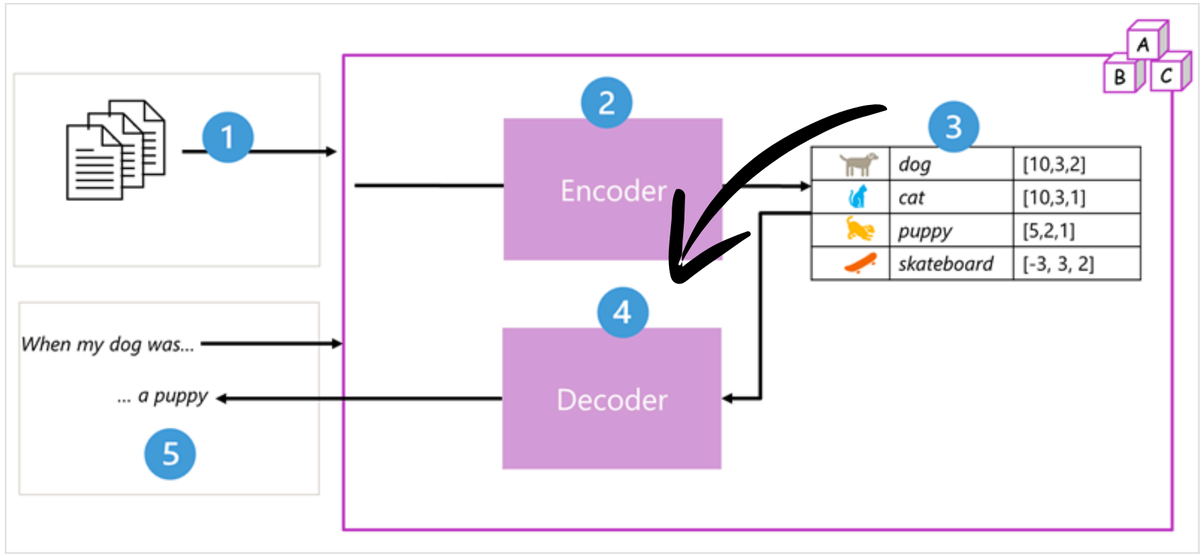
- (we already learned what and how it works, have a look at the NLP chapter)



Decoder

- The decoder block works on a new sequence of text tokens and uses the embeddings generated by the encoder to generate an appropriate natural language output.

- Given an input sequence like "When my dog was", the model can use the attention technique to analyze the input tokens and the semantic attributes encoded in the embeddings to predict an appropriate completion of the sentence, such as "a puppy."



Using language models

- Organizations and developers can train their own language models from scratch, but in most cases it's more practical to use an existing foundation model, and optionally fine-tune it with your own training data.

- On Microsoft Azure, you can find foundation models in the Azure OpenAI service and in the Model Catalog.

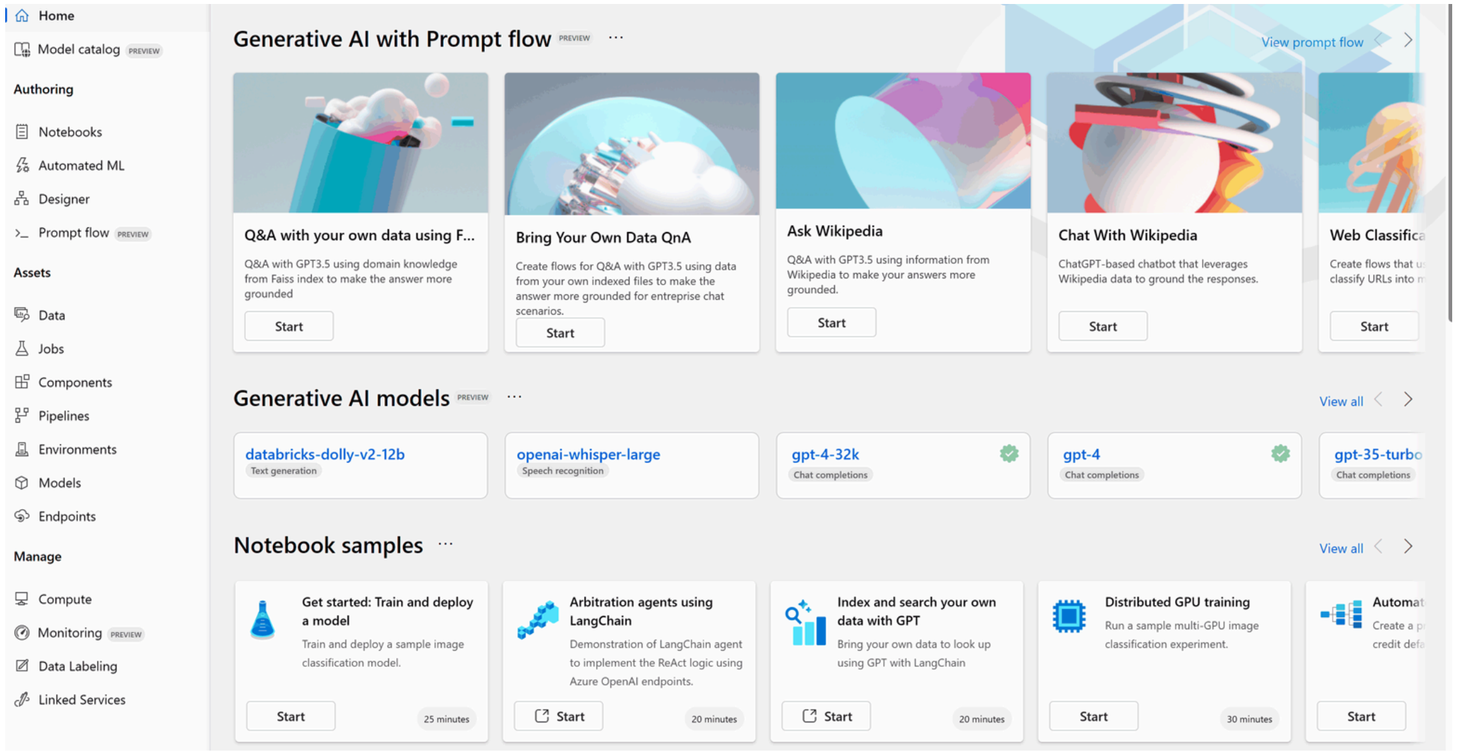
- The Model Catalog is a curated source of models for data scientists and developers using Azure AI Studio and Azure Machine Learning.

- In addition to the Azure OpenAI models, the model catalog includes the latest open-source models from Microsoft and multiple partners, including:
  - OpenAI
  - HuggingFace
  - Mistral
  - Meta and others.

Azure Model Catalog

Azure AI Model Catalog – Foundation Models | Microsoft Azure

Accelerate generative AI development with top-tier foundation models that are packaged for out-of-the-box use. Fine tune AI models using MaaS.

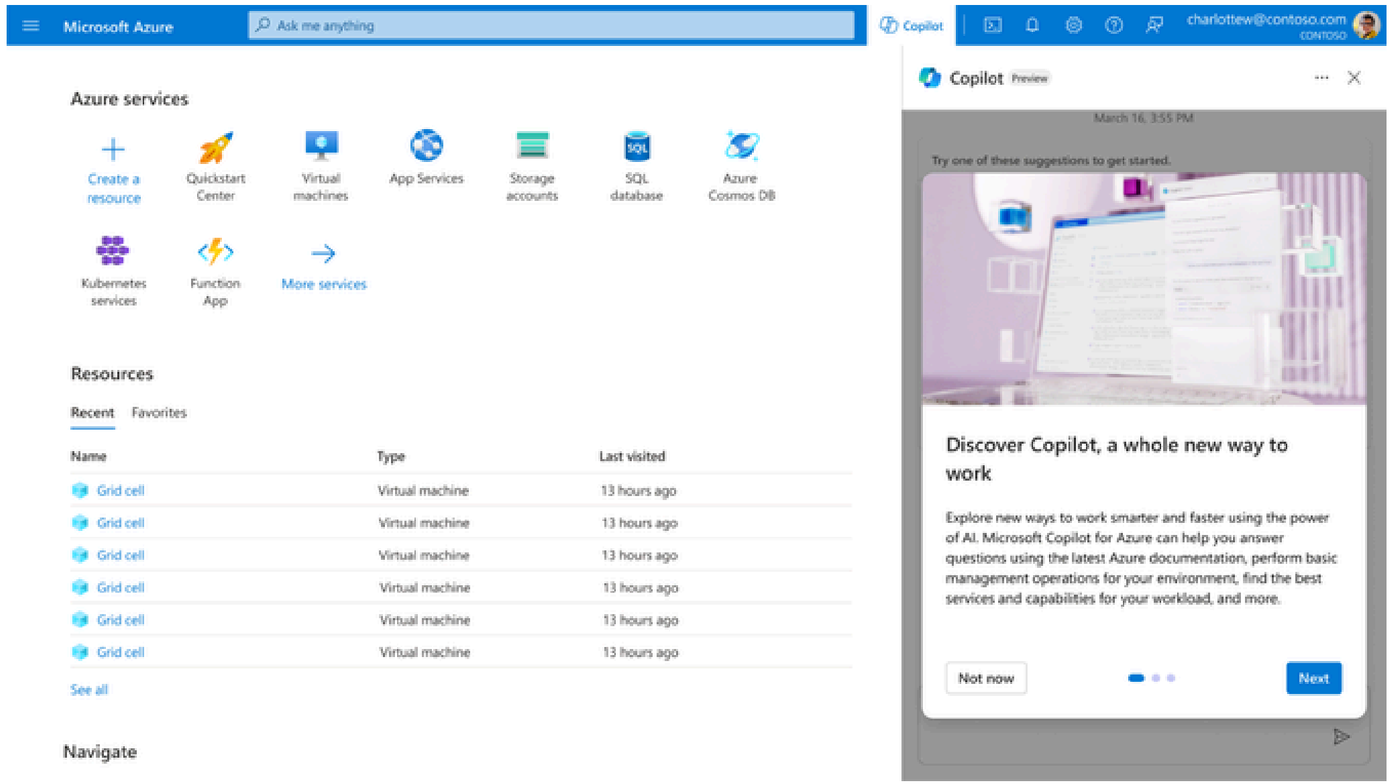


Large and small LM's

Large Language Models (LLMs)	Small Language Models (SLMs)
LLMs are trained with vast quantities of text that represents a wide range of general subject matter – typically by sourcing data from the Internet and other generally available publications.	SLMs are trained with smaller, more subject-focused datasets
When trained, LLMs have many billions (even trillions) of parameters (weights that can be applied to vector embeddings to calculate predicted token sequences).	Typically have fewer parameters than LLMs.
Able to exhibit comprehensive language generation capabilities in a wide range of conversational contexts.	This focused vocabulary makes them very effective in specific conversational topics, but less effective at more general language generation.
Their large size can impact their performance and make them difficult to deploy locally on devices and computers.	The smaller size of SLMs can provide more options for deployment, including local deployment to devices and on-premises computers; and makes them faster and easier to fine-tune.
Fine-tuning the model with additional data to customize its subject expertise can be time-consuming, and expensive in terms of the compute power required to perform the additional training.	Fine-tuning can potentially be less time-consuming and expensive.

What are Copilots?

- Copilots are generative AI assistants that are integrated into applications, often as chat interfaces. They provide contextualized support for common tasks in those applications, helping users be more productive and creative.



Levels of Copilot Adoption

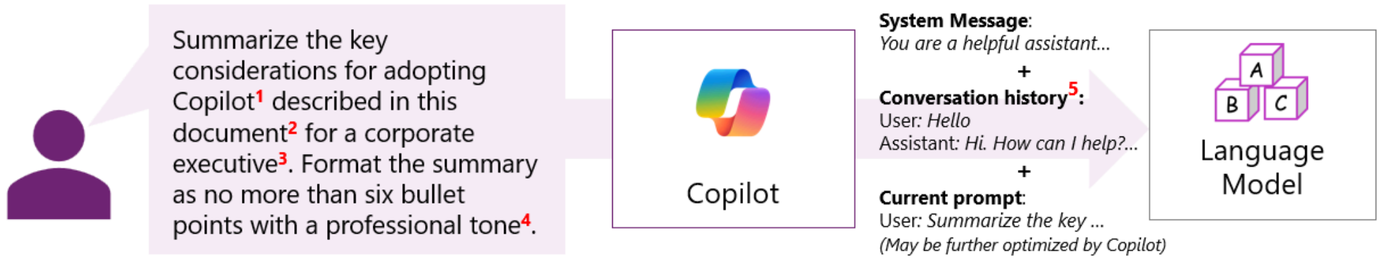
- There are three main ways organizations can adopt and use copilots:
  - Off-the-shelf use: Using pre-built copilots like Microsoft Copilot for Microsoft 365 to empower users and increase productivity.
  - Extending Microsoft Copilot: Customizing Microsoft Copilot to support your organization's specific business processes and tasks.
  - Custom development: Building completely new, custom copilots to integrate generative AI into your own business apps and services.

Copilot Use Cases

- Microsoft Copilot is integrated into a wide range of Microsoft applications, unlocking productivity, security, and AI capabilities:
    - Web browsing with AI: Use Copilot in the Edge browser, Bing search, and the standalone Copilot app.
    - AI assistance for information workers: Integrate Copilot into Microsoft 365 apps like Word, PowerPoint, and Outlook.
    - AI-powered business processes: Leverage Copilot in Dynamics 365 apps for customer service, sales, and supply chain.
    - AI-assisted data analytics: Use Copilot in Microsoft Fabric and Power BI.
    - Software development: GitHub Copilot helps developers be more productive.

Prompt Considerations for Copilots

- The quality of responses from copilots depends on the prompts users provide. To get the most useful completions:
  - Start with a specific goal for what you want the copilot to do.
  - Provide a source to ground the response in a specific scope of information.
  - Add context to maximize response appropriateness and relevance.
  - Set clear expectations for the response.
  - Iterate based on previous prompts and responses to refine the result.



Extending and Developing Copilots

- Microsoft provides two tools for customizing or building your own copilots:
  - Copilot Studio: A low-code platform for creating conversational AI experiences, with managed infrastructure.
  - Azure AI Studio: A platform-as-a-service for professional developers to build custom copilots with full control over the language model and integration.

Introduction to Azure AI Foundry

- A unified web portal for AI development on Microsoft Azure
- Brings together multiple AI services under one roof
    - Machine Learning Models
    - AI services
    - Prompt engineering
    - Custom code
- Designed for collaboration between developers and data scientists
- Simplifies the process of building AI solutions

Components of Azure AI Studio

- Azure Machine Learning capabilities
- Azure OpenAI service integration
- Azure AI Services for specialized tasks:
  - Speech recognition
  - Computer vision
  - Language processing
  - Document intelligence

The Building Blocks: AI Hubs

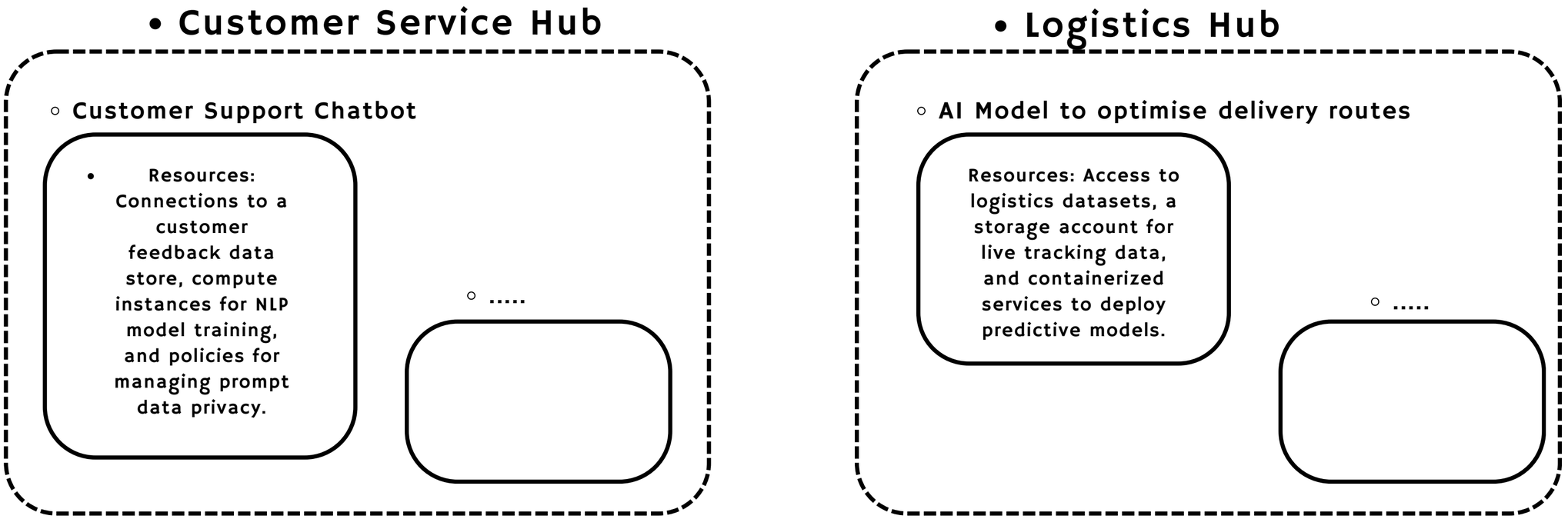
- An AI Hub is your collaborative workspace
- Functions as the foundation for all AI development projects
- Enables teams to:
  - Create and manage connections to resources, such as data stores, GitHub, Azure AI Search indexes, and others.
  - Manage computer power
  - Control access and permissions
  - Define policies to manage behavior, such as automatic compute shutdown.

The Building Blocks: Projects

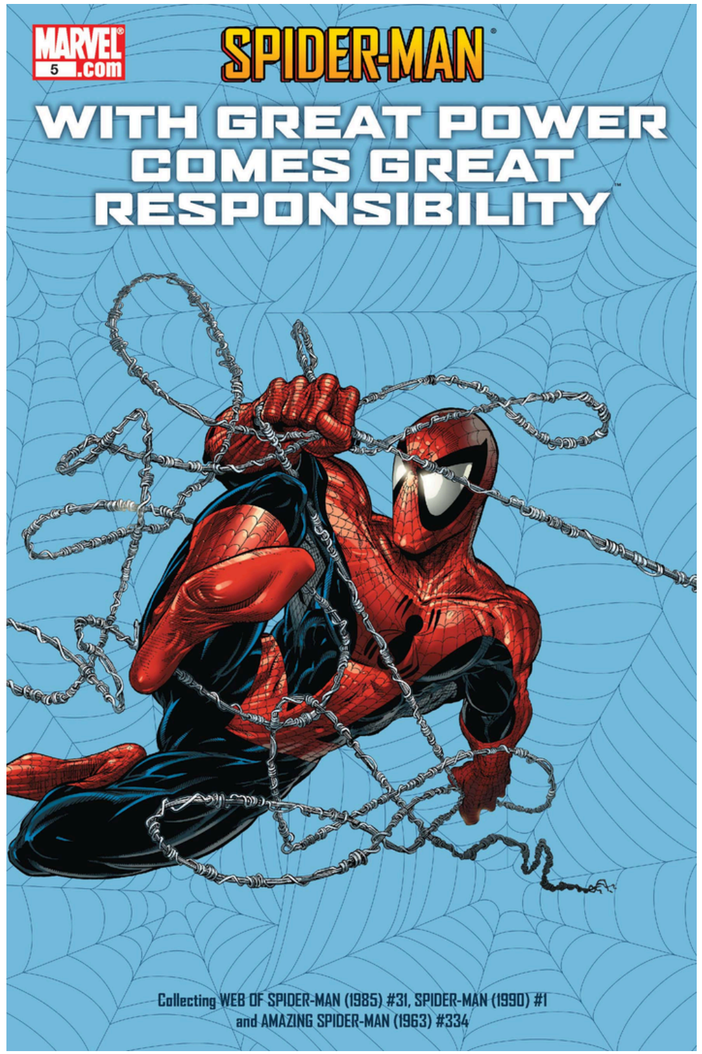
- Projects are specific workspaces within an AI Hub
- Use a project for:
  - Model deployments
  - Testing environments
  - Use prompt flow to define flows that combine models, prompts, and custom code.
  - Use Visual Studio Code in your browser to create custom code.
  - Deploy solutions as web apps and containerized services.

Example Use Case

- Large retail company that wants to leverage AI solutions across multiple departments
- Customer Service and Logistics. Each department has unique data, goals, and compliance requirements, so maintaining separate AI hubs can ensure tailored solutions and resource isolation.



Responsible generative AI



- Generative AI solutions are incredibly powerful tools, but they require well-designed safety mechanisms to be used responsibly

- Microsoft has developed a four-stage process to ensure responsible AI development

(These stages correspond closely to the functions in the NIST AI Risk Management Framework.)

- The goal is to identify, measure, and address potential problems before they affect users

The Four Stages of Responsible AI

- Stage 1: Identify potential harms
- Stage 2: Measure the presence of these harms
- Stage 3: Mitigate the risks at multiple layers
- Stage 4: Operate the solution responsibly

Stage 1: Identifying Potential Harms

- The first stage in a responsible generative AI process is to identify the potential harms that could affect your planned solution. There are four steps in this stage, as shown here:

1. Identify potential harms
2. Prioritize identified harms
3. Test and verify the prioritized harms
4. Document and share the verified harms

Identifying Potential Harms

Step I: Identify potential harms

- Look for possible negative outcomes like:
  - Generation of offensive or discriminatory content
  - Creation of factually incorrect information
  - Support for unethical or illegal activities

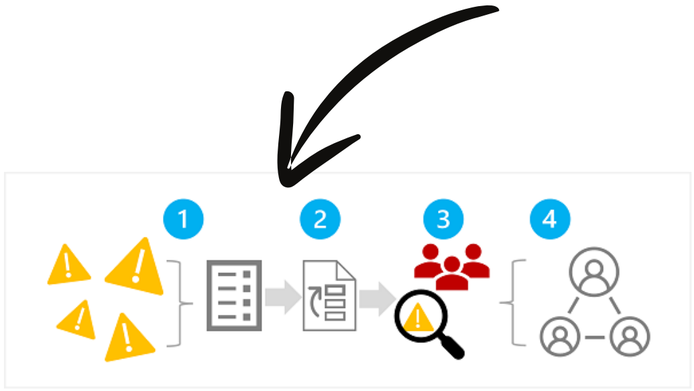
- Review existing documentation and guidelines

  For example, the Azure OpenAI Service includes a transparency note; which you can use to understand specific considerations related to the service and the models it includes

Identifying Potential Harms

Step 2: Prioritize the harms

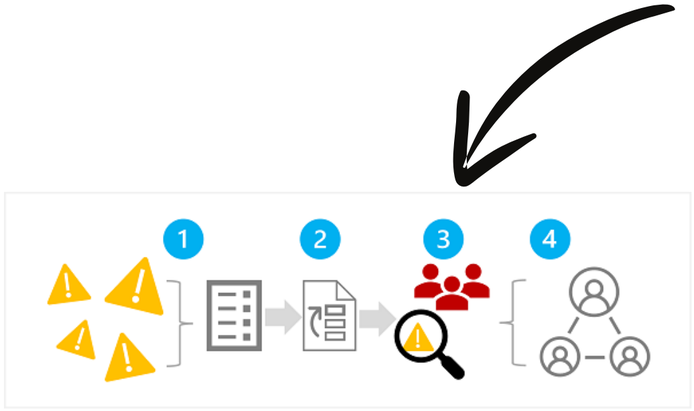
- For each potential harm you have identified, look at:
  - Likelihood of occurrence
  - Impact if it occurs
- Focus on high-priority risks first



Identifying Potential Harms

Step 3: Test and verify the presence of harms

- Use the “red team” testing approaches
- Deliberately probe for weaknesses
- Document test results and findings
- Identify previously unknown risks
- Update safety measures based on results



Identifying Potential Harms

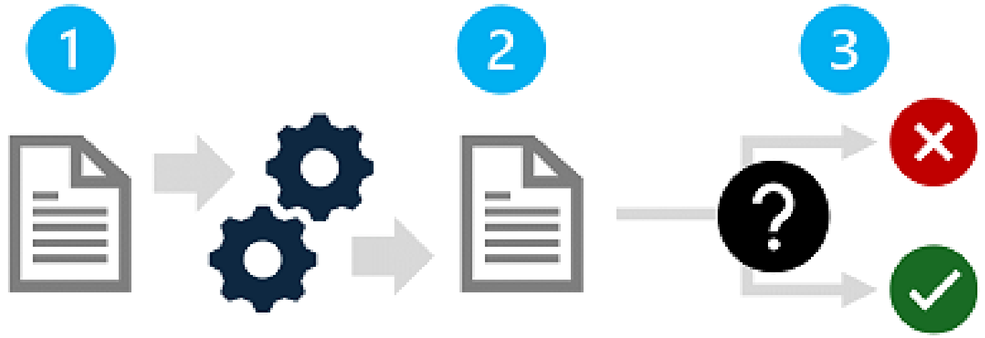
Step 4: Documentation and Sharing

- Create detailed documentation of:
  - Identified harms
  - Test results and evidence
  - Mitigation strategies

- Share findings with stakeholders
- Maintain and update documentation regularly

Measure potential harms

- Step 1: Prepare diverse test prompts
- Step 2: Generate outputs from these prompts
- Step 3: Evaluate outputs using pre-defined criteria



- Each step must be systematic and documented

Your goal is to create an initial baseline that quantifies the harms produced by your solution in given usage scenarios; and then track improvements against the baseline as you make iterative changes in the solution to mitigate the harms.

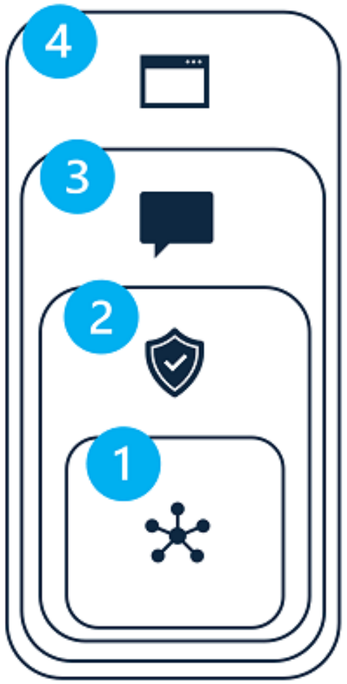
Testing Approaches: Manual vs Automated

- Manual Testing:
  - Initial small-scale evaluation
  - Ensures consistency in evaluation criteria
  - Helps refine testing approach
- Automated Testing:
  - Handles larger volumes of tests
  - Uses classification models
  - More efficient for ongoing monitoring

Mitigate potential harms

- Mitigate Potential Harms with Layered Approach

1. Model Layer: Choose and fine-tune appropriate models.
2. Safety System Layer: Use filters, abuse detection, and alerts.
3. Metaprompt & Grounding Layer: Structure prompts with contextual data.
4. User Experience Layer: Restrict inputs and ensure transparent documentation.



Operating Responsible AI: Pre-Release Steps

- Compliance Reviews are essential before release:
    - Legal verification
    - Privacy assessment
    - Security checks
    - Accessibility evaluation
- All documentation must be thoroughly reviewed
- Multiple teams should be involved in the review process

Release Management Strategies

- Implement a phased delivery approach
- Create detailed plans for:
    - Incident response
    - System rollback capabilities
    - User feedback collection
    - Performance monitoring
    - Implement immediate blocking capabilities for harmful content

Azure AI Content Safety Features

- Several Azure AI resources provide built-in analysis of the content they work with, including Language, Vision, and Azure OpenAI by using content filters.

- Features in Azure AI Content Safety include:

Feature	Functionality
Prompt shields	Scans for the risk of user input attacks on language models
Groundedness detection	Detects if text responses are grounded in a user's source content
Protected material detection	Scans for known copyrighted content
Custom categories	Define custom categories for any new or emerging patterns

Thank you so much!

Johannes Hayer

good luck for the exam!