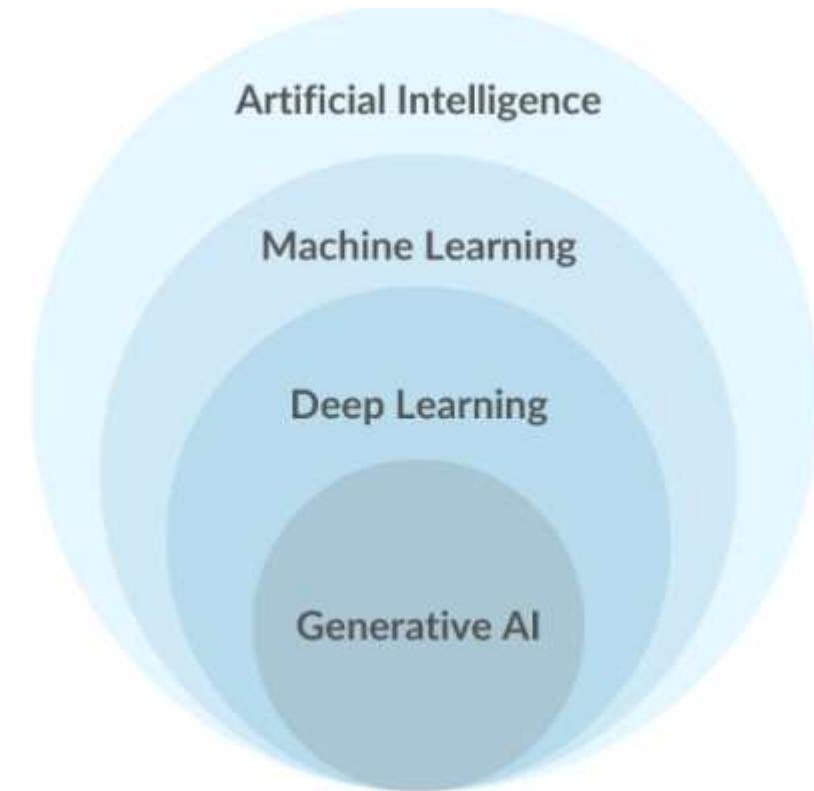


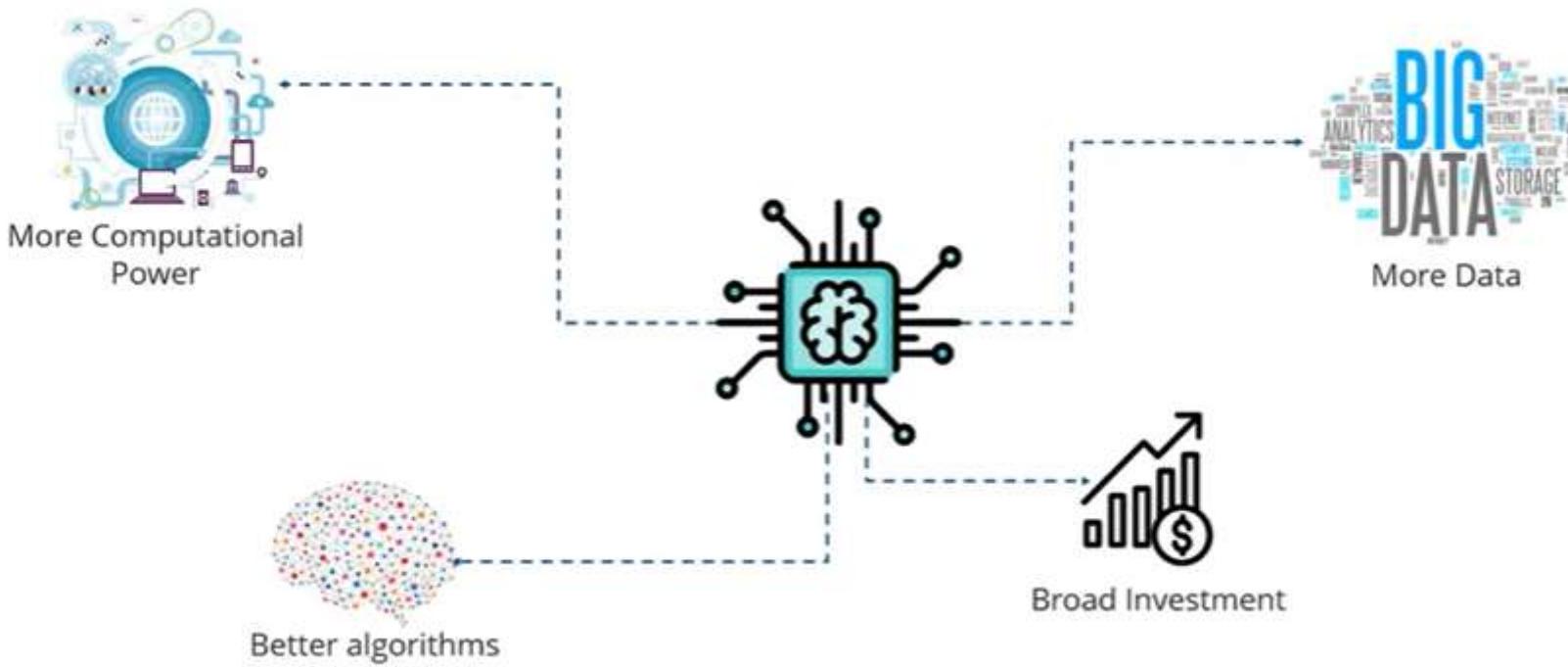
# Gen AI

# What is Generative AI?

Generative AI is a type of artificial intelligence technology that can produce various types of content including text, imagery, audio and synthetic data.



# AI



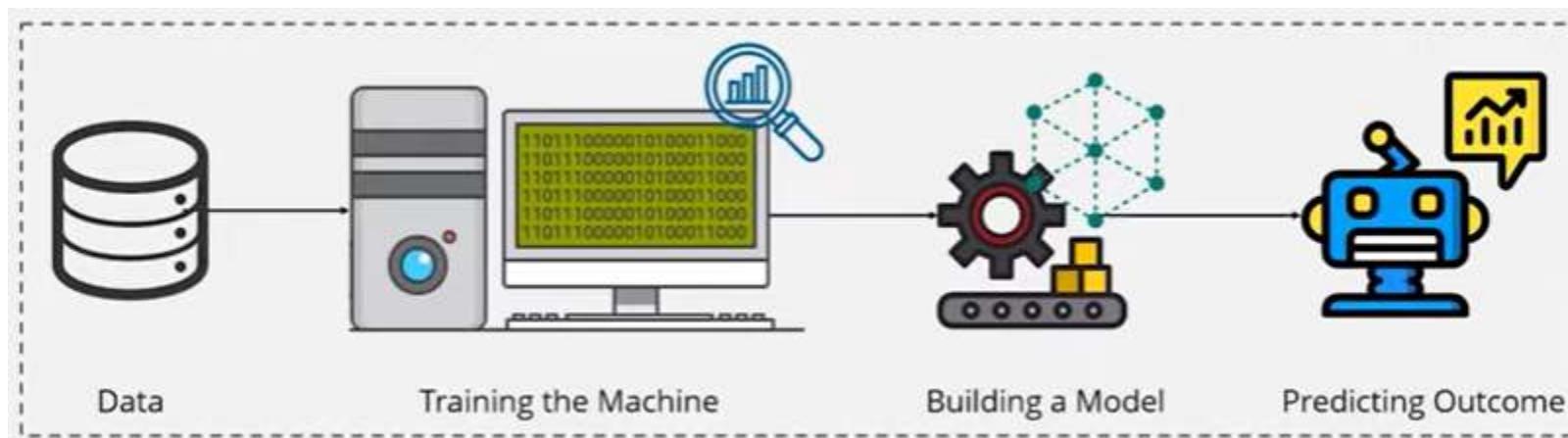
# Artificial Intelligence

- AI is a discipline .
- AI is a branch of computer science that deals with the creation of intelligent agents and our system can reason, learn and act autonomously.
- AI includes the theory and methods to build machines that think and act like humans.



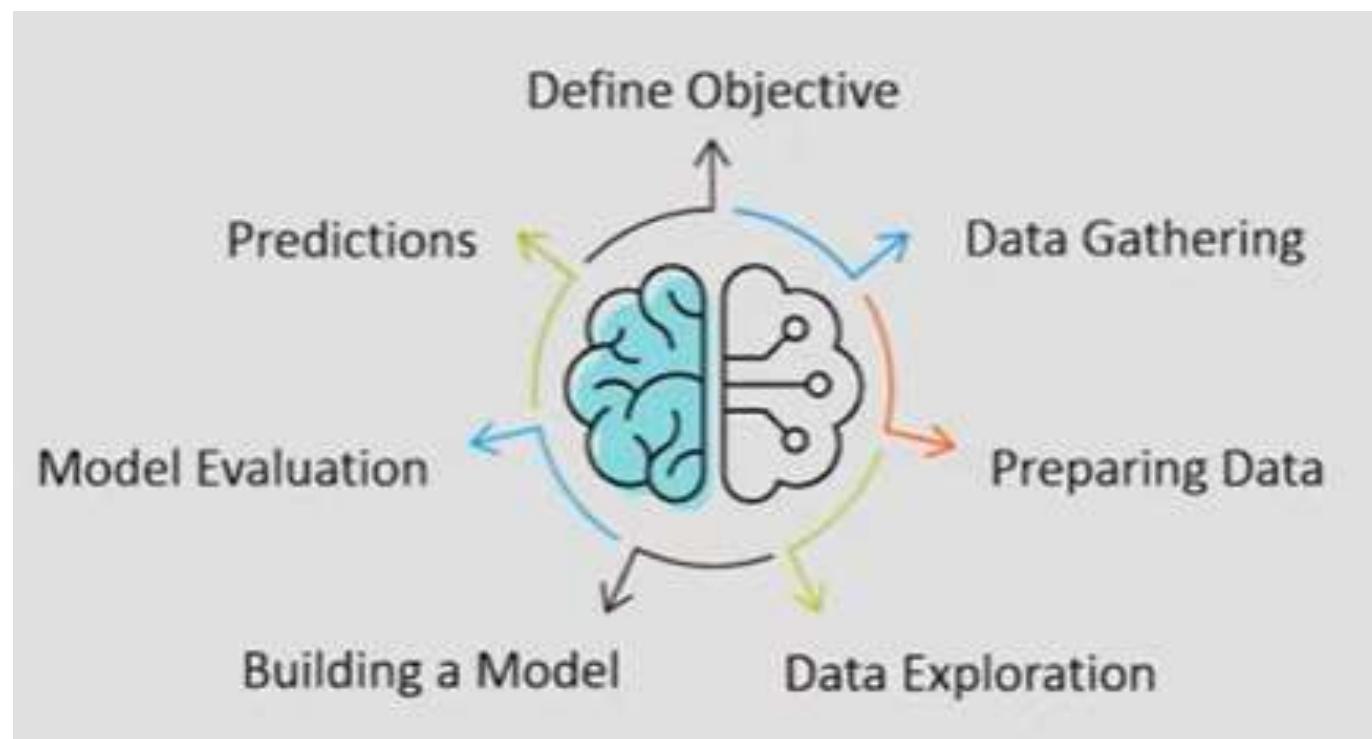
# Machine Learning

- Machine learning is a subfield of AI.
- It is a program or system that trains a model from input data.
- The trained model can make useful predictions from new never before seen data drawn from the same one used to train the model.
- Machine learning gives the computer the ability to learn without explicit programming.



# Machine Learning Process

The Machine Learning process involves building a Predictive model that can be used to find a solution for a Problem Statement



# 1. Define the objective of the Problem

- To predict the possibility of rain by studying the weather conditions
- What are we trying to predict?
- What are the target features?
- What is the input data?
- What kind of problem are we facing? Binary classification or Clustering?

## 2. Data Gathering

Data will be collected such as

- weather conditions
- Humidity level
- Temperature
- Pressure
- Region

It can be collected manually or using web scraping

### 3. Data Preparation

- Data preparation is very important part
- It involves cleaning of the data
- Getting rid of inconsistency in data like missing values
- Getting rid of duplicate values
- Corrupted data

Otherwise it would lead to any wrongful computation

## 4. Data Analysis

It involves understanding the patterns and trend in the data.

At this stage all the useful insights are drawn and correlations between the variables are understood.

## 5. Building a Machine Learning Model

- A predictive model is built by using the Machine Learning Algorithms such as Linear Regression. Decision Trees etc
- Machine Learning model is built by using the training data set
- The model is the Machine Learning algorithm that predicts the output by using the data fed to it.

## 6. Model Evaluation and Optimization

- The efficiency of the model is evaluated and any further improvement in the model are implemented.
- Machine Learning model is evaluated by using the testing data set
- The accuracy of the model is calculated
- Further improvement in the model are done by using parameter tuning.

## 7. Predictions

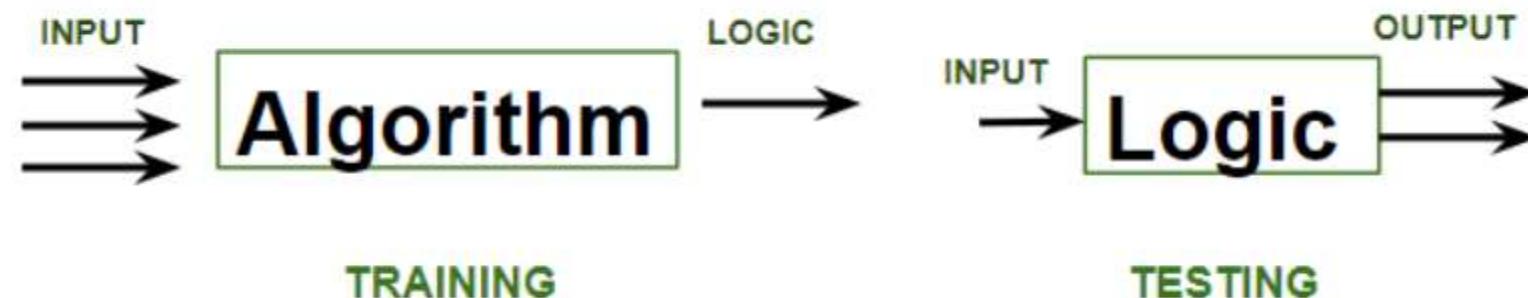
The final outcome is predicted after performing parameter tuning and improving the accuracy of the model.

# Machine Learning Types

- **Supervised Learning (Labelled data):** Labelled data comes with a tag like a name, a type or a number
- **Unsupervised Learning (Unlabelled data):** Unlabelled data does not come with any tag
- **Reinforcement learning:** Trial and error method

# Supervised Model

- Supervised machine learning is a type of machine learning where the model is trained on a labeled dataset (independent variables), meaning that each example in the dataset includes the input data as well as the correct output.
- The goal of supervised learning is to learn a mapping from inputs to outputs (dependent variables), so that the model can predict the correct output for new, unseen inputs.



# Supervised Model

- Supervised learning is where the model is trained on a labelled dataset. A labelled dataset is one that has both input and output parameters.
- The labeled dataset used in supervised learning consists of input features and corresponding output labels.
- The input features are the attributes or characteristics of the data that are used to make predictions, while the output labels are the desired outcomes or targets that the algorithm tries to predict.
- Examples of Supervised Learning: **Regression, Decision Tree, Random Forest, KNN, Logistic Regression**, etc.

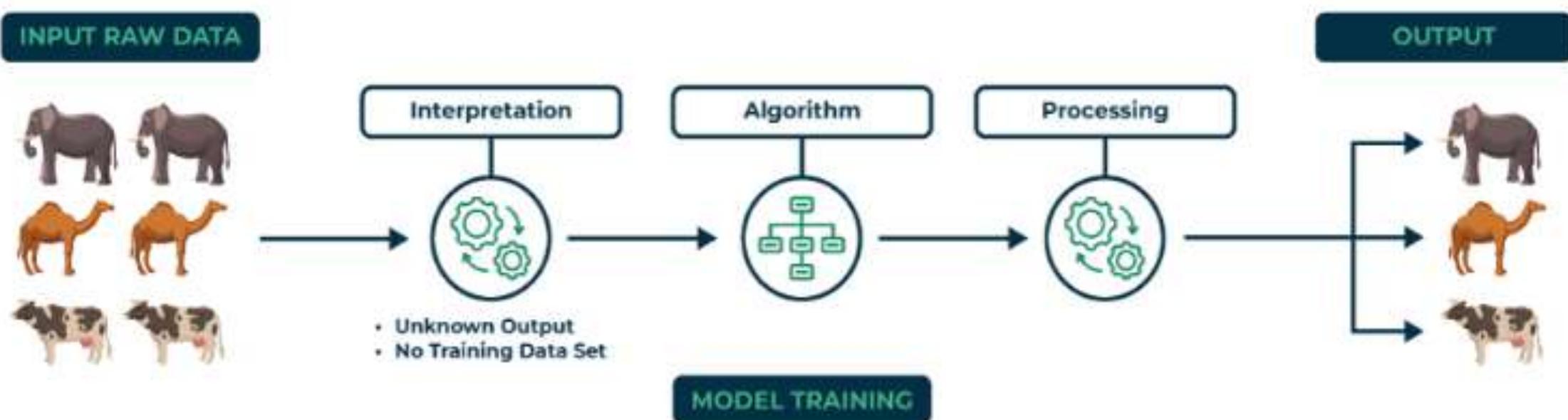
# Training the system

- While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and the rest as testing data.
- In training data, we feed input as well as output for 80% of data. The model learns from training data only. We use different machine learning algorithms to build our model. Learning means that the model will build some logic of its own.
- Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% of data that the model has never seen before, the model will predict some value and we will compare it with the actual output and calculate the accuracy.

# Unsupervised Model

- Unsupervised model is the learning that learns from unlabelled data.
- This means that the data does not have any pre-existing labels or categories. The goal of unsupervised learning is to discover patterns and relationships in the data without any explicit guidance.
- Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore the machine is restricted to find the hidden structure in unlabeled data by itself.
- Examples of Unsupervised Learning: **Apriori algorithm, K-means clustering.**

# Unsupervised Model



# Reinforcement Learning Algorithms

- Using this algorithm, the machine is trained to make specific decisions. The machine is exposed to an environment where it trains itself continually using trial and error.
- This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions.
- Example of Reinforcement Learning: **Markov Decision Process**
- **Example: Training an agent to play a game like Tic-Tac-Toe or a simple grid-world game where it learns the optimal strategy to win.**

# Regression vs Classification vs Clustering

## Regression

- Supervised Learning
- Output is a continuous quantity
- Main aim is to forecast or predict
- Eg: Predict stock market price
- Algorithm: Linear Regression

## Classification

- Supervised Learning
- Output is a categorical quantity
- Main aim is to compute the category of the data
- Eg: Classify emails as spam or non-spam
- Algorithm: Logistic Regression

## Clustering

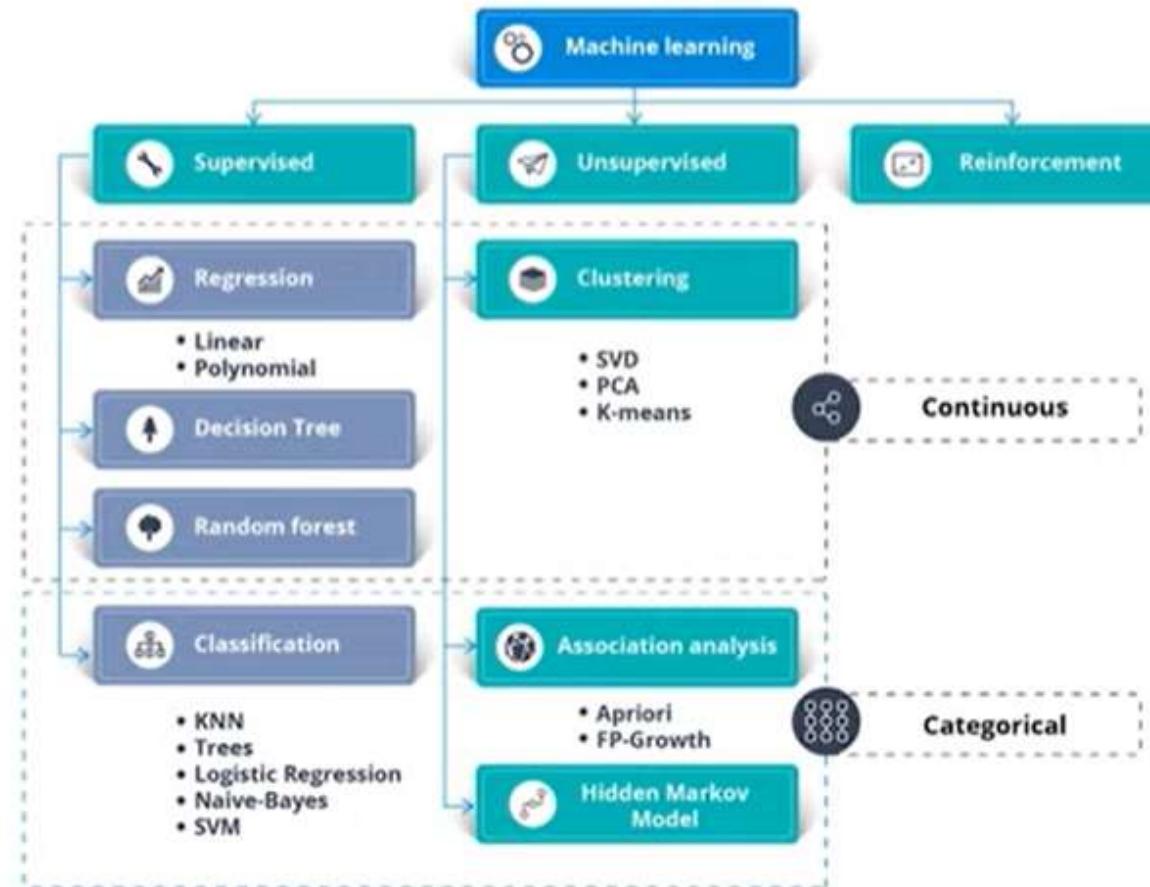
- Unsupervised Learning
- Assigns data points into clusters
- Main aim is to group similar items clusters
- Eg: Find all transactions which are fraudulent in nature
- Algorithm: K-means

# Top 10 Common Machine Learning Algorithms

These algorithms can be applied to almost any data problem:

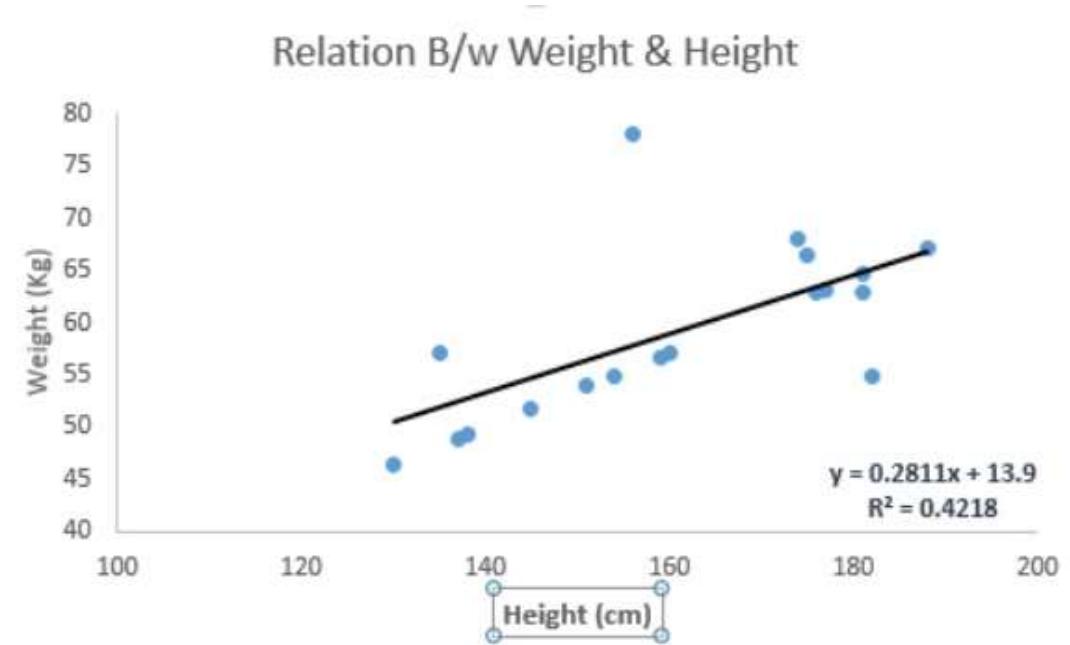
- Linear Regression
- Logistic Regression
- Decision Tree
- SVM
- Naive Bayes
- kNN
- K-Means
- Random Forest
- Dimensionality Reduction Algorithms
- Gradient Boosting algorithms
  - GBM
  - XGBoost
  - LightGBM
  - CatBoost

# Machine Learning Algorithms



# Linear Regression

- It is used to estimate real values (cost of houses, number of calls, total sales, etc.) based on a continuous variable(s).
- Here, we establish the relationship between independent and dependent variables by fitting the best line.
- It is represented by a linear equation  $Y = a * X + b$ .

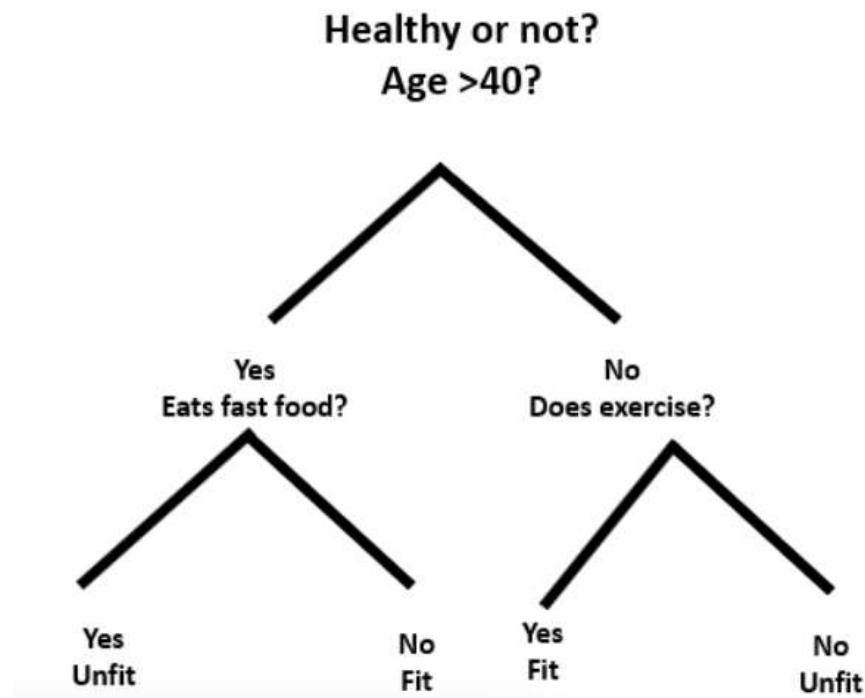


# Logistic Regression

- It is used to estimate discrete values ( **Binary values like 0/1, yes/no, true/false** ) based on a given set of independent variable(s).
- In simple words, it predicts the probability of the occurrence of an event by fitting data to a logistic function.
- Hence, it is also known as logic regression. Its output values lie between 0 and 1 (as expected).
- Let's say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it, or you don't.

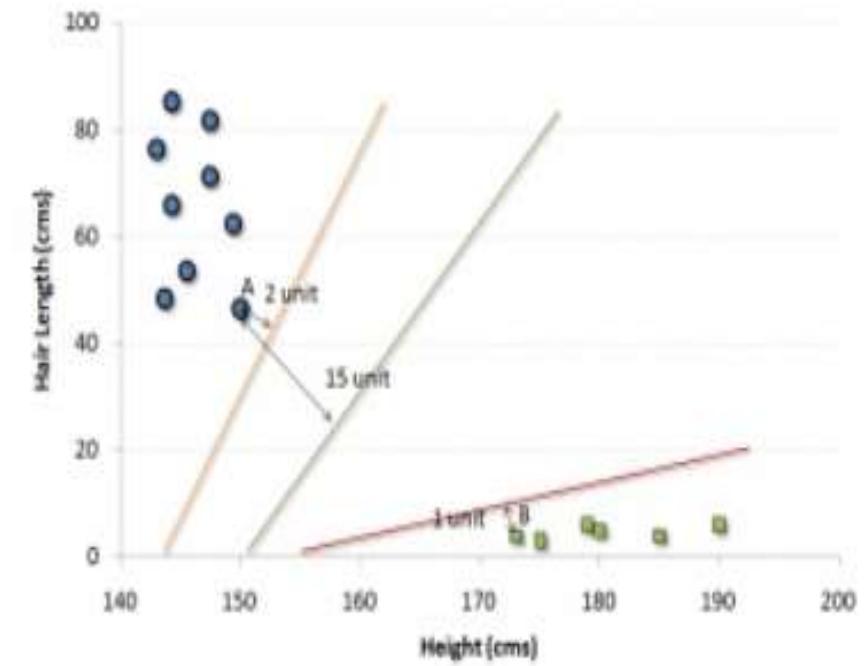
# Decision Tree

- It is a type of supervised learning algorithm that is mostly used for classification problems.
- Surprisingly, it works for both categorical and continuous dependent variables.
- In this algorithm, we split the population into two or more homogeneous sets.



# SVM (Support Vector Machine)

- In SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate.
- For example, if we only had two features like the Height and Hair length of an individual, we'd first plot these two variables in two-dimensional space where each point has two coordinates



# Naïve Bayes

- **Naive Bayes** classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- **For example**, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.
- The Naive Bayesian model is easy to build and particularly useful for very large data sets.

# kNN (k-Nearest Neighbors)

- It can be used for both **classification** and **regression** problems. However, it is more widely used in classification problems in the industry.
- K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors.
- If you want to learn about a person with whom you have no information, you might like to find out about his close friends and the circles he moves in and gain access to his/her information!

# K-Means

- It is a type of **unsupervised algorithm** which solves the clustering problem. Its procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters).
- Data points inside a cluster are homogeneous and heterogeneous to peer groups.
- Imagine you have a dataset containing information about customers of a supermarket, such as their age, income, and spending score. You want to segment these customers into different groups based on their similarities to target them with more personalized marketing strategies.

# Random Forest

- In Random Forest, we've got a **collection of decision trees** (also known as “Forest”). To classify a new object based on attributes, each tree gives a classification, and we say the tree “**votes**” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).
- Imagine you work for a telecom company and you want to predict which customers are likely to churn (cancel their subscription). You have a dataset containing information about customers, such as their usage patterns, account information, and customer service interactions.

# Dimensionality Reduction Algorithms

- Corporates/ Government Agencies/ Research organizations are not only coming up with new sources, but also they are capturing data in great detail.
- For example, **E-commerce** companies are capturing more details about customers like their demographics, web crawling history, what they like or dislike, purchase history, feedback, and many others to give them personalized attention more than your nearest grocery shopkeeper.

# Gradient Boosting Algorithms

- **GBM:** GBM is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. Boosting is actually an ensemble of learning algorithms that combines the prediction of several base estimators in order to improve robustness over a single estimator.
- **XGBoost:** It has an immensely high predictive power, making it the best choice for accuracy in events. It possesses both a linear model and the tree learning algorithm, making the algorithm almost 10x faster than existing gradient booster techniques.

# Gradient Boosting Algorithms

- **LightBGM:** LightGBM is a gradient-boosting framework that uses tree-based learning algorithms.
- It is efficient with Faster training speed and higher efficiency, Lower memory usage, Better accuracy, Parallel and GPU learning supported, Capable of handling large-scale data
- **Catboost:** CatBoost is one of open-sourced machine learning algorithms from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. The best part about CatBoost is that it does not require extensive data training like other ML models and can work on a variety of data formats, not undermining how robust it can be.

# Demo

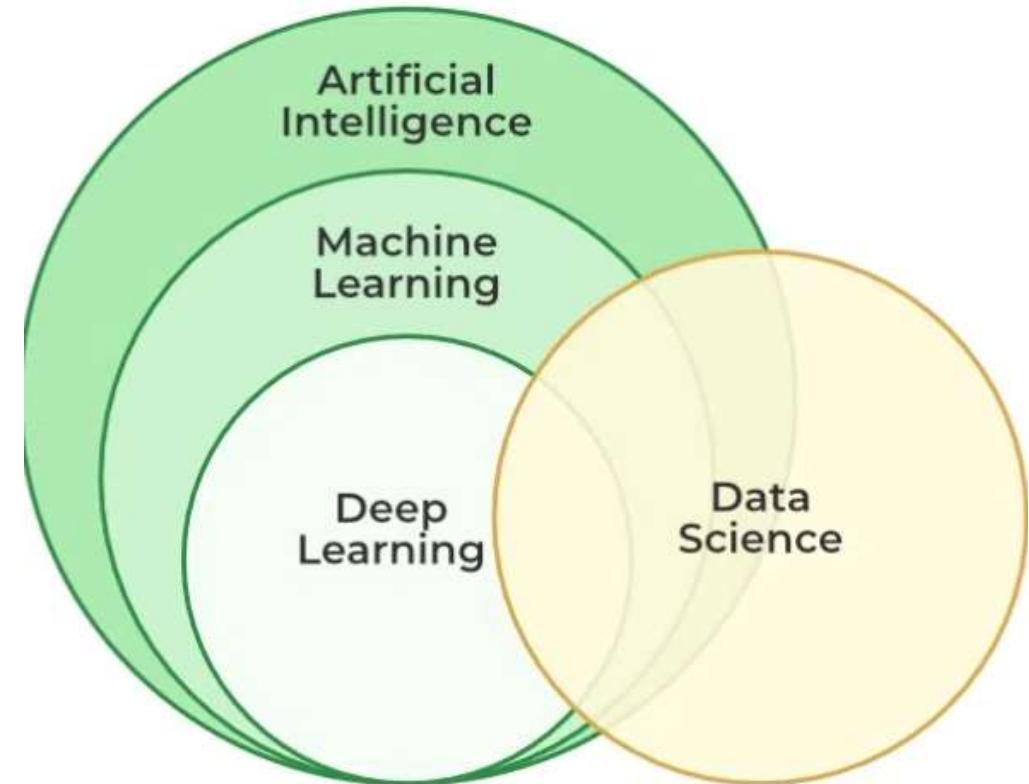
<https://github.com/CloudSihmar/machine-learning-rain-prediction.git>

# Deep Learning

- Deep learning is a branch of machine learning which is based on artificial **neural networks**.
- It is capable of learning **complex patterns** and **relationships** within data. In deep learning, we don't need to explicitly program everything.
- These neural networks are inspired by the structure and function of the **human brain's biological neurons**, and they are designed to learn from large amounts of data. Deep learning is a branch of machine learning which is based on artificial neural networks.

# Deep Learning

- Deep Learning is a subfield of Machine Learning that involves the use of neural networks to model and solve complex problems. Neural networks are modeled after the structure and function of the human brain and consist of layers of interconnected nodes that process and transform data.



# Deep Learning

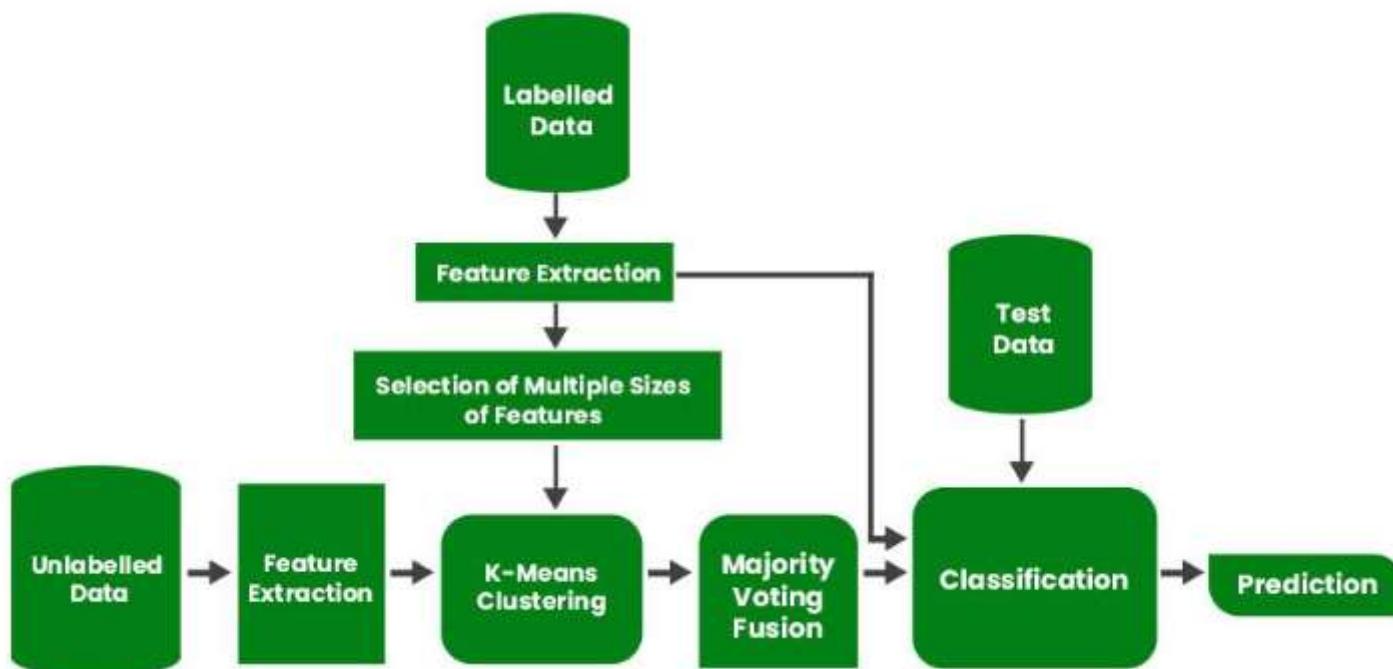
- Neural Network has **multiple layers** of interconnected nodes. These networks can learn complex representations of data by discovering hierarchical patterns and features in the data.
- Deep Learning algorithms can automatically learn and improve from data without the need for manual feature engineering.

# Deep Learning

- Neural Network can use both labeled and unlabeled data which is known as **semi-supervised** learning.
- Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning.
- It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model.
- The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning.

# Deep Learning

- Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it's too expensive or difficult to label all of it.



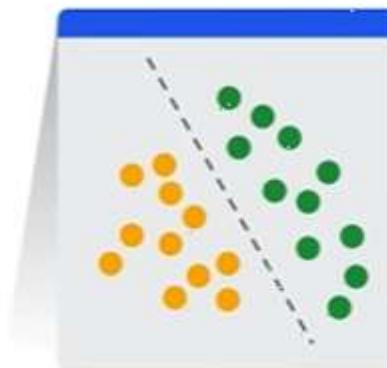
# Generative AI

- It is a subset of deep learning which means it used artificial neural networks which can process both labelled and unlabelled using supervised, unsupervised and semi-supervised methods.
- Large language models are also a subset of deep learning.

# Deep learning models

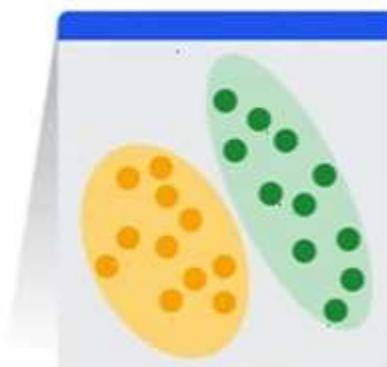
It is divided into two models

- Discriminative Model
- Generative Model



## Discriminative

- Used to classify or predict
- Typically trained on a dataset of labeled data
- Learns the relationship between the features of the data points and the labels



## Generative

- Generates new data that is similar to data it was trained on
- Understands distribution of data and how likely a given example is
- Predict next word in a sequence

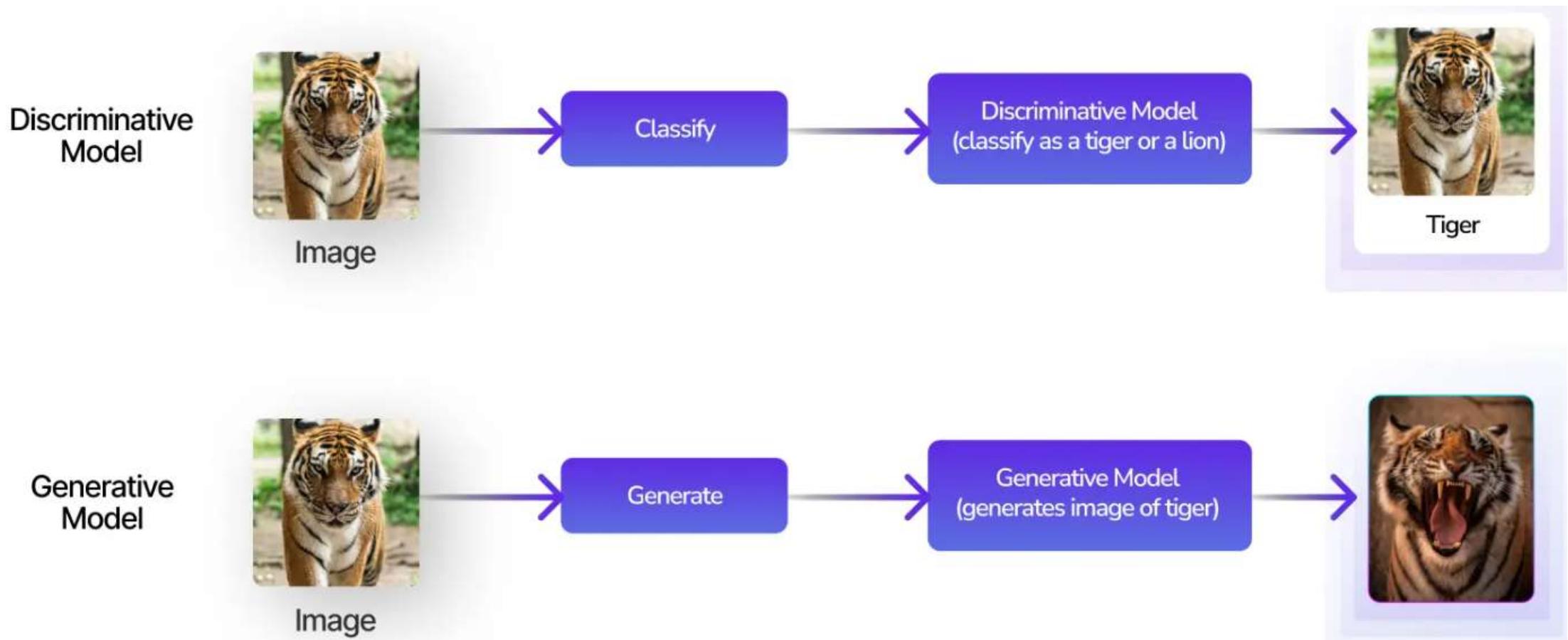
# Discriminative Model

- It is a type of model that is used to **classify** or **predict labels** for data points.
- Discriminative models are typically trained on the data set of labelled data points and they learn the relationship between the features of the data points and the labels.
- Once Discriminative model is trained, it can be used to predict the label for new data points.

# Generative Model

- Generative model generates new data instances based on a learned probability distribution of existing data.
- Generative models generate new contents

# Discriminative vs Generative Model



# Not Gen AI

If the output is

- Number
- Discrete
- Class
- Probability

# Is Gen AI

If the output is

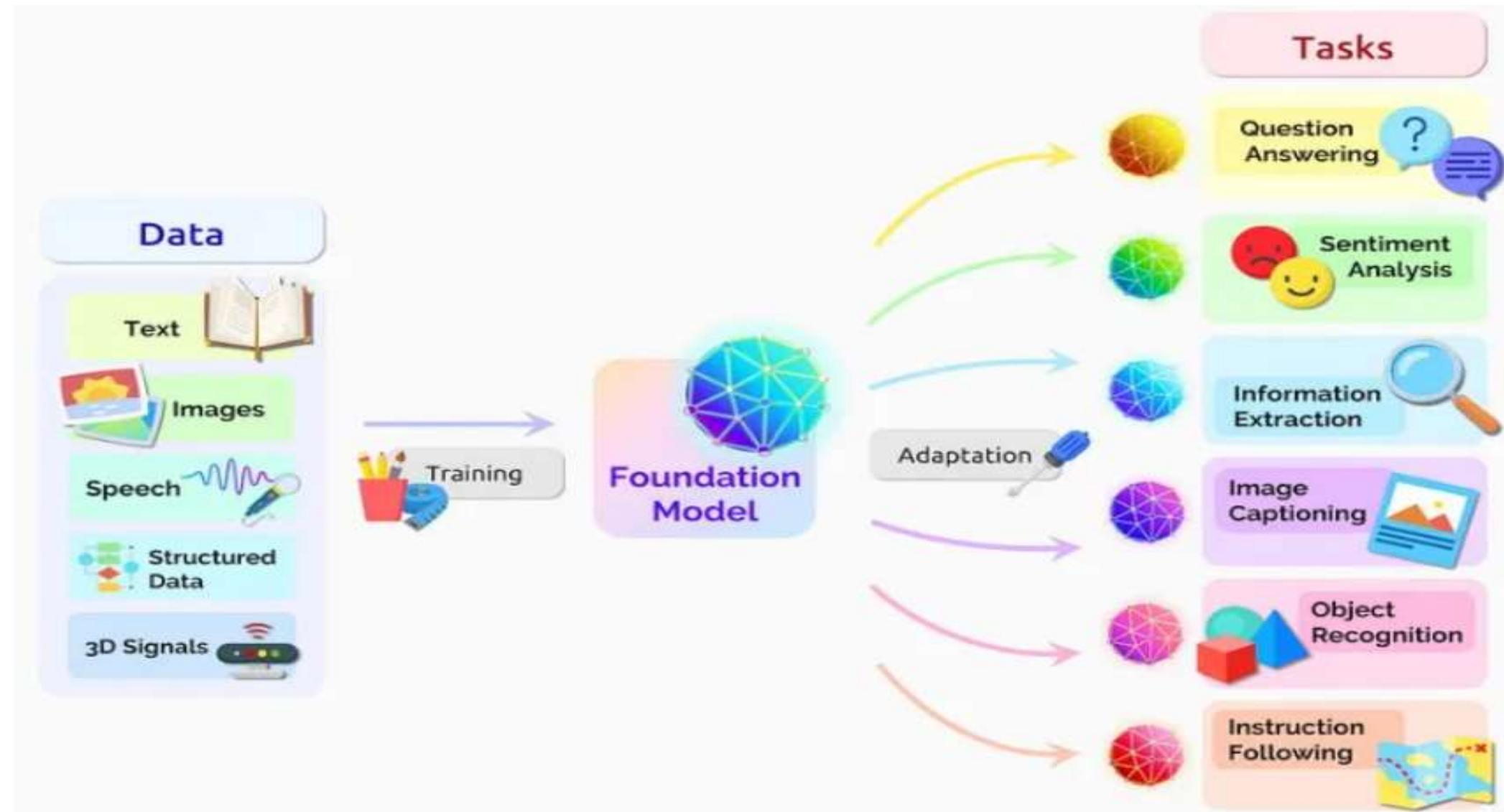
- Natural language
- Image
- Audio

# Gen AI

- Gen AI is a type of Artificial Intelligence that creates new content based on what it has learned from existing content.
- The process of learning from existing content is called training and results in the creation of a statistical model.
- When given a prompt, GEN AI uses a statistical model to predict what an expected response might be and this generates new content
- It learns the underlying structure of the data then generates new samples that are similar to the data it was trained on.

# Key Terminologies

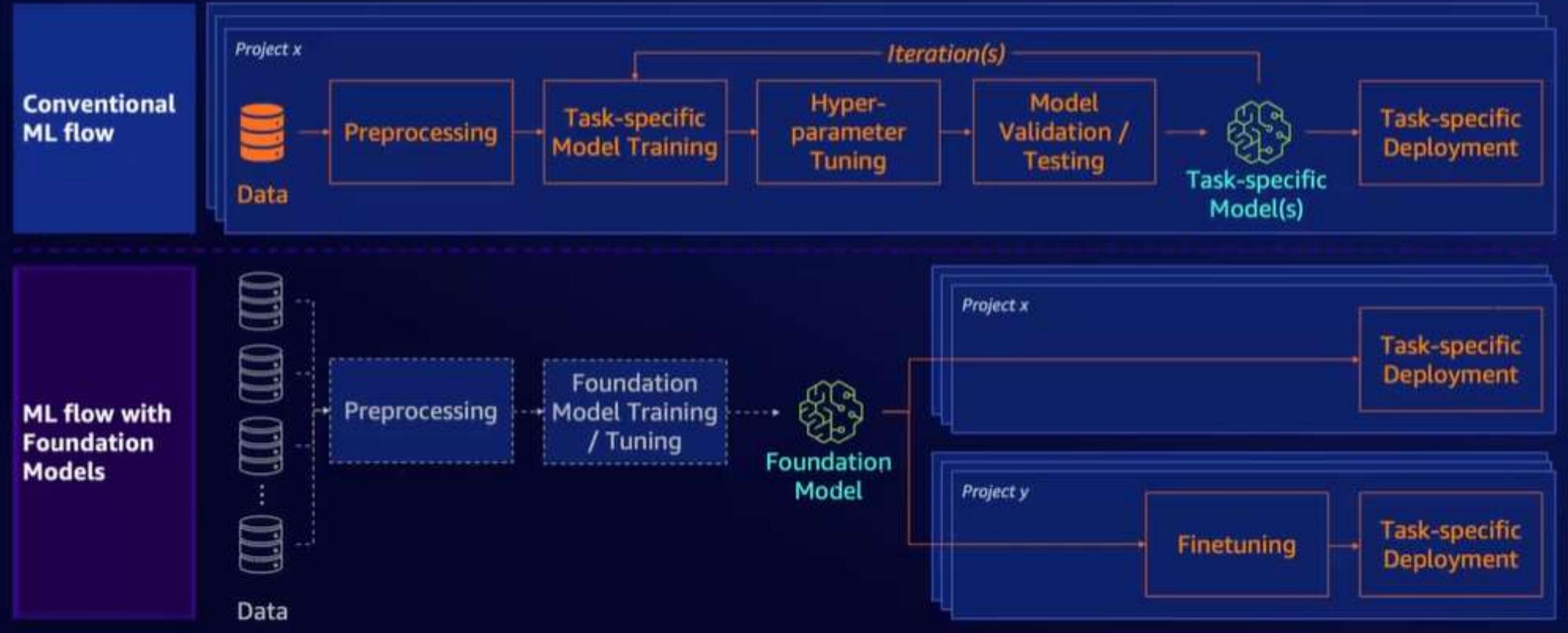
- Large Language Model
- Prompt Engineering
- Embeddings
- Fine Tuning



# The Generative AI Application Landscape



# What is a Foundation Model?



# 1. LLM

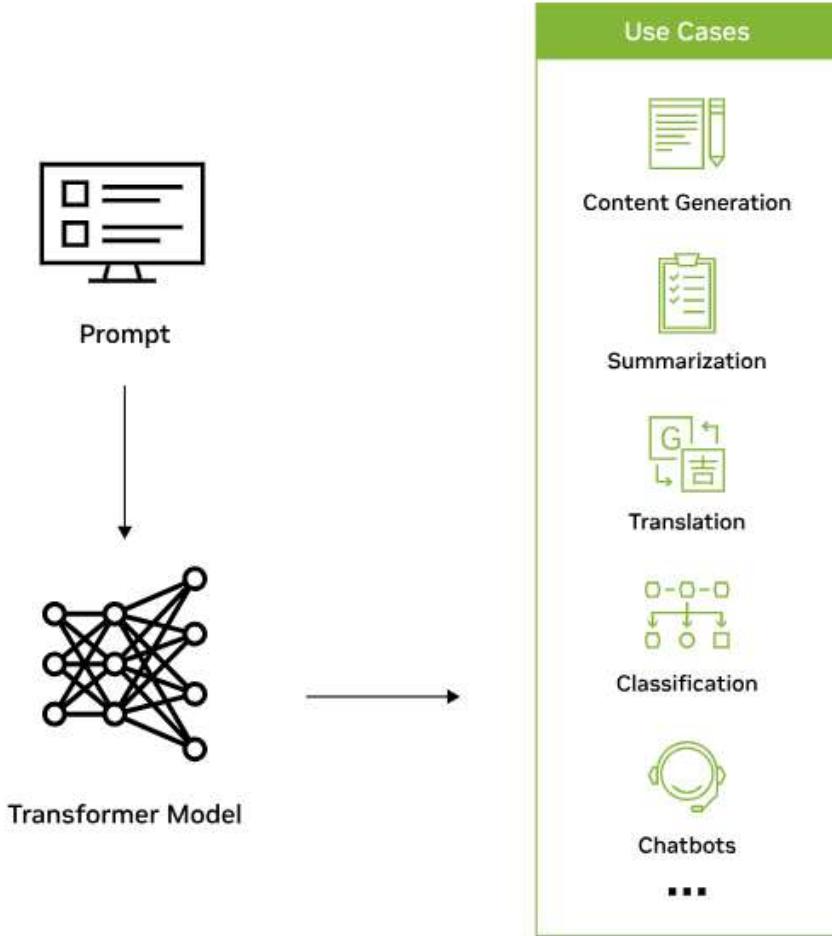
- Large Language models refer to large, general-purpose language models that can be **pre-trained** and then **fine-tuned** for specific purposes.

Large language models are trained to solve common language problems like

- Text classification
- Question Answering
- Document summarization
- Text generation
- Language translation

# LLM

- A large language model is a type of artificial intelligence algorithm that uses deep learning techniques and massively large data sets to understand, summarize, generate and predict new content.



# LLM

- Large language models use **transformer** models and are trained using **massive datasets** — hence, large. This enables them to recognize, translate, predict, or generate text or other content.
- Large language models are also referred to as **neural networks (NNs)**, which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.
- Their problem-solving capabilities can be applied to fields like **healthcare**, **finance**, and **entertainment** where large language models serve a variety of NLP applications, such as translation, chatbots, AI assistants, and so on.
- Large language models also have large numbers of parameters, which are akin to memories the model collects as it learns from training. Think of these parameters as the model's knowledge bank.

# LLM

- LLM is using **transformer model** behind the scene.
- **Transformer model** is based on the **attention** mechanism. The attention mechanism allows you to see the entire sentence (or even the paragraph) at once rather than one word at a time.
- This allows the transformer model to understand the context of a word better.

# Transformer Model

- A transformer model is the most common architecture of a large language model.
- It consists of an **encoder** and a **decoder**.
- A transformer model processes data by tokenizing the input, then simultaneously conducting mathematical equations to discover relationships between tokens. This enables the computer to see the patterns a human would see were it given the same query.
- Transformer models work with self-attention mechanisms, which enables the model to learn more quickly than traditional models like long short-term memory models.
- Self-attention is what enables the transformer model to consider different parts of the sequence, or the entire context of a sentence, to generate predictions.

# Large Language Models

- **Generic or raw language models** predict the next word based on the language in the training data. These language models perform information retrieval tasks.
- **Instruction-tuned language models** are trained to predict responses to the instructions given in the input. This allows them to perform sentiment analysis, or to generate text or code.
- **Dialog-tuned language models** are trained to have a dialog by predicting the next response. Think of chatbots or conversational AI.

## 2. Prompting techniques

Prompt engineering refers to the process of crafting specific prompts or input queries to guide the behavior of large language models (LLMs) during inference. This technique is particularly relevant for fine-tuning the output of LLMs to generate desired responses or outputs for specific tasks or applications.

# Methods to shape the model's response:

- **Zero-shot prompting** - This is a method where the LLM is given only a prompt that describes the task and no additional data. For example, if you want the LLM to answer a question, you just prompt "what is prompt design?".
- **One-shot prompting** - This is a method where the LLM is given a single example of the task that it is being asked to perform. For example, if you want the LLM to write a poem, you might give it a single example poem.
- **Few-shot prompting** - This is a method where the LLM is given a small number of examples of the task that it is being asked to perform. For example, if you want the LLM to write a news article, you might give it a few news articles to read.

- 
- 1 Clear Instructions
  - 2 Adopt a persona
  - 3 Specify the format
  - 4 Avoid leading the answer
  - 5 Limit the scope

# 1. SFR



You

What were the biggest AI breakthroughs of 2023? Give them as 5 bullet points, one sentence long each, for a layperson

## Stylize

Simple  
Technical  
Creative  
Formal  
Informal  
Persuasive  
Like a pirate

## Format

Bullet points  
Summary  
Headings & subheadings  
Blog article  
Tweet  
Table  
Code snippet

## Restrict

One sentence  
2-3 sentences  
One paragraph  
Extremely detailed  
Only use academic papers  
No sources older than 2020

## 2. Role prompting

HU

You

You are an elementary school teacher speaking to your students. Explain string theory in one paragraph

< 2 / 2 >



ChatGPT

String theory is a fascinating idea in physics that suggests everything in the universe is made up of tiny, vibrating strands called strings. Imagine a guitar string: when you pluck it, it vibrates and makes a sound. Similarly, these tiny strings in string theory vibrate in different ways, and their vibrations determine the kind of particle they represent, like an electron or a photon. This theory is exciting because it tries to explain

- 3.Zero shot prompting & Few shot prompts
4. Chain of thoughts.

I'm providing a sample of my writing to help you understand my style. Please analyze this sample for tone, structure, and language u

0

↑

format.

0

Write a blog post on the bullet points we created about artificial intelligence, write|

↑

# 3. Embeddings

- **Embeddings**, in the context of natural language processing (NLP), refer to the mapping of words or phrases from a high-dimensional space (such as a vocabulary) to a lower-dimensional vector space, where words with similar meanings or contexts are closer together. These vector representations capture semantic relationships and contextual information about words, enabling machine learning models to effectively process and understand natural language.
- Embeddings are representations of values or objects **like text, images, and audio** that are designed to be consumed by machine learning models and semantic search algorithms.
- They translate objects like these into a **mathematical form** according to the factors or traits each one may or may not have, and the categories they belong to.

**Embeddings play a crucial role in NLP tasks such as:**

- text classification
- sentiment analysis
- machine translation
- named entity recognition

# Embedding

- Embedding is the process of creating vectors using deep learning. An "embedding" is the output of this process — in other words, the vector that is created by a deep learning model for the purpose of similarity searches by that model.
- Neural networks are made of several layers: an input layer, an output layer, and any number of "hidden" layers in between.
- The creation of embeddings is a hidden layer. It usually takes place before additional layers process the input.



# How are embeddings used in large language models (LLMs)?

- For large language models (LLMs), such as the models used for AI tools like **ChatGPT**, embedding is taken a step further.
- The context of every word becomes an embedding, in addition to the word itself. The meanings of entire sentences, paragraphs, and articles can be searched and analyzed.
- Although this takes quite a bit of computational power, the context for queries can be stored as embeddings, saving time and compute power for future queries.

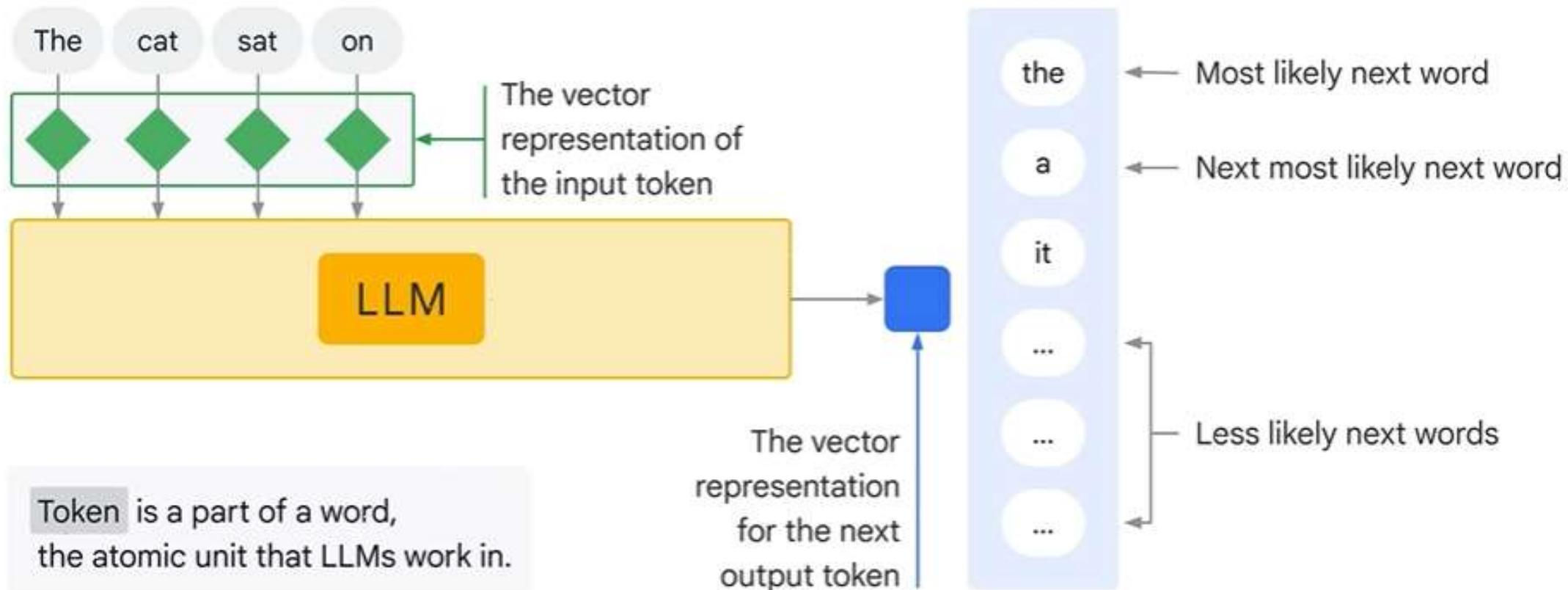


# food

```
[  
    -0.022321066, -0.027544279, -0.006137953, 0.0024092742, -0.0003515296,  
    -0.007889225, -0.01853968, -0.037405808, -0.004294867, -0.012602357,  
    0.009385457, 0.02421176, -0.004383281, -0.011901848, -0.020702416,  
    0.00431187, 0.055986296, 0.0040738326, 0.021300908, -0.03713377,  
    -0.004944368, 0.012269106, 0.02614326, -0.019777471, 0.0017206672,  
    0.014867109, 0.004597514, -0.021913003, -0.0042336574, 0.013180447,  
    0.032699477, -0.027245032, -0.0112965545, -0.009358253, -0.004951169,  
    -0.015030335, -0.007501565, -0.018906936, 0.0074675595, -0.0067568496,  
    0.0011952856, 0.0035501514, 0.012316713, 0.016132105, -0.014839904,  
    -0.0044002836, 0.010405616, -0.0066276295, -0.015751246, -0.007141109,  
    0.02729944, -0.00031348618, -0.015669633, -0.015656032, 0.015696838,  
    0.013554505, -0.001635654, 0.006069943, 0.009167824, -0.01917898,  
    -0.006059741, 0.013969369, -0.03006067, 0.007691995, 0.0005028531,  
    -0.005468049, -0.007263528, 0.009970348, -0.016254524, 0.0041248407,  
    0.01964145, 0.037324198, -0.0011340762, -0.0013296065, 0.010392014,  
    -0.01268397, -0.013037625, -0.006811258, -0.0045601083, 0.0028955496,  
    0.027435461, -0.024320576, -0.006161757, 0.014350229, 0.009392259,  
    -0.006328383, -0.025490358, 0.023273215, 0.0083925035, -0.003934411,  
    0.023722084, -0.0072499258, -0.0053626327, 0.017818768, -0.03327077,  
    0.019097365, 0.0022086431, 0.046709653, -0.016825814, -0.032182597,  
    ... 1436 more items  
]
```

- Word embeddings represent words as dense vectors in a continuous vector space. Each word is mapped to a vector of fixed size, where similar words are closer together in the vector space. Popular word embedding techniques include Word2Vec, GloVe (Global Vectors for Word Representation), and FastText.
- Sentence or document embeddings represent entire sentences or documents as vectors. These embeddings capture the semantic meaning and contextual information of the entire text. Techniques such as Doc2Vec (for documents) and Universal Sentence Encoder (for sentences) are commonly used for generating sentence/document embeddings.
- Contextualized embeddings capture the context in which words or phrases appear in a given sentence. Unlike static word embeddings, which assign the same vector representation to a word regardless of context, contextualized embeddings generate different representations for the same word based on its surrounding context. Examples include ELMo (Embeddings from Language Models) and BERT (Bidirectional Encoder Representations from Transformers).
- Pre-trained embeddings are embeddings that have been learned from large corpora of text data using unsupervised learning techniques. These embeddings capture general semantic relationships and can be fine-tuned or used as features in downstream NLP tasks. Pre-trained word embeddings are often used as initialization for neural network models in NLP tasks. They don't require further training like word embedding.

## Generic language model - A next word predictor...



# Usage of embedding

**Text Classification:** Converting text into numerical vectors for classification tasks (e.g., sentiment analysis).

**Similarity Measurement:** Measuring the similarity between words or phrases.

**Named Entity Recognition:** Identifying and classifying entities in text.

**Machine Translation:** Converting text from one language to another using embeddings to represent words.

# Embedding Lab : Covered with Finetuning lab

<https://github.com/CloudSihmar/embedding.git>

# Instruction Tuned language model

## Prompting

e.g.

*Summarize a text of X*

*Generate a poem in the style of X*

*Give me a list of keywords based  
on semantic similarity for X*

Input

## Large Language Model

Output

Summarization

Writing

Keyword extraction

# Dialog Tuned Language Models

Dialog-tuned models are a special case of instruction tuned where requests are typically **framed as questions** to a chat bot.

Dialog tuning is a further specialization of instruction tuning that is expected to be in the context of a longer back and forth conversation, and typically works better with **natural question-like phrasings**.



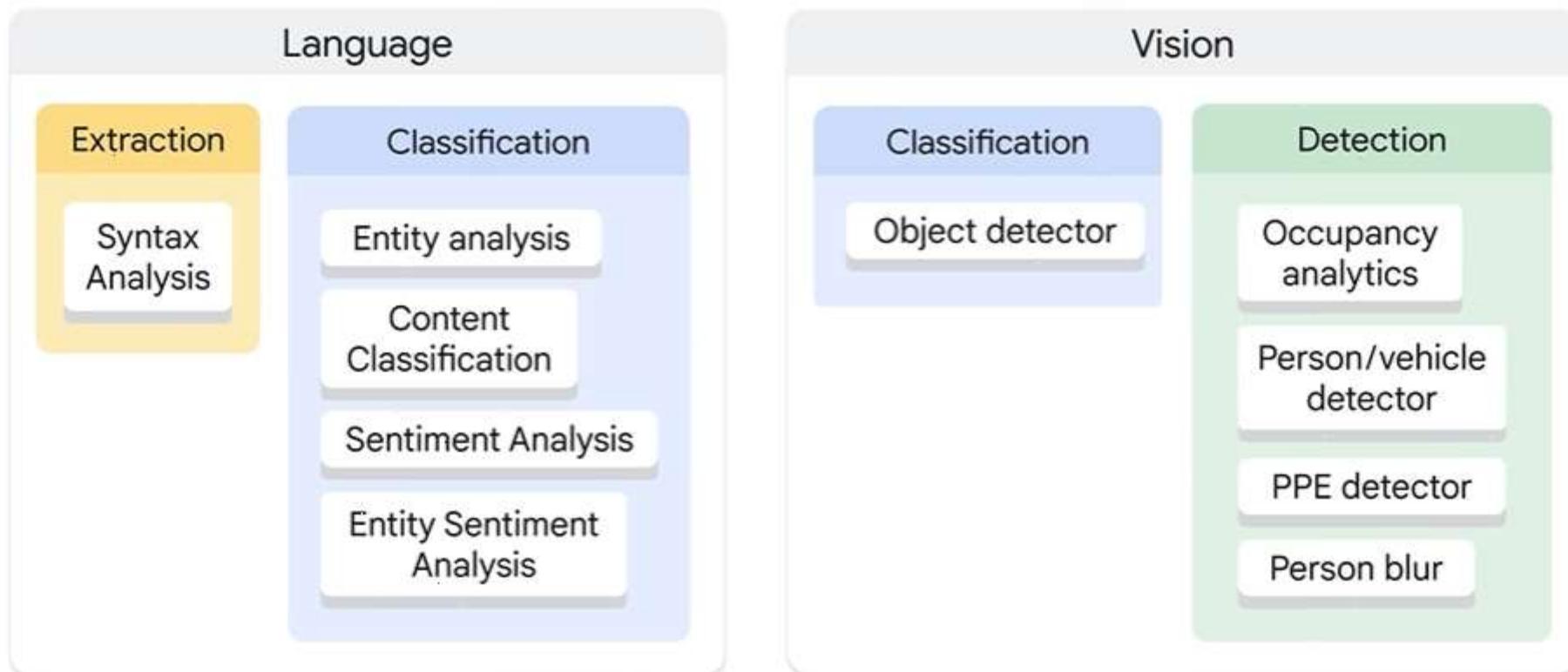
## Prompt examples

[User] Is the comment "do you like the weather?" ok or toxic?  
[Bot] ok.  
[User] can you briefly say why?  
[Bot]

## Model Output

It's just a question about the weather, people are not usually upset by that.

# Model Garden Task Specific Models

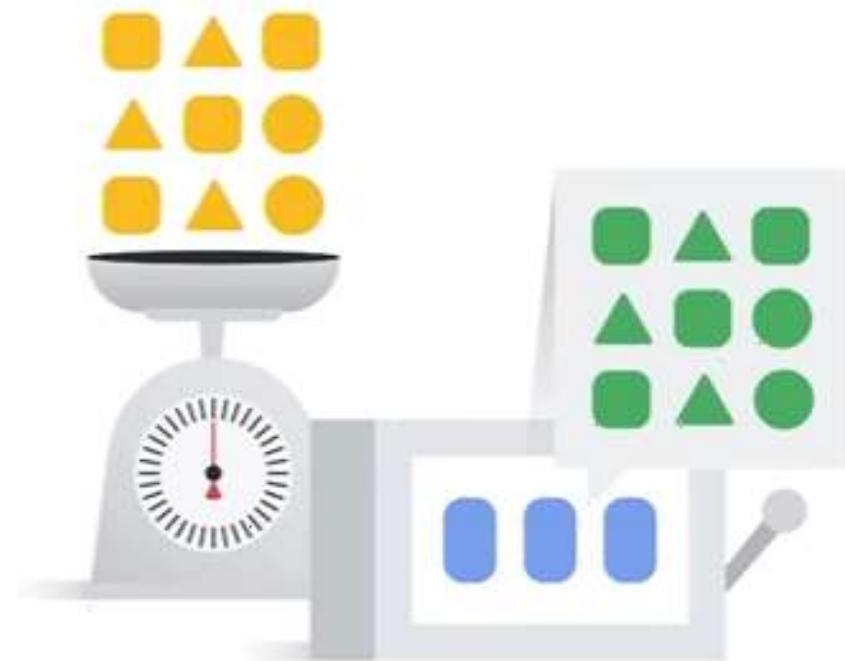


# 4. Fine-tuning

- Fine-tuning refers to the process of taking a pre-trained model and further training it on a new, often smaller dataset or for a specific task. This allows the model to adapt its parameters to better fit the nuances of the new data or task

## Fine tuning

Bring your own dataset of and retrain the model by tuning every weight in the LLM. This requires a big training job (like really big) and hosting your own fine-tuned model.



# How fine-tuning typically works:

- **Pre-trained Model:**

Start with a pre-trained model that has been trained on a large dataset for a general task, such as image classification with a **convolutional neural network (CNN)** or language modeling with a **recurrent neural network (RNN)**.

Pre-trained models are usually trained on large, diverse datasets and have learned useful features or representations that can be transferred to related tasks or datasets.

- **Task-specific Adaptation:**

Fine-tuning involves taking the pre-trained model and adjusting its parameters (weights) by training it on a new dataset or for a specific task.

The new dataset may be smaller and more specific to the target task, such as a dataset of **medical images for diagnosing diseases** or a dataset of **legal texts for contract analysis**.

By training on the new dataset, the model learns to adapt its learned representations to better suit the target task or dataset.

- **Training Procedure:**

During fine-tuning, the parameters of the pre-trained model are typically updated using a smaller learning rate compared to the initial training phase.

This smaller learning rate helps prevent catastrophic forgetting, where the model forgets useful features learned during pre-training.

The fine-tuning process may involve training the entire model (fine-tuning all layers) or only specific layers, depending on the similarity between the pre-trained task and the target task.

# Benefits of fine-tuning

- Fine-tuning leverages the knowledge encoded in pre-trained models and adapts it to new tasks or datasets, often requiring less labeled data and computation compared to training a model from scratch.

It allows practitioners to quickly develop effective models for specific tasks without starting from scratch, saving time and resources.

- Overall, fine-tuning is a valuable technique in machine learning for transferring knowledge from pre-trained models to new tasks or datasets, enabling efficient and effective model development for a wide range of applications.

# Fine Tuning Example with GradientAI

[https://github.com/ramannkhanna2/FineTuning\\_LLmModel.git](https://github.com/ramannkhanna2/FineTuning_LLmModel.git)

# Gen AI Use cases

## Generative AI use cases



### Customer experience

- Chatbots
- Personalization
- Virtual assistants
- Content moderation



### Employee productivity

- Conversational search
- Text summarization
- Code generation



### Creative content production

- Artwork
- Music
- Images
- Animations and video



### Business operations

- Document processing
- Maintenance assistants
- Quality control
- Visual inspection

# Case study : Chatbots for Customer Support

- Bank of America implemented a virtual assistant named Erica, powered by generative AI. Erica assists customers in managing their accounts, providing balance information, helping with transactions, and answering general banking queries. The chatbot uses natural language processing to understand customer inquiries and responds in a conversational manner. This has led to improved customer satisfaction and reduced response times.

# Case study : Code Assistance

- GitHub Copilot, powered by OpenAI's Codex, assists developers by suggesting entire lines or blocks of code as they write. This accelerates the coding process and helps developers with code completion

# Case study : Content Generation

- HubSpot integrated OpenAI's GPT-3 into its content strategy tool. Marketers can input a brief and receive multiple content suggestions, including blog post introductions, email subject lines, and social media captions. This has significantly expedited the content creation process, allowing marketers to generate ideas and drafts quickly.

# Case study : Medical Image Analysis

- **Aidoc employs generative AI** for medical image analysis, particularly in **radiology**. The system can quickly analyze and highlight abnormalities in medical scans, assisting radiologists in identifying potential issues more efficiently. By automating the initial analysis, Aidoc's AI helps reduce the time required for diagnosis and enhances the overall workflow in healthcare settings.

# Case study: Creativity and Art

- **Text-based Art:** OpenAI's **DALL-E** generates images based on textual prompts. Users can describe a scene, and the model creates an image representing that description.
- **Image Generation:** AI art generators like **Runway ML's models** allow artists to experiment with creative visuals by generating images or modifying existing ones using generative algorithms.

# Case study: Educational Personalization

- **DreamBox** Learning is an educational platform that uses generative AI to provide personalized learning experiences for students. The platform adapts to each student's progress, adjusting the difficulty of lessons and suggesting targeted activities to address specific learning needs. This approach has shown improvements in student engagement and learning outcomes.
- These case studies demonstrate how generative AI technologies are being applied in real-world scenarios, leading to tangible benefits such as improved efficiency, enhanced user experiences, and breakthroughs in fields like healthcare and education. The examples provided showcase the versatility and impact of generative AI across different industries.

# Typical SDLC



# Improving Effectiveness in software development

- AI is anticipated to be invaluable to all aspects of modern software companies, and you can improve the software development process by incorporating it into different sections.
- This technology empowers software engineers to work more efficiently, resulting in more successful and impactful software development projects.

# Use cases in Software Development

- GenAI can assist developers by automatically generating code snippets or even entire programs based on high-level specifications or requirements. It can analyze existing codebases, understand patterns and structures, and generate new code that follows best practices and coding conventions.
- With the assistance of AI, tests can be automated, eliminating the need for manual execution by quality assurance analysts. This not only saves significant time but also ensures a more comprehensive coverage of test scenarios. By enabling fast and accurate testing, AI greatly minimises the failure rate and shortens the overall development cycle.
- GenAI systems can help in code review processes by analyzing code changes, identifying potential bugs, and suggesting improvements. By learning from past code reviews and bug fixes, GenAI can provide valuable insights to developers, improving code quality and reducing the likelihood of introducing errors.
- GenAI-powered NLP models can assist in writing documentation, generating API references, and providing answers to developer queries. They can understand natural language inputs, extract relevant information from documentation and code comments, and provide accurate responses to developer inquiries.

- GenAI can assist in project management by analyzing project data, predicting project timelines, and recommending task prioritization strategies. It can identify dependencies, estimate task durations, and optimize resource allocation to ensure projects are delivered on time and within budget.
- GenAI can automate deployment processes by orchestrating infrastructure provisioning, configuring environments and managing deployment pipelines. It can streamline CI/CD workflows, detect deployment errors, and automatically roll back changes when issues arise.
- GenAI can help in identifying security vulnerabilities, analyzing code for potential security risks, and recommending remediation strategies. It can leverage machine learning algorithms to detect patterns indicative of security threats, such as code injection, cross-site scripting (XSS), and SQL injection.
- GenAI can analyze code performance, identify bottlenecks, and suggest optimizations to improve efficiency and resource utilization. It can learn from profiling data, runtime metrics, and historical performance improvements to provide tailored recommendations for optimizing code.

# Use cases in Retail/Marketing

- GenAI algorithms can analyze customer behavior, purchase history, and preferences to provide personalized product recommendations. By understanding individual preferences and trends, retailers can increase customer engagement, boost sales, and enhance customer satisfaction.
- GenAI models can analyze historical sales data, market trends, weather patterns, and other factors to predict future demand for products. Retailers can optimize inventory levels, reduce stockouts and overstock situations, and minimize holding costs by leveraging accurate demand forecasts generated by GenAI.
- GenAI-powered pricing algorithms can dynamically adjust prices based on real-time market conditions, competitor pricing, demand elasticity, and other factors. Retailers can maximize revenue, improve price competitiveness, and optimize profit margins by implementing dynamic pricing strategies.
- GenAI-powered chatbots and virtual assistants can handle customer inquiries, provide product information, process orders, and assist with customer service inquiries. By automating routine tasks and inquiries, retailers can improve operational efficiency, reduce customer service costs, and deliver faster response times to customers.

- GenAI algorithms can enable **visual search capabilities**, allowing customers to search for products using images rather than text. Retailers can enhance the shopping experience, increase conversion rates, and drive sales by enabling customers to easily find products that match their preferences through visual search.
- GenAI models can analyze transaction data, user behavior, and patterns indicative of fraudulent activity to detect and prevent fraud in real time. Retailers can minimize financial losses, protect customer data, and maintain trust by deploying robust fraud detection systems powered by GenAI.
- GenAI algorithms can optimize various aspects of the supply chain, including logistics, transportation, warehousing, and distribution. By optimizing supply chain operations, retailers can reduce costs, improve delivery times, minimize stockouts, and enhance overall efficiency.
- GenAI-powered **sentiment analysis** tools can analyze customer feedback, social media mentions, and reviews to gauge customer sentiment and identify emerging trends. Retailers can gain valuable insights into customer preferences, opinions, and sentiments, enabling them to tailor marketing strategies, product offerings, and customer experiences accordingly.

# Case Study: Vodafone/VIAVI 5G Testing

- **Problem Statement:** 5G and ORAN testing by VIAVI/Vodafone requires 6 weeks to run on a test environment. Part of this is the large test suite and post collating test data requires human review of **60000+ LTE network parameters**
- **Solution:** Testing cycles reduced to 10 days. ML engine reviews code diff for a release cycle and identifies test cases required for coverage. Anomaly detection model runs on top of the generated data providing realtime insights on running tests and allowing early termination for critical failures.

# Strategic Decision Making

- AI can significantly impact strategic decision-making because it bases on historical data and analytics to automate processes and reduces the reliance on human involvement.
- AI-driven decision-making also offers the advantage of eliminating human biases and errors.
- By leveraging data and applying machine learning algorithms, computers can gather, analyse, and utilise data to make intelligent and informed decisions.

# Case Study: Dropbox Incident Management

- **Problem Statement:** Dropbox did a major migration of FS to Rust. Rework was done around realtime file update and sharing, locks etc. Since this could not be a partial rollout or canary monitoring needed to be robust during and post rollout.
- **Solution:** Dropbox ingested all logs and metrics into an in-house anomaly detection engine. The engine was able to isolate hardware, RAID and business logic failures allowing the team to focus on the right signals for application stability

# Transforming the Developer Role

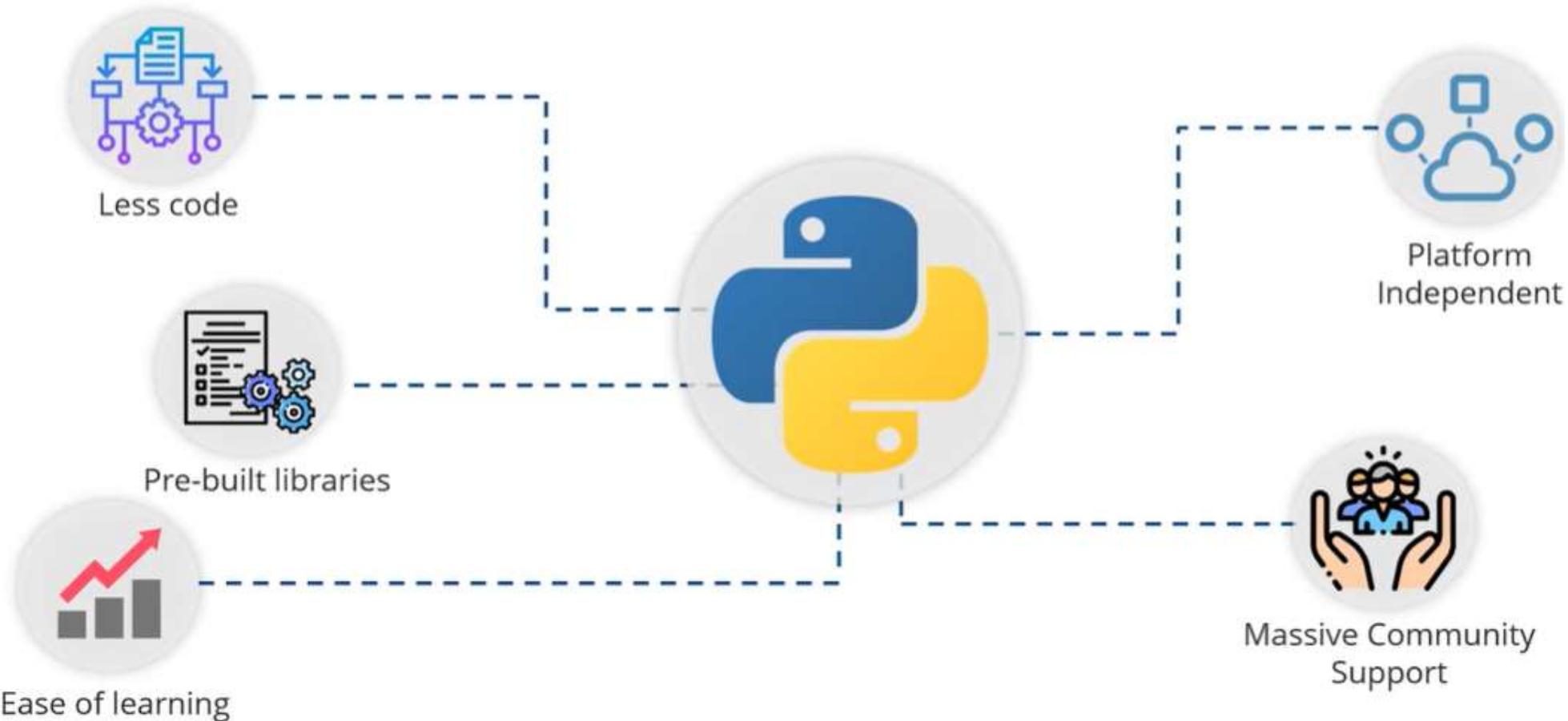
- The role of software developers has evolved because of the presence of AI. By automating tasks and assigning them to AI-powered systems, developers can concentrate their abilities on a diverse set of intricate tasks and develop collaborative abilities that complement AI. This evolution will improve the software development process rather than render it obsolete.
- The presence of AI in the development process opens up the need for a new breed of software developers - those who can effectively work alongside AI and even code it. Although there is speculation that AI may eventually write code, it will not replace programmers entirely. The complexity of software development demands human expertise to provide direction and context.

# More Accurate Estimation

- By utilising AI trained on data from previous projects, you can provide precise estimates, enabling predictions regarding the time, effort, and budget needed for a project.
- While a knowledgeable prediction requires expertise and contextual understanding, AI can be trained to incorporate these aspects as well.
- Atlassian has developed a new model and done a beta rollout that can break an epic to tasks/stories and give development timelines for a team roster.

# Python for AI

# Python For AI



# Python Packages For AI: TensorFlow

Tensorflow library was developed by Google in collaboration with Brain team. It is popularly used in Machine Learning algorithms.

## Features of Tensorflow:

- **Responsive construct:** TensorFlow includes TensorBoard, a visualization tool that allows you to visualize and debug your models, track training metrics, and more.
- **Flexible:** It allows developers to build and train a wide variety of machine learning models, including deep learning models, neural networks, and more.
- **Easily trainable:** It provides high-level APIs, such as Keras, that make it easy to build and train models with just a few lines of code.
- **Parallel neural network training:** It is allowing you to train and deploy models on different platforms, from mobile devices to large-scale distributed systems. You can train multiple neural networks

# Python Packages For AI: Scikit-learn

It is a Python library associated with **NumPy** and **SciPy**. It is considered as one of the best libraries for working with complex data.

## **Features of Scikit Learn:**

- **Feature Extraction and Transformation:** Scikit-learn includes tools for feature extraction and transformation, such as PCA (Principal Component Analysis), feature selection, and feature scaling.
- **Cross Validation:** It is a technique used to assess the performance and accuracy of a machine learning model.
- **Unsupervised Learning Algorithms:** clustering, factor analysis, all unsupervised neural network will be implemented

# Python Packages For AI: NumPy

It is python library mainly used for computing **scientific/mathematical** data. Even Tensorflow is using Numpy internally for multiple operations.

## Features:

- **Supports multi-dimensional arrays:** you can apply operations to entire arrays without using explicit loops.
- **Numerical analysis:** Simplify the complex computation.
- **Memory Efficiency:** NumPy arrays are memory efficient compared to Python lists, especially for large arrays, as they store data in contiguous blocks of memory.

# Python Packages For AI :Theano

It is a Python library that allows you to define, optimize and evaluate mathematical expressions involving **multi-dimentional** arrays efficiently.

It is very similar to Tensorflow but you can not use it in a production environment.

## Features:

- Tight integration with NumPy
- Transparent use of a GPU
- Extensive unit-testing and self verification

# Python Packages For AI: Keras

It simplifies the implementation of neural networks. It also provides some of the best utilities for compiling models, processing data-sets, visualization of graphs and much more.

## **Features:**

- Runs smoothly on both CPU and GPU
- Supports all types of Neural Networks
- Completely Python based

# Natural Language Analyses with NLTK

It is an open source Python library for Natural Language Processing, text analysis and text mining.

## **Features:**

- Study natural languages text
- Text analysis
- Sentimental analysis

# Deep Learning

# Deep Learning

- We have the applications which are already integrated with AI
- Netflix application
- Amazon Application
- Self driving cars
- Chatbots

These applications or tools are integrated with AI to provide the recommendations as per user's interest.

# Machine Learning

- Machine learning is a subset of AI
- It provides the stats-tool to analyse, visualise the data, predictions, forecasting, clustering etc

# Deep Learning

- It is a subset of Machine Learning
- The main aim of Deep learning is to mimic the human brain
- Deep learning is focussed on perceptron

# Why deep learning is becoming so popular?

- Data is getting generated exponentially due to all social networking sites and the data should be stored efficiently.
- Storing and keeping the data is not an option.
- The data should be utilised to make the product better then AI has been introduced.
- Hardware advancement (GPU)

# Chatbots

# Chatbots

- Many companies are using chatbots for their internal and external use.

Messaging Apps

FB bots of companies,  
Hike – Natasha

Customer Service

Swiggy Bot,  
Zomato Bot,  
Vogo Bot

Company internal use

Internal HR bot,  
Internal Finance tool  
bot

# Types of ChatBots

- **Simple chatbots:** They have limited capabilities which are task specific and called rule based bots.
- **Smart Chatbots:** AI-enabled smart chatbots are designed to simulate near-human interaction with customers.
- **Hybrid Chatbots:** They are combination of simple and smart chatbots
- **Social Messaging chatbots**
- **Menu based chatbots**

# AI Chatbots vs Rule-based Chatbot

## AI chatbot

### Pros

- + Uses NLP – Natural Language Processing
- + The user can write questions freely
- + Develops itself over time

### Cons

- Heavy investment at first, profitable in the long run
- Need chat history or input of sample questions to work
- Complex and somewhat demanding setup

## Rule-based chatbot

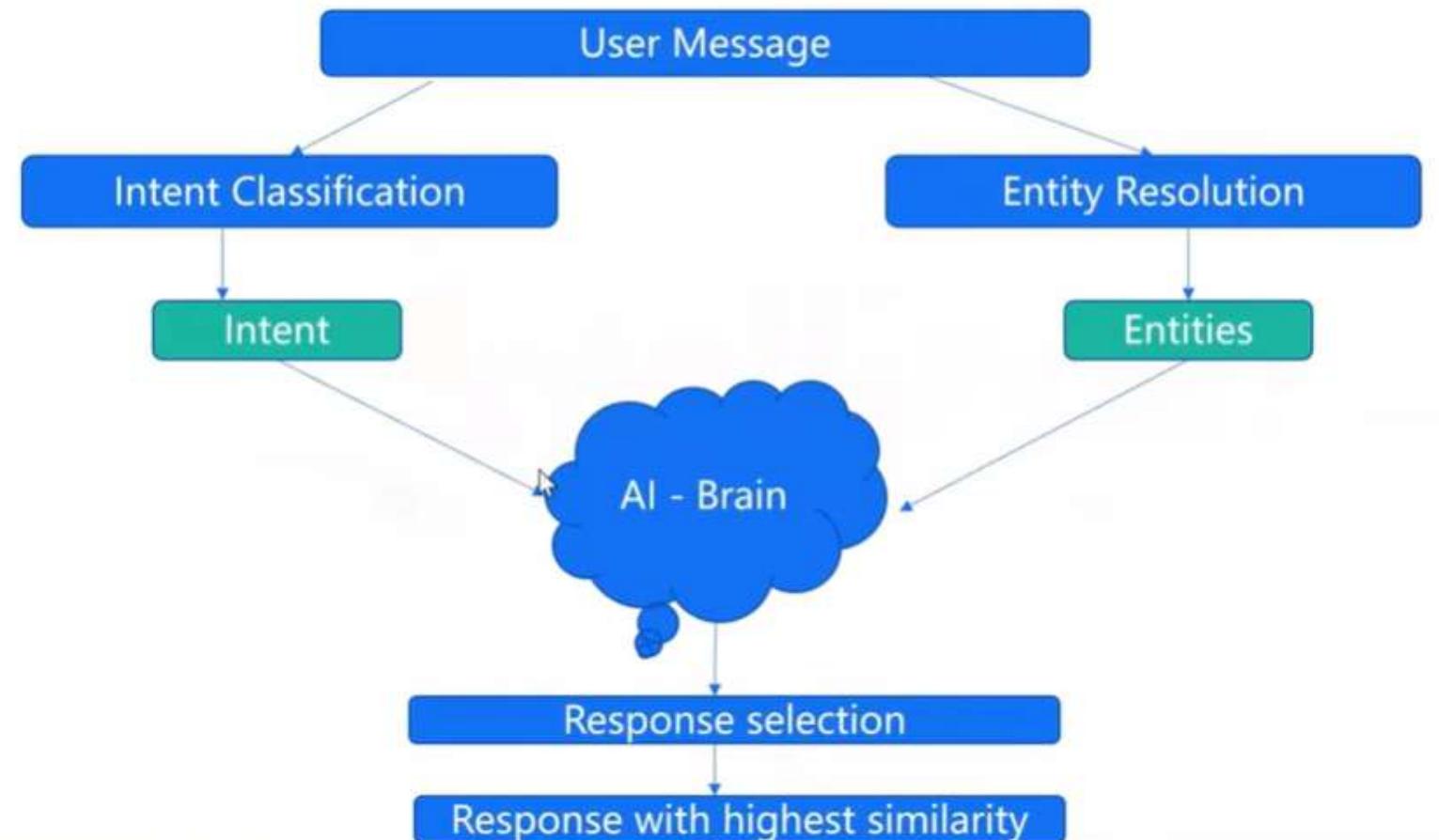
### Pros

- + Simple implementation as it is based on pre-programmed rules
- + Can answer the most common questions, is a professional FAQ
- + Can reach big automation despite its simplicity

### Cons

- Can't solve complex questions
- Does not understand written text or synonyms
- Needs solid preliminary work and thought to work well

# How a Chatbot works?



# Lab: Rule-Based ChatBot

- <https://github.com/CloudSihmar/chatbot-rulebased.git>

# Chatbot with Python and NLTK: Steps

## **1. Reading Text Corpus**

- The first step is to read the text corpus, which is a collection of text documents. This can be done using various methods depending on the format of the corpus (e.g., reading from a file, scraping from the web)

# Chatbot with Python and NLTK: Steps

## **2. PreProcessing (Stop words removal, lower case conversion)**

**Stop Words Removal:** Stop words are common words that do not contribute much to the meaning of a sentence. Removing these words can help reduce the dimensionality of the data and improve the performance of models.

**Lower Case Conversion:** Convert all text to lowercase to ensure uniformity and to prevent duplicate words due to case differences.

# Chatbot with Python and NLTK: Steps

## **3. Tokenization, Stemming and Lemmatization**

**Tokenization** is the process of breaking down text into individual words or tokens. This step is essential for further analysis, as it allows the text to be represented as a sequence of tokens.

**Stemming:** Reducing words to their root or base form (e.g., "running" to "run") using an algorithmic approach. This helps in reducing the number of unique words in the corpus.

**Lemmatization:** Similar to stemming but more accurate, lemmatization reduces words to their base or dictionary form (e.g., "better" to "good"). Lemmatization considers the context and part of speech of the word.

# Chatbot with Python and NLTK: Steps

## 4. Bag of Words

- **BoW** is a representation of text that describes the occurrence of words within a document. It involves creating a vocabulary of all unique words in the corpus and then counting the frequency of each word in each document. The result is a matrix where each row represents a document, each column represents a word, and each cell represents the frequency of that word in the document.

# Chatbot with Python and NLTK: Steps

## **5. One hot encoding**

- One-hot encoding is a method used to represent categorical data, such as words in a text corpus, as binary vectors. Each word in the vocabulary is assigned a unique index, and then each word in a document is represented by a vector where all elements are zero except for the element at the index corresponding to that word, which is set to one.

# Lab: AI-based ChatBot

<https://github.com/CloudSihmar/ai-chatbot.git>

# AWS AI Services

# What is AWS?

- AWS is a cloud service offered by Amazon
- It provides service over internet
- AWS services can be used to build, monitor and deploy any type of application in the cloud.
- AWS can also be used to store your data
- AWS uses the pay for what you use model

# AWS Sage Maker

- Amazon SageMaker is a cloud machine learning platform that helps users in building, training, tuning and deploying machine-learning models in a production-ready hosted environment.

# SageMaker Benefits



# Machine Learning with SageMaker

## 1. Build:

- It provides more than 15 widely used ML algorithm for training purpose.
- Now to build a model, collect and prepare training data or choose from Amazon S3 bucket.
- Sagemaker helps developers to customize ML instances with Jupyter notebook interface.

Choose and optimize the required algorithm such as

- K-means
- Linear regression
- Logistic Regression

# Machine Learning with SageMaker

## **2. Test and Tune**

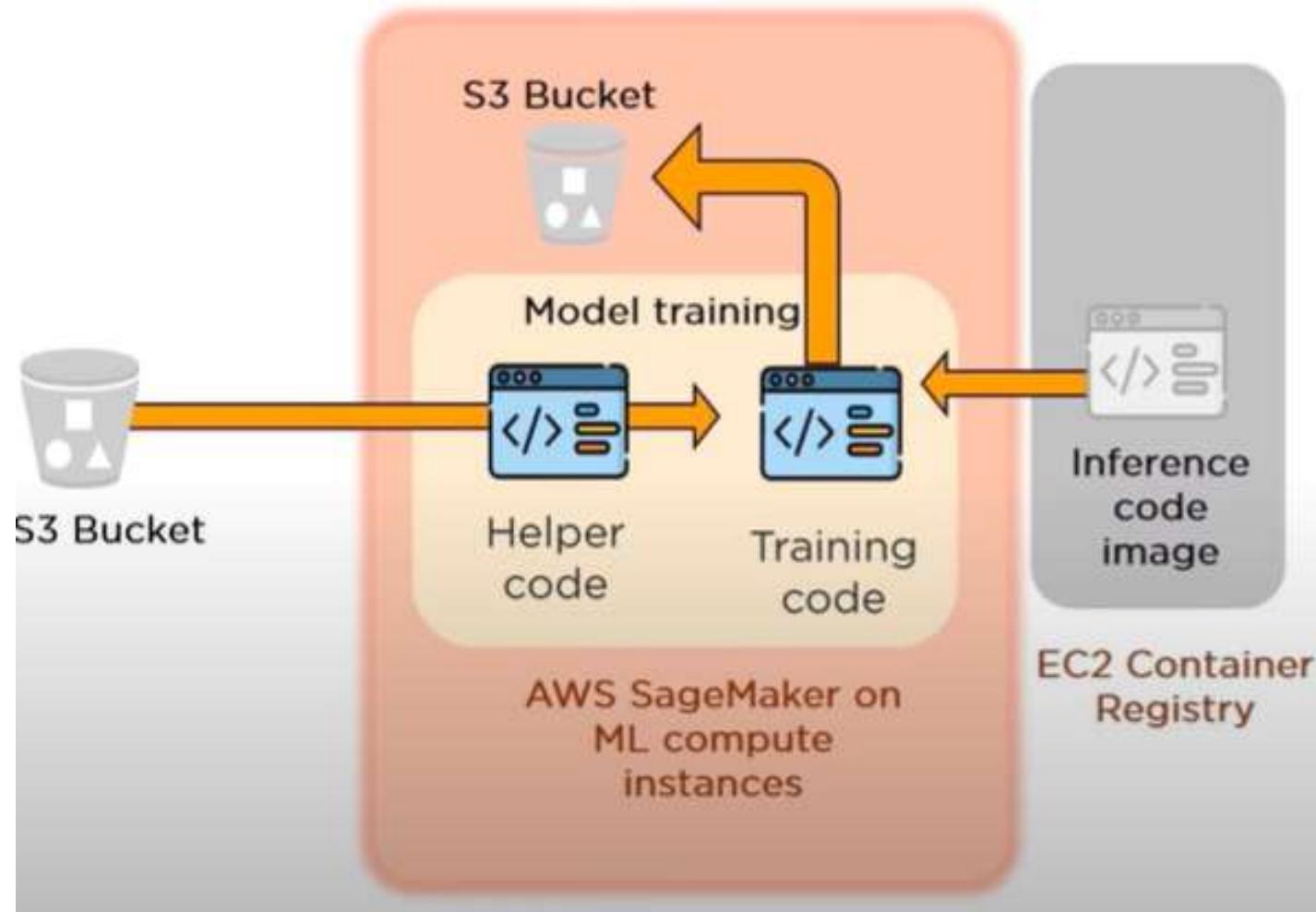
- Setup and manage the environment for training
- Train and tune a model with Amazon SageMaker
- It implements hyperparameter tuning by adding a suitable combination of algorithm parameters
- It divides the training data and stores in Amazon S3
- It divides the training data and stores in Amazon S3 whereas the training algorithm code is stored in ECR.
- Later, SageMaker sets up a cluster for the input data, trains it and stores it in Amazon S3 itself.

# Machine Learning with SageMaker

## 3. Deploy:

- Once tuning is done, models can be deployed to SageMaker endpoints
- In the endpoints, real-time prediction is performed.
- Now, evaluate your model and determine whether you have achieved your business goals

# Train a Model with AWS SageMaker



# Validate a Machine Learning Model

After training a model, evaluate it to determine whether its performance and accuracy enable you to achieve your business goals.

- **Offline testing:** Use historical, not live, data to send requests to the model for inferences.
- **Online testing with live data:** SageMaker supports A/B testing for models in production by using production variants.
- **Validating using a holdout set:** Machine learning practitioners often set aside a part of the data as a "holdout set." They don't use this data for model training.
- **k-fold validation:** In this validation approach, you split the example dataset into  $k$  parts. You treat each of these parts as a holdout set for  $k$  training runs, and use the other  $k-1$  parts as the training set for that run. The value  $k$  is typically in the range of 5-10.

# Companies using SageMaker

- ADP
- Zalando
- Dow Jones
- ProQuest
- Intuit

# Amazon Bedrock

- Amazon Bedrock is a part of AWS that offers developers access to foundational models and the tools to customize them for specific applications.
- Developers do not need to build their own infrastructure to train and host their applications.
- Amazon Bedrock is to make it as easy as possible for developers to build and deploy **Generative AI applications**.

# Amazon Bedrock

- It offers foundation models (LLMs) which are built by other companies-to serve as a **backbone** of a new application.
- It partners with **AI21 labs**, **Anthropic** and **Stability AI** to offer its LLMs for developers to build on.
- Developers can then add their own custom data to further train the model and build out their applications before deploying it with AWS's cloud.
- Amazon is providing access to its own AI model, **Titan**. Developers interested in using Amazon's models will have to use AWS.

# Foundation models



## Jurassic-2

Instruction-following LLMs for any language task including question answering, summarization, text generation, and more



## Command

Transform your products with foundational large language models built for enterprises



## Amazon Titan

Text summarization, generation, classification, open-ended Q&A, information extraction, embeddings and search



## Claude

LLM for conversations, question answering, and workflow automation based on research into training honest and responsible AI systems



## Stable Diffusion

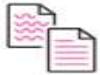
Generation of unique, realistic, high-quality images, art, logos, and designs



## Llama

Meta is looking to unlock the power of large language models. Our latest version of Llama is now accessible to individuals, creators, researchers and businesses of all sizes so that they can experiment, innovate and scale their ideas responsibly

# Applications can be developed using Amazon Bedrock



## Text generation

Create new pieces of original content, such as short stories, essays, social media posts, and web page copy.



## Search

Search, find, and synthesize information to answer questions from a large corpus of data.



## Image generation

Create realistic and artistic images of various subjects, environments, and scenes from language prompts.



## Chatbots

Build conversational interfaces such as chatbots and virtual assistants to enhance the user experience for your customers.



## Text summarization

Get a summary of textual content such as articles, blog posts, books, and documents, to get the gist without having to read the full content.



## Personalization

Help customers find what they're looking for with more relevant and contextual product recommendations than word matching.

# Labs

<https://github.com/CloudSihmar/ai-bedrock/blob/main/cloude.py>

<https://github.com/CloudSihmar/ai-bedrock/blob/main/lima.py>

<https://github.com/CloudSihmar/ai-bedrock/blob/main/stable-diffusion.py>

# AWS AI Services

**Computer  
Vision**

Automated Data  
Extraction &  
Analysis

**Language  
AI**

Improve  
Customer  
Experience

**Business  
Metrics**

**Code  
& DevOps**

**Industrial  
AI**

**Healthcare**

# Computer Vision



Amazon Rekognition



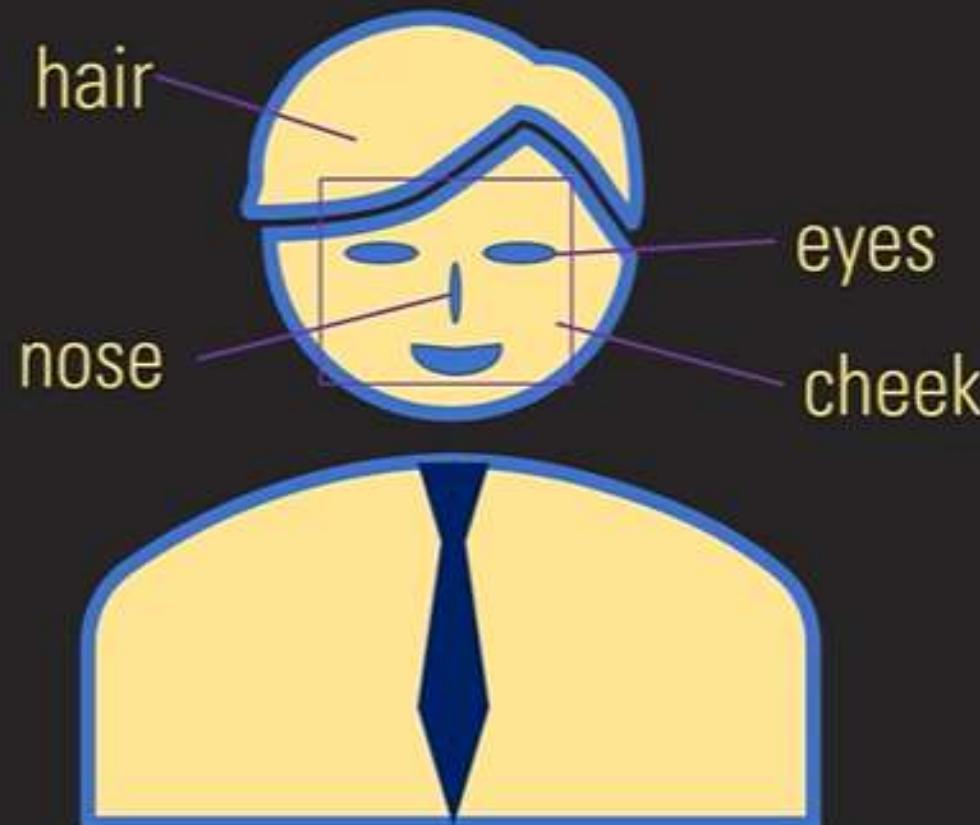
Amazon Lookout for Vision



AWS Panorama



## Face detection & analysis



Identify facial features



## Text detection

Identify Text  
in a image



# Amazon Rekognition: Use cases

- Automating content moderation for user-generated content.
- Enhancing security with facial recognition for identity verification.
- Analyzing customer engagement and behavior in retail environments.

# Amazon Rekognition: Case Study



Aella Credit provides instant loans to individuals with a verifiable source of income in emerging markets using biometric, employer, and mobile phone data.

"Identity verification and validation have been a major challenge in emerging markets. The ability to properly identify users is a key hindrance in building credit for billions of people in emerging markets. Using Amazon Rekognition for identity verification on our mobile application has reduced verification errors significantly and given us the ability to scale. We can now detect and verify an individual's identity in real time without any human intervention, thereby allowing faster access to our products. We tried various well-advertised solutions, but none of the popular alternatives could accurately map out various skin tones. Amazon Rekognition helped us effectively recognize faces of our customers in our markets. It also helped us with KYC in discovering overlapping profiles and duplicate datasets."

**Wale Akanbi, CTO & Co-Founder - Aella Credit**

# Amazon Rekognition: Case Study



C-SPAN is a public service that provides gavel-to-gavel proceedings of the U.S. House of Representatives and the U.S. Senate, and to other forums where public policy is discussed, debated and decided—all without editing. With 3 network stations and 5 other video feeds, there is a lot of content that must be indexed and made searchable.

"By using Amazon Rekognition, we are able to tag who is speaking / on camera at what time down to the second, if needed. Rekognition allows us to index twice as much content as we do currently - from 3500 hours a year to 7500 hours a year which would allow us to index 100% of our first run content and it was shockingly easy to set up, even with 97,000 entities from our database."

**Alan Cloutier, Technical Manager - C-SPAN Archives**

# Amazon Rekognition: Case Study



ARMED™ is dedicated to the development and integration of cutting-edge technology to combat acts of political violence, terrorism, organized criminal activities, and insider threats.

"Our specialty is on safeguarding major events in the face of increasing complex and malevolent environments. Amazon Rekognition powers ARMED™'s Data Fusion System, providing a real-time ability to track individuals in video streams and recognize persons of interest. These capabilities enable predictive analytics that supply security specialists with immediate and potentially life-saving information. We're excited to be working with the Rekognition team in developing a world-class platform that provides security organizations and clients with enhanced situational awareness."

**Shaun McCarthy, CEO - ARMED Inc.**

# Labs

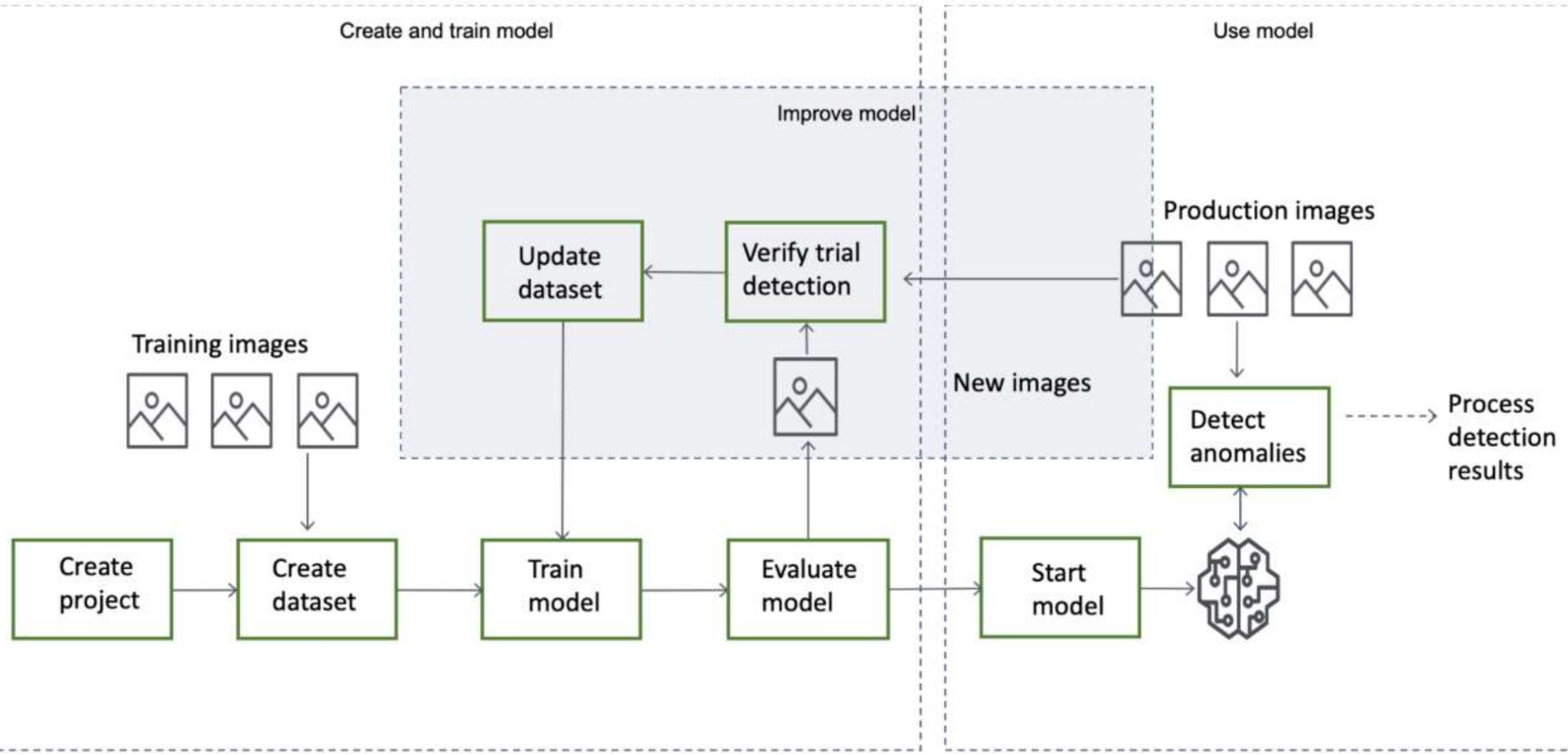
<https://github.com/CloudSihmar/ai-bedrock/blob/main/face.py>



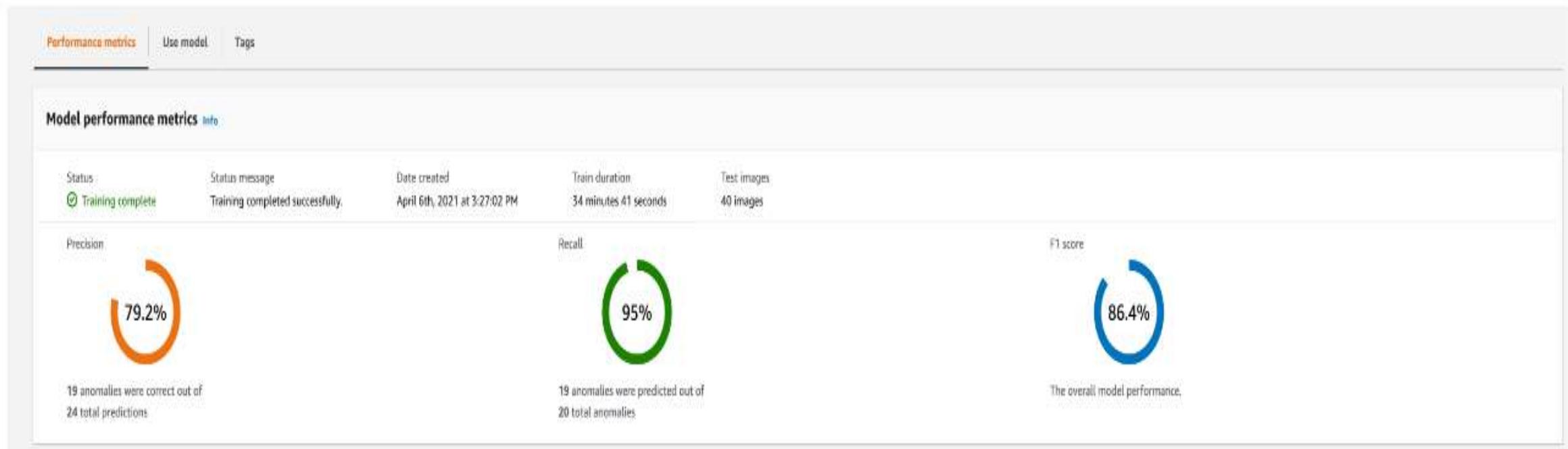
## Spot product defects

Analyze new product images





# Amazon Lookout Result



# Amazon Lookout Result

- **Precision:** Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. In other words, it is the measure of how many of the correctly predicted positive cases were actually positive. A high precision indicates that the model is good at not misclassifying negative cases as positive.

$\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives})$

- **Recall:** Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all actual positive observations. It measures how well the model can find all the relevant cases within the dataset.

$\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives})$

- **F1-score:** The F1-score is the harmonic mean of precision and recall. It provides a single score that balances both precision and recall. It's useful when you want to compare two models and need a single metric to base your decision on.

$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

# Amazon Lookout: Case Study



Dafgards is a household name in Sweden, manufacturing 15 different types of frozen pizzas at over 100 pizzas per minute.

"To uphold our brand and deliver the freshest and tastiest customer experience, we want to ensure that all our pizzas are adequately covered in cheese and with the correct toppings. Early in our production process we installed a machine vision system to detect proper coverage of cheese across a pizza's surface. While this system served well for our original inspection requirement, the system is unable to detect defects on new product types that include multiple toppings. Amazon Lookout for Vision automates and scales inspection of diverse product types such as a cheese pizza with vegetables using the same machine vision system. We successfully expanded our quality assurance for new product types with minimal impact to operations."

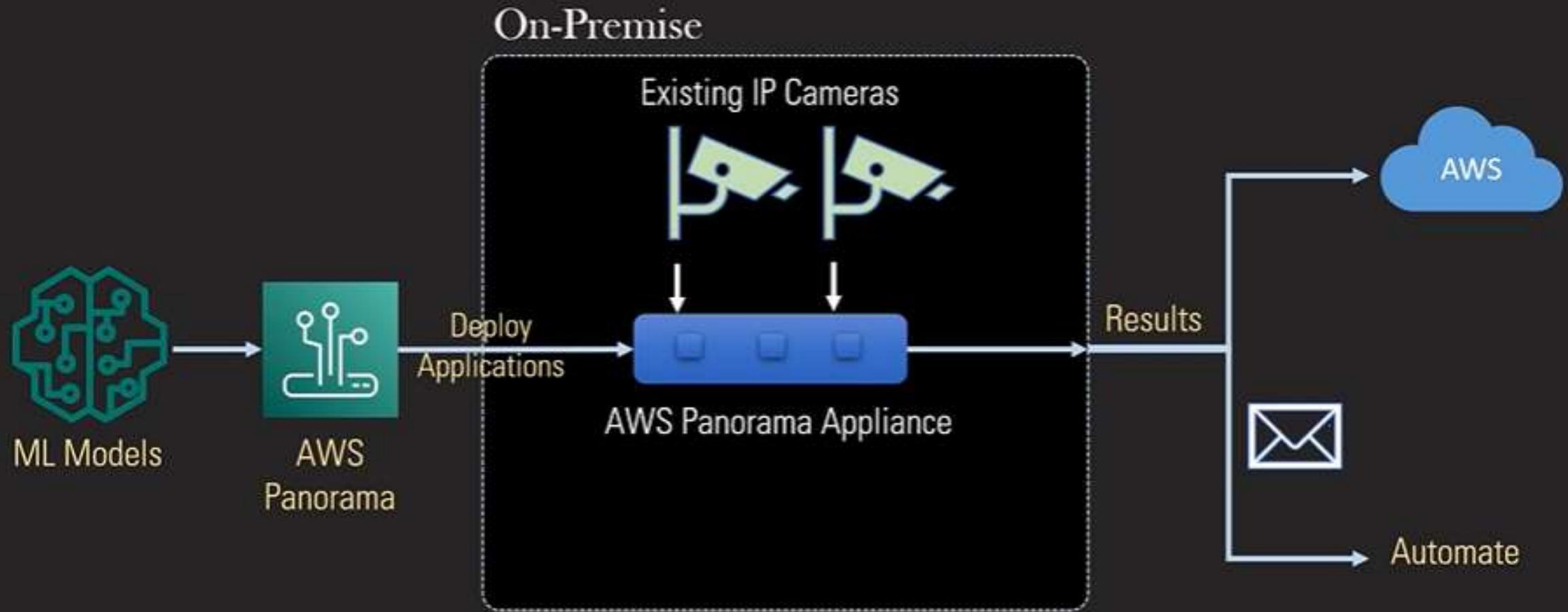
Fredrik Dafgård, Head of Operational Excellence & Industrial IoT

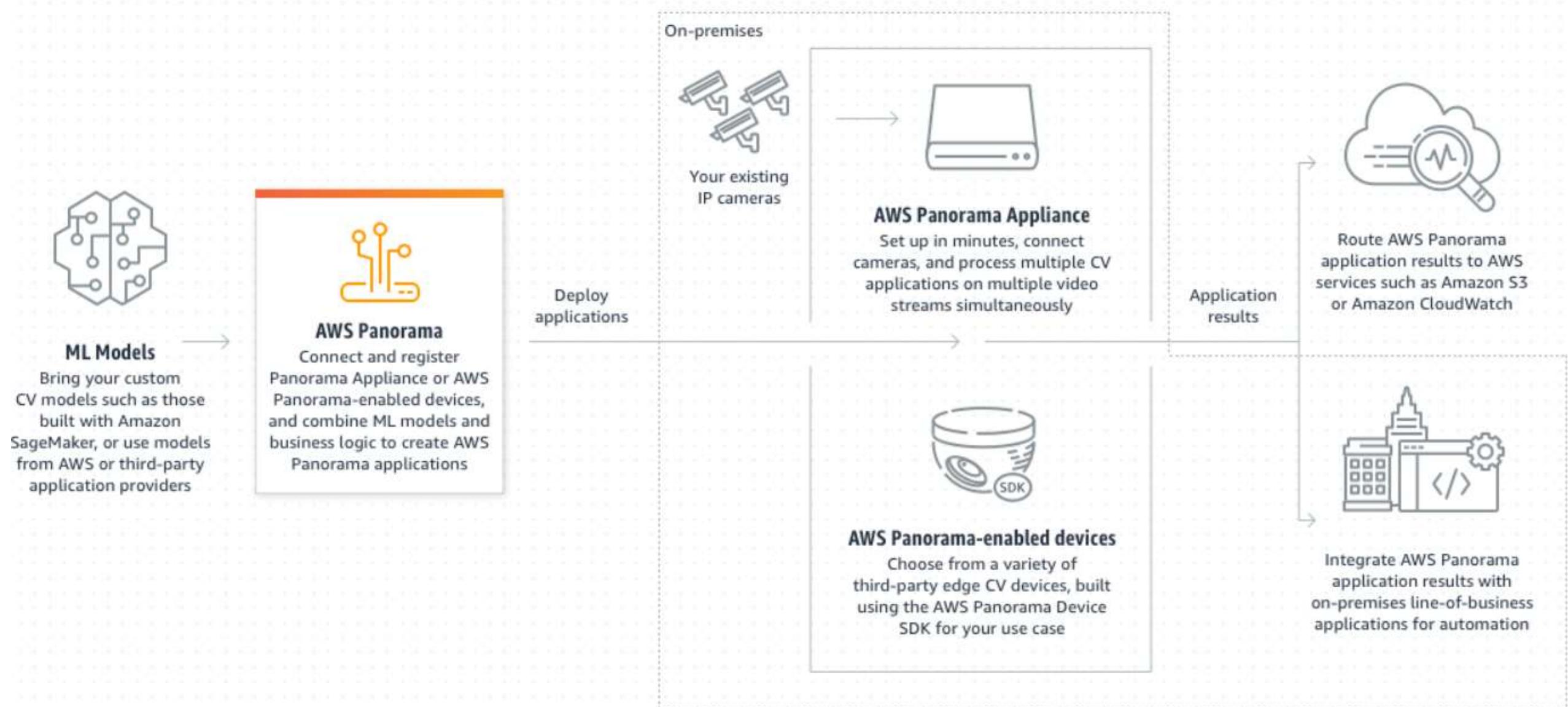
# Amazon-lookout: Demo Lab

- <https://github.com/CloudSihmar/amazon-lookout.git>



# Computer Vision On-Premise





# Use Cases

- **Improve supply chain logistics**

Easily track throughput, optimize freight operations, and recognize objects such as parts or products, or text in labels or barcodes.

- **Optimize traffic management**

Monitor traffic lanes for issues such as stopped vehicles, and send real-time alerts to staff to keep traffic flowing.

- **Evaluate manufacturing quality**

Quickly detect manufacturing anomalies so you can take corrective action and decrease costs.

# Use Cases

- **Safety zones**

Set up dynamic safety zones to alert employees when they are entering hazardous environments.

- **Retail visitor analytics**

Improve the customer experience with advanced retail insights into peak travel times and queue monitoring.

- Drive-thru and curbside pickup optimization

Increase customer satisfaction with faster drive-thru operations and easier curbside check-in.



# Extract text from scanned documents



# Amazon Textract

- It is a scalable document analysis service
- Extract **printed text, handwritten text** and **structured data** from documents like **financial reports, medical records, tax form** and so on.
- It is also capable of processing **receipts, invoices** or **expense** related documents at scale
- It used deep learning technology
- It can extract data from document without any templates and configuration.
- You do not need any machine learning expertise to use this.

# What Amazon Textract can extract?

- Printed text and handwritten text
- Printed text in **Spanish, Italian, French, Portuguese and German**
- **ASCII** symbols
- Tables, Forms (key-value), relationship, confidence score
- Labelled data, implied data, expense line items for goods or services receipts.
- Supported documents are **PNG, JPEG and PDF**

# Amazon Textract APIs and Pricing

APIs	Detect Document Text	Analyze Document	Analyze Document	Analyze Expense
<b>Functionality</b>	<ul style="list-style-type: none"><li>• OCR</li></ul>	<ul style="list-style-type: none"><li>• OCR</li><li>• Tables</li></ul>	<ul style="list-style-type: none"><li>• OCR</li><li>• Forms (Key-value pairs)</li></ul>	<ul style="list-style-type: none"><li>• Key-value pairs</li><li>• Expense items</li></ul>
<b>Pricing</b>	<ul style="list-style-type: none"><li>• 1M pages: \$1.50/1000 pages</li><li>• Over 1M pages: \$0.60/1000 pages</li></ul>	<ul style="list-style-type: none"><li>• 1M pages: \$15.00/1000 pages</li><li>• Over 1M pages: \$10.00/1000 pages</li></ul>	<ul style="list-style-type: none"><li>• 1M pages: \$50.00/1000 pages</li><li>• Over 1M pages: \$40.00/1000 pages</li></ul>	<ul style="list-style-type: none"><li>• 1M pages: \$10.00/1000 pages</li><li>• Over 1M pages: \$8.00/1000 pages</li></ul>

# Benefits of Amazon Textract

- Easy to integrate with existing application
- Scalable document analysis service
- No need to implement from scratch
- No machine learning expertise is required
- Ready to use
- Transparent pricing
- OCR, extract structured data like tables, forms and expense related documents.

# Amazon Textract: Use Cases

- Automating Document Processing
- Enhancing Compliance and Auditing
- Improving Customer Onboarding
- Streamlining Claims Processing
- Optimizing Financial Operations
- Enhancing Digital Transformation Initiatives
- Boosting Efficiency in Human Resources
- Supporting Legal and Contract Management
- Improving Healthcare Administration
- Facilitating Research and Development

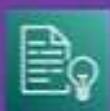
# Amazon Textract: Case Study

## Anthem's Solution

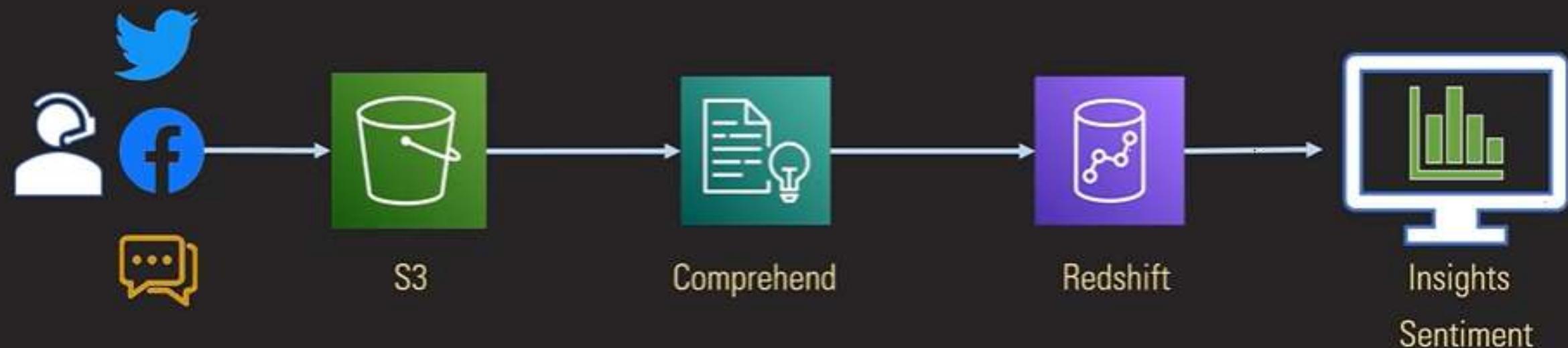
- Turning to Amazon Web Services (AWS), **Anthem** used Amazon Textract—a fully managed machine learning service that automatically **extracts printed text, handwriting**, and other data from scanned documents—to digitize and automate its claims process.
- Anthem chose Amazon Textract for its image-processing capability, ability to detect **tables** and **forms**, and adherence to security and compliance standards.
- After a medical provider submits documents to Anthem's provider portal, the documents enter a data store and are sent to a **digital-processing function powered by Amazon Textract**. Inside this function, Amazon Textract uses optical character recognition to extract data and uses machine learning to automatically index and classify each document. The processed claim is then made available to Anthem stakeholders.

# Lab

<https://github.com/CloudSihmar/ai-bedrock/blob/main/textert.py>



Find valuable insights in documents



# Amazon Comprehend

- A fully managed natural language processing service that leverages machine learning to find insights and relationships in text.
- ML expertise is not required
- Comprehend identifies the **language of the text, extracts key phrases, places, people, brands or events**
- It understands if the text expresses **positive or negative** sentiments and automatically organizes a collection of text files by topic
- You can build a custom set of entities or text classification models that are tailored uniquely to your organization needs.
- Amazon comprehend Medical can be used to identify conditions, medications, dosages, strengths and frequencies from sources like doctor's notes, clinical trial reports, and patient health records.

# Amazon Comprehend: Use Cases

- Sentiment analysis on customer reviews and feedback.
- Topic modeling to understand common themes in large text datasets.
- Entity recognition to extract key information from documents.

# Amazon Comprehend: Case Study

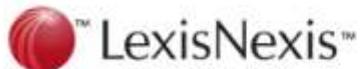


FINRA is a not-for-profit organization dedicated to investor protection and market integrity. It regulates one critical part of the securities industry – brokerage firms doing business with the public in the United States.

"FINRA receives millions of documents with unstructured data to support investigative, examination, and compliance processes. Our investigators and examiners had to manually go through documents page by page or run very targeted searches to find what they needed. With Amazon Comprehend, we can quickly extract individuals and organization, match extracted entities to FINRA records, flag individual of interest, and detect similarities with other documents."

-Dmytro Dolgopolov, Senior Director of Technology of FINRA

# Amazon Comprehend: Case Study



LexisNexis Legal & Professional is a global provider of content and technology solutions for legal and business professionals, serving customers in more than 175 countries, offering over 2 billion searchable archives.

"We provide legal professionals with insightful research and analytics to help them make informed decisions. Therefore, we are always looking for better ways to discover insights from legal documents. Thanks to Amazon Comprehend's automatic machine learning (ML), we can now build accurate custom entity recognition models without getting into the complexities associated with ML. The entities that we care about the most, such as judge and attorney, can be identified quickly from over 200 million documents at above 92% accuracy."

-Rick McFarland, Chief Data Officer of LexisNexis

# Amazon Comprehend: Case Study



Siemens built an AWS survey-response processing solution that sends completed surveys to Amazon Comprehend for language identification then to Amazon Translate to execute translations. After Amazon Comprehend anonymizes any names, Amazon SageMaker detects and organizes responses into categories and topics. In addition to returning analyzed, sorted survey results at least 75 percent faster than before, the AWS solution makes the surveying program much less expensive.

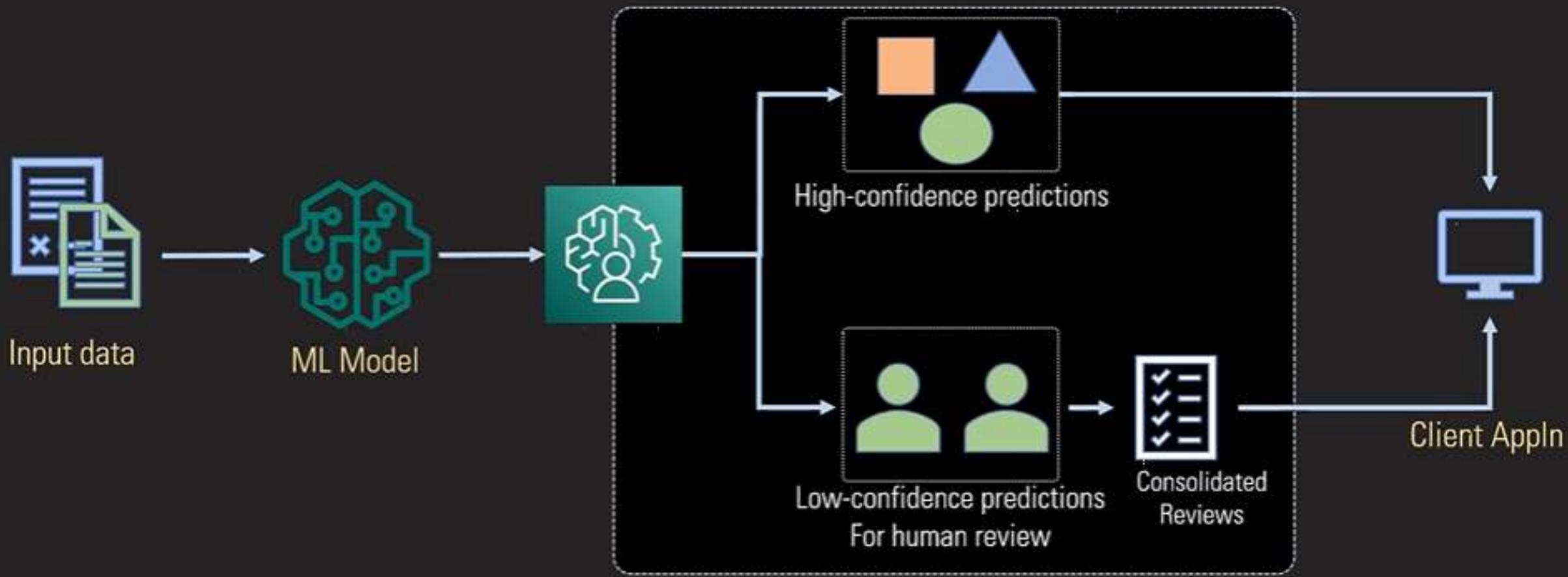
“Procuring human processing and analysis of past employee surveys cost multiple euros per interview. By using Amazon Comprehend and other AWS services, we are getting translation, processing, and analysis for less than one euro per interview.”

# Amazon Comprehend: Lab

<https://github.com/CloudSihmar/ai-bedrock/blob/main/comprehend.py>



# Incorporate human review of ML predictions



# Current State

- Customer infuse ML into a variety of applications
- ML handles major use cases with high speed and low cost
- Machine learning gives probabilistic output
- Low confidence ML results need human review

# ML and Humans work together for multiple use cases

Image Moderation	Form Data Extraction	Customer Support	Media Analysis
Invoice Processing	Loan Approval	Supplier Payments	Prescription Processing
Dating Applications	Model Accuracy Monitoring	Document Categorization	Advertising Analysis
Preventive Maintenance	Bill Processing	Weather Data Processing	Text Classification

# Human review ML predictions

What is the key message of this video ad?

Are the subtitles accurately  
generated by transcription ML models?

# Combining Humans and ML is hard

- Need ML scientists , engineering and operations teams
- Need to manage large number of reviewers
- Need to write custom software to manage review tasks
- Difficult to achieve high review accuracy

# Amazon A2I benefits



# Human review workforce options



## Amazon Mechanical Turk

An on-demand 24/7 workforce of over 500,000 independent contractors worldwide, powered by Amazon Mechanical Turk



## Private

A team of workers that you have sourced yourself (your own employees or contractors)



## Vendors

AWS Marketplace vendors that specialize in human review tasks

# How to use Amazon A2I

## **Step 1: Define human review workflow**

- When to trigger human-review?
- Who should review?
- What should humans see?

# How to use Amazon A2I

## **Step 2: Use Amazon A2I workflow in your API calls**

- Use directly in Amazon Rekognition and Amazon Textract APIs
- Get to know in real time if humans review is needed
- Use with custom ML models

# Amazon A2I: Use Cases

- Quickly extract critical information from healthcare documents
- Process millions of financial service data points
- Integrate human review for any ML workflow

# Amazon A2I: Case Study



Change Healthcare is a leading independent healthcare technology company that provides data and analytics-driven solutions to improve clinical, financial and patient engagement outcomes in the U.S. healthcare system.

"At Change Healthcare, we help accelerate healthcare's transformation by innovating to remove inefficiencies, reduce costs and improve outcomes. We have a robust set of integrated artificial intelligence engines that bring new insights, impact, and innovation to the industry. Critical to our results is enabling human-in-the-loop to understand our data and automate workflows. Amazon Augmented AI (Amazon A2I) makes it easy to build the workflows required for human review of ML predictions. With Amazon A2I becoming HIPAA eligible, we are able to involve the human in the workflow and decision-making process, helping to increase efficiency with the millions of documents we process to create even more value for patients, payers, and providers."

**Luyuan Fang, Chief AI Officer, Change Healthcare**

# Amazon A2I: Case Study



Deloitte is helping transform organizations around the globe. The organization continuously evolves how it works and how it looks at marketplace challenges so it can continue to deliver measurable, sustainable results for its clients and communities.

"Part of setting our clients up for success is helping them leverage the latest technology. Using machine learning enables us to help improve our clients' systems and boost their productivity while reducing time to market for products, services, and applications. As part of providing the latest advancements in ML to our clients, we see the benefits of human-in-the-loop systems adding an extra layer of confidence to ML applications," said Beena Ammanath, Managing Director at Deloitte Consulting LLP. "Our clients in the insurance industry, for example, could use A2I to help verify the accuracy of ML models for automated image-based vehicle damage detection and analysis of text-based insurance claims. We're excited to see the many ways our clients across industries could benefit from incorporating A2I into their ML workflows."

**Beena Ammanath, Managing Director at Deloitte Consulting LLP**

# Demo

<https://github.com/CloudSihmar/amazon-a2i-textract.git>



## Build chatbots with conversational AI



User



Virtual Agent

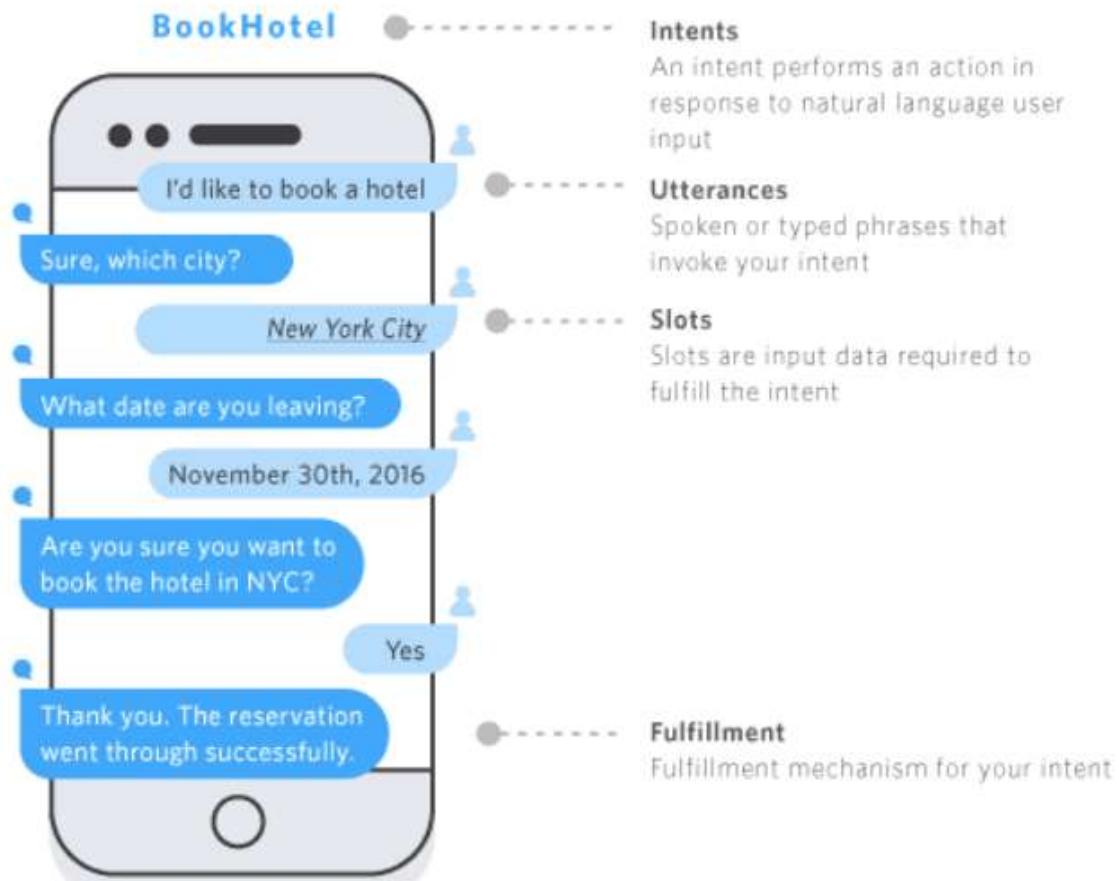
# Amazon Lex

- Amazon Lex is a service for making conversational interfaces into any application using text and voice.
- It provides high-level deep learning capabilities of automatic speech recognition for transform speech to text.
- It also enabled natural language understanding to recollect the intent of the text, to enable us to create applications with highly attractive user experiences and lifelike conversational communication.
- With Amazon Lex, you can rapidly and smoothly build natural language conversational bots (chatbots).

# How Amazon Lex works?

- Amazon Lex works on the same technology as Alexa.
- It combined with AWS Lambda which we are able to use without difficulty trigger features for the execution of our back-end business logic for information retrieval and updates.
- Once built, our bot can be used directly to chat platforms, IoT devices, and mobile clients.
- It provides a secure, scalable, easy-to-use, end-to-end solution to build, monitor, and publish our bots.

# Amazon Lex



# Amazon Lex use cases

- Call Center Voice Assistants and Chatbots
- QnA Bots and Informational Bots
- Application Bots
- Enterprise Productivity chatbots

# Bot Templates

## Airline services

Automate the airline customer service experience for flight reservations and deliver a faster, more connected, omnichannel experience.

[Learn more !\[\]\(9ecae652e5a9fd116487cf1ed2a277a9\_img.jpg\)](#)

[Create](#)

## Auto insurance

Enable rich, customized interactions so insurers can provide faster handling of payments processing, claims reports, policy updates, and policy quotes.

[Learn more !\[\]\(1f26ca522bf7ef21935cda1c862a9a43\_img.jpg\)](#)

[Create](#)

## Financial solutions

Automate the customer service experience for common cardholder activities such as activating the card, making payments, and reporting a lost or stolen card.

[Learn more !\[\]\(542fa88720b66094898d7f1c24e561a4\_img.jpg\)](#)

[Create](#)

## Telecommunications

Automate the customer interaction for activating a SIM, making a payment, and reporting a lost or stolen device, so you can deliver a natural and efficient customer experience.

[Learn more !\[\]\(588cd713e315f62f3cbd4780cbf1a1ef\_img.jpg\)](#)

[Create](#)

## Retail orders

Deliver rich customer service experiences for order management, such as getting order status, tracking a package, and canceling or modifying an order.

[Learn more !\[\]\(81a6a4fcf3421e461c07b4c82f6ffe44\_img.jpg\)](#)

[Create](#)

# Amazon Lex: Intent

- Intent: An intent represents an action that the user wants to perform
- For example, you might create an intent that orders pizzas, books hotels, check balance, apply for a loan, payment issue etc
- Sample utterances: How a user might convey the intent
- For example: a user might say “can I order a Pizza” or “I want to order a Pizza”

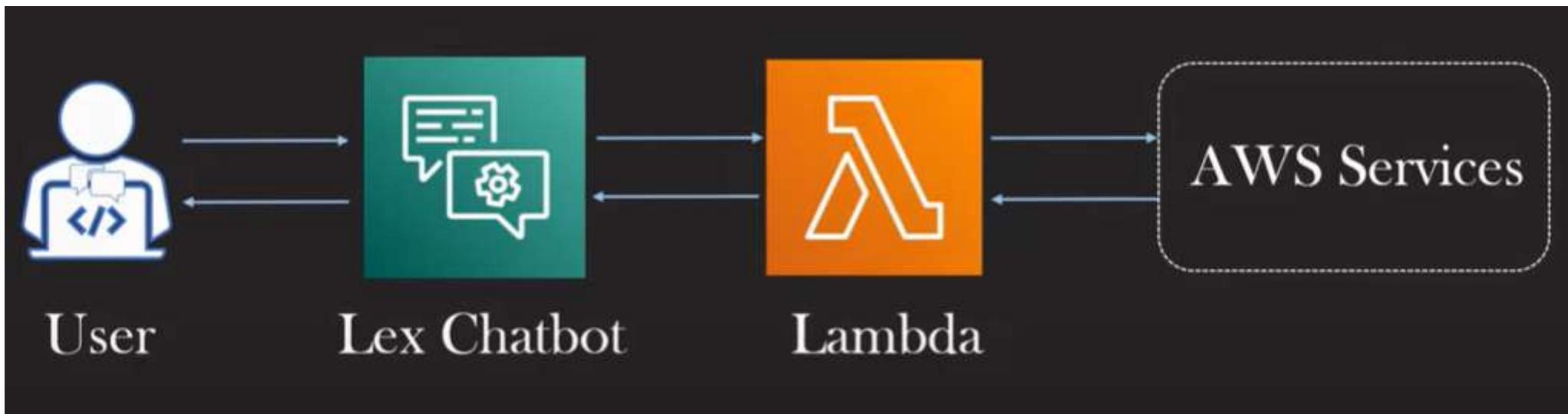
# Amazon Lex: Slot

- Slot: It is an information that Amazon Lex needs to fulfil an intent. Each slot has a slot type. You can create your custom slot types or use built-in slot types.
- For Example: The OrderPizza intent requires slots such as pizza size and pizza type.

# Slot Type

- Each slot has a type, you can create your slot type or use build-in slot types.
- For example, you might create and use the following slot types for the OrderPizza intent:
- Size: With enumeration values Small, Medium and Large
- Crust- With enumeration values Thick and Thin

# Amazon Lex



# Amazon Lex: Case Study



**SAINT LOUIS  
UNIVERSITY**  
— EST. 1818 —

Founded in 1818, SLU is recognized for world-class academics, life-changing research, compassionate health care, and a strong commitment to faith and service.

[Read the blog post »](#)

In order for students and parents to ask questions and get access to their institutions' answers, SLU used the open source QnABot (powered by Amazon Lex) on Amazon Web Services (AWS) to create a chatbot that answered questions on multiple devices.

They went "all in" with Alexa and QnABot because, "every minute we can save our students from having to search for the information they need online is another minute that they can spend gaining the most from their college experience."

**David Hakanson, University Vice President, CIO and Chief Innovation Officer**

# Amazon Lex: Case Study



As Australia's largest business bank, NAB works with small, medium and large businesses to help them start, run and grow, as well as funding some of the most important infrastructure in Australia's communities – including schools, hospitals and roads.

[Watch the video »](#)

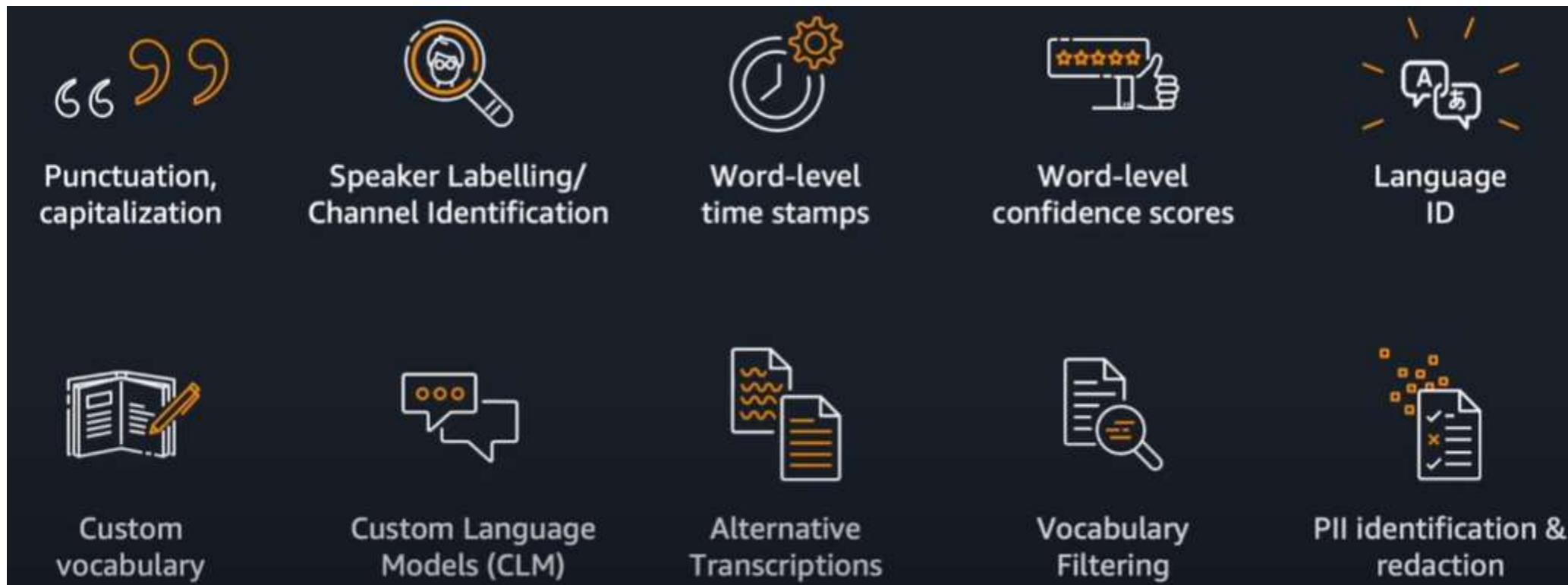
NAB used Amazon Lex to develop automated conversations between their contact center and digital channels. The bank is seeing a high level of 'containment' - people whose calls can be satisfied by an automated or digital channel, in this case 80% and climbing by Amazon Lex. They are also able to add new channels or experience in days versus months.



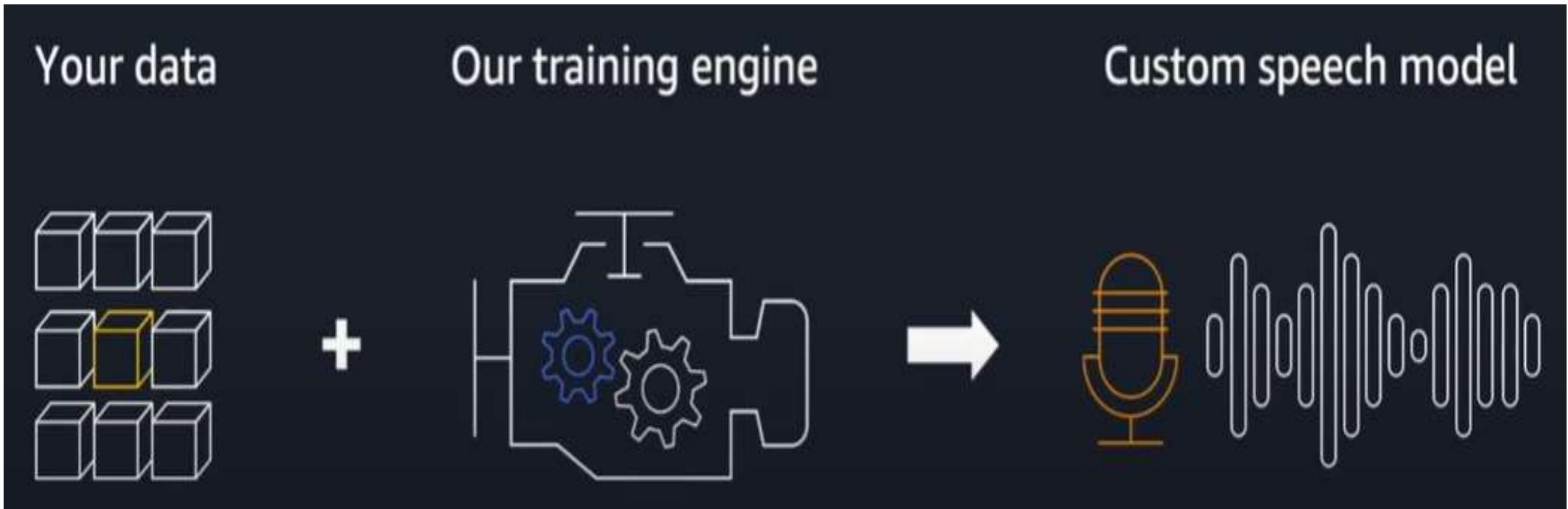
## Convert speech to text



# Amazon Transcribe Features



# Custom Language Model (CLM)



# Custom Language Model (CLM)



## Step 1. Upload data to Amazon S3 [Info](#)

Upload data as plain text files for either model training or tuning. You must provide text that is similar to the audio you want to transcribe. If you provide tuning data, it must have a different S3 prefix from your training data.



## Step 2. Train custom language model [Info](#)

Configure your custom language model for automatic training.



## Step 3. Run transcription jobs [Info](#)

Use your trained custom language model to transcribe audio files.

# Custom Vocabulary

- Less training data required
- No option to provide tuning data
- Focuses on words, independent of context
- Manually input new words
- No additional charge

# Custom Language Model (CLM)

- More training data required
- Option to provide tuning data
- Use each term's context to transcribe audio
- Can automatically add words to the vocabulary
- Tiered pricing which is added to existing transcription charge

# Amazon Transcribe: Case Study



Octopus Energy is a UK-based retail electricity and gas supplier specializing in sustainable energy.

" In our mission to provide customers with fair and transparent pricing along with excellent service, we analyze millions of minutes of contact center calls each month. We've leveraged our own domain-specific data to train custom speech recognition models using Amazon Transcribe. These custom models provide enhanced transcription accuracy and enable us to more efficiently and intelligently identify, understand, and serve our customers' energy needs. "

**Dr. Javier Asensio-Cubero, Data Platform Lead - Octopus Energy**

## Cerner

Cerner Corporation is an American supplier of health information technology services, devices, and hardware.

" Extreme accuracy in clinical documentation is critical to workflows and overall caregiver satisfaction. By leveraging Amazon Transcribe Medical's transcription API, Cerner is in initial development of a digital voice scribe that automatically listens to clinician-patient interactions and unobtrusively captures the dialogue in text form. From there, our solution will intelligently translate the concepts for entry into the codified component in the Cerner EHR system. "

**Jacob Geers, Solutions Strategist - Cerner Corporation**

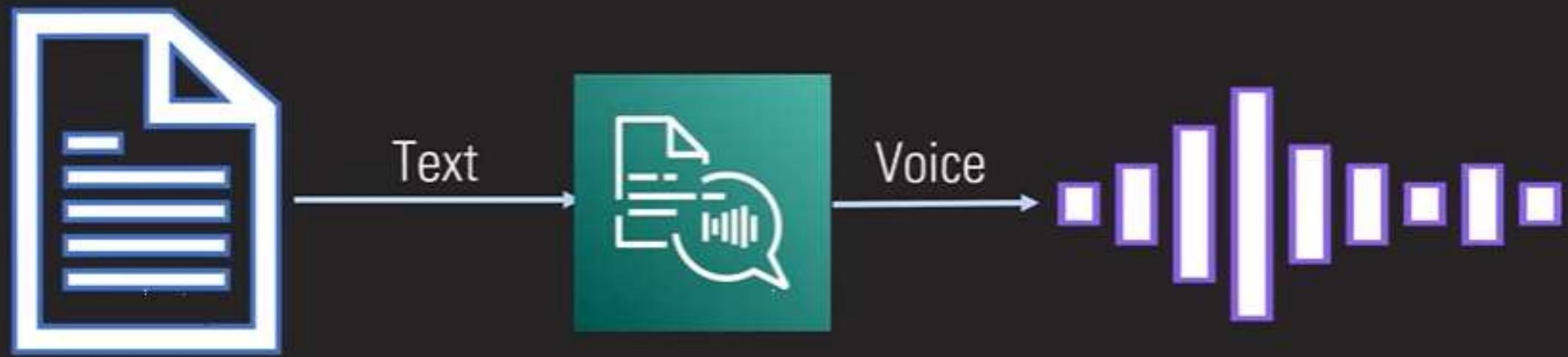


# Amazon Transcribe: Lab

<https://github.com/CloudSihmar/ai-bedrock/blob/main/transcribe-s3.py>



## Convert text to speech



# Amazon Polly

- Amazon Polly is a text-to-speech (TTS) service that leverages advanced deep learning technologies to convert written text into natural-sounding speech.
- Developed by Amazon Web Services (AWS), Polly provides a lifelike voice experience, allowing users to create engaging and dynamic content in multiple languages and voices.

# Amazon Polly: Features and Capabilities

- **Wide Range of Voices:** Amazon Polly boasts a diverse selection of voices, enabling users to choose the tone and style that best suits their content.
- **Customization Options:** Users can fine-tune the speech output by adjusting parameters such as pitch, rate, and volume.
- **Multilingual Support:** Amazon Polly's support for multiple languages facilitates effective communication, breaking down language barriers and reaching a broader audience.
- **Support for a large portfolio of languages and voices** – Amazon Polly supports dozens of voices languages, offering male and female voice options for most languages. This number will continue to increase as we bring more neural voices online.
- **Cost-effective** – Amazon Polly's pay-per-use model means there are no setup costs. You can start small and scale up as your application grows.

# Amazon Polly: Case Study

## FICO



FICO Customer Communication Services (CSS) team focuses on providing automated voice, text and email communications to help organizations with fraud detection and customer service. Learn how they use Amazon Polly Text-to-Speech service to power a range of voice applications to improve the customer experience.

## WaFd Bank



WaFd is a US retail and commercial bank with over 200 branches in eight states.

"When claims first call in, we use Amazon Transcribe, which takes what they are saying and translates it into text. Once that speech has been translated to text, that get sent to Amazon Lex, which determines what intend is the person trying to accomplish. Once Amazon Lex finds a response then it has to use Amazon Polly to translate text into speech, which is what the customer ultimately hears. Checking account balance took 4:30 minutes in the old system and now you can do it in about 25 seconds."

# Amazon Polly: Case Study



## Vodafone New Zealand

Vodafone New Zealand is one of New Zealand's leading digital services and connectivity companies, and we believe every New Zealander will thrive with access to the world's best digital services. We offer more than 3 million connections to Consumer and Business customers, and our existing mobile network covers 98.5% of where Kiwis live, work and play.

"Vodafone is proud to launch Amazon Polly's new Kiwi voice within our call centre, which will answer millions of customer calls monthly. It is important we integrate a unique New Zealand identity into our customer service channels, including te reo Māori phrases and correct pronunciation. It's been great to collaborate with AWS on developing this sophisticated new Kiwi voice, and enable other organisations in Aotearoa New Zealand to adopt it through Amazon Connect and Amazon Polly."

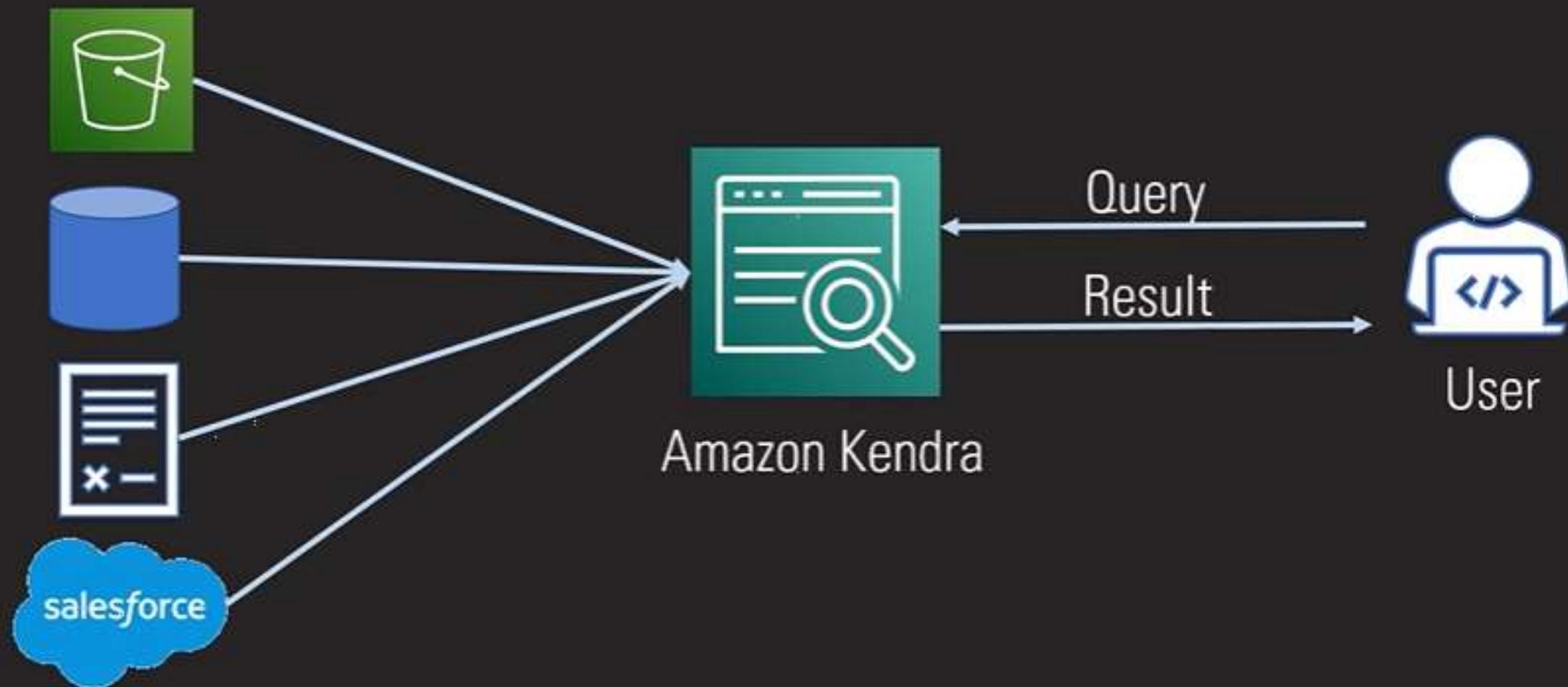
**Mark Fitzgerald, Head of Digital Tools – Vodafone New Zealand**

# Amazon Polly: Lab

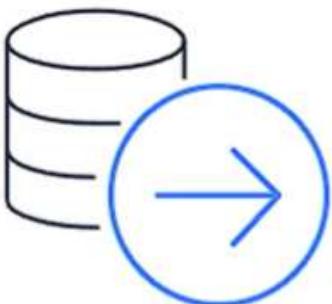
<https://github.com/CloudSihmar/ai-bedrock/blob/main/polly.py>



# Enterprise search service powered by ML



# Amazon Kendra



## Step 1 **Create index**

Create an index where you'll add your data sources.



## Step 2 **Add data sources**

Use Kendra's connectors for popular sources like file systems, web sites, Box, DropBox, Salesforce, SharePoint, relational databases, and Amazon S3.



## Step 3 **Test and deploy**

Test and tune the search experience directly in the console. Plus, access sample code for each component of the search experience so you can easily deploy to new or existing applications.

# Amazon Kendra: Case study



**3M**

3M is a Minnesota-based multinational corporation that produces adhesives, medical products, and much more.

" At 3M, science is at the heart of everything we do. For us, R&D is the heartbeat of 3M; we invest deeply into the science that makes 3M strong. That investment is the key that enables us to introduce over 1200 new products every year. When our material scientists lead new research, they need access to information from prior relevant research—information that's buried in the many patents we hold in our huge knowledge base. Finding the right information is often exhausting (but not exhaustive) and time-consuming. To address this problem, we decided to use Amazon Kendra, a powerful new AWS offering. Kendra lets our scientists find the information they need by handling natural language queries quickly and accurately. With Kendra, we expect our engineers and researchers will find information much faster than they did before we used Amazon Kendra. Our scientists are enthusiastic about this new superpower and we expect them to be able to innovate faster, collaborate more effectively, and accelerate the ongoing stream of unique products for our customers. "

**David Frazee, Technical Director, 3M Corporate Research Systems Lab**

# Amazon Kendra: Case study



## Citibot

Citibot provides tools for citizens and their local and state governments for efficient and effective communication. Citibot helps residents in cities like Fort Worth, Texas get answers to questions, report issues, send messages directly to staff, and receive real-time alerts.

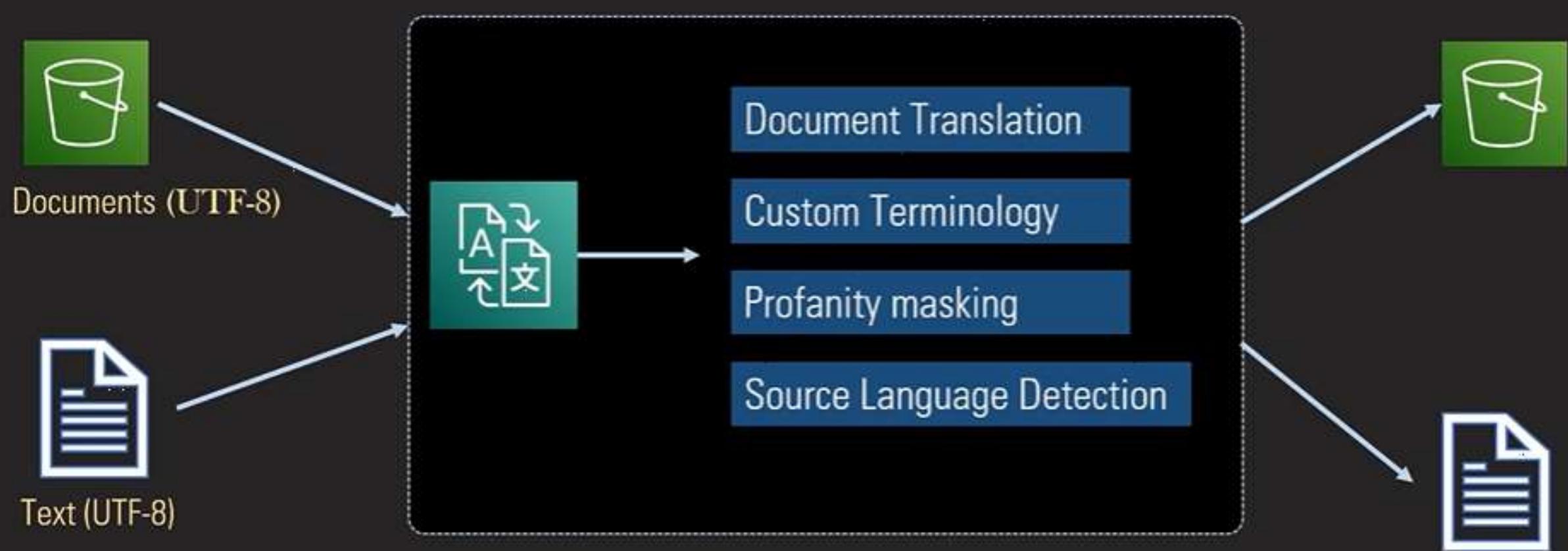
[Read the blog post »](#)

"To help cities better serve these residents, Citibot expanded its Amazon Lex chatbot to integrate with Amazon Kendra-- its ability to answer natural language questions, pulling in data from multiple sources, helps cities improve the scalability and effectiveness of engagement with constituents at a time when it is needed the most. During the COVID-19 outbreak, some cities have been reporting a 50% abandonment rate in their call centers, and some states have reported wait times of up to three hours with ~80% of volume being COVID-19 related. We expect Amazon Kendra to enable citizens to find the answers they seek quickly by engaging with the chatbot and reduce wait times by up to 90%. Post COVID-19, cities will continue to utilize the Amazon Kendra-powered chatbot to provide faster and easier access to more information about things such as public facility hours or road closures due to busy public events."

**Bratton Riley, Founder and CEO of Citibot**



# Language Translation



# Amazon Translate: Use Cases

## **Translate user-generated content**

Automatically translate high volumes of user-generated content, such as social media feed stories, profile descriptions, and comments, in real time.

## **Analyze online conversations in different languages**

Understand sentiments toward your brand, product, or service using a natural language processing (NLP) application that analyzes text in multiple languages.

## **Create cross-lingual communications between users**

Add real-time translation within chat, email, helpdesk, and ticketing applications so an English-speaking agent can communicate with customers across multiple languages.

# Amazon Translate: Case Study

## The BMW Group Streamlines Multilingual Business Processes Using Amazon Translate

BMW group uses Amazon Translate for machine translation of technical and customer facing documents to support its global operations.

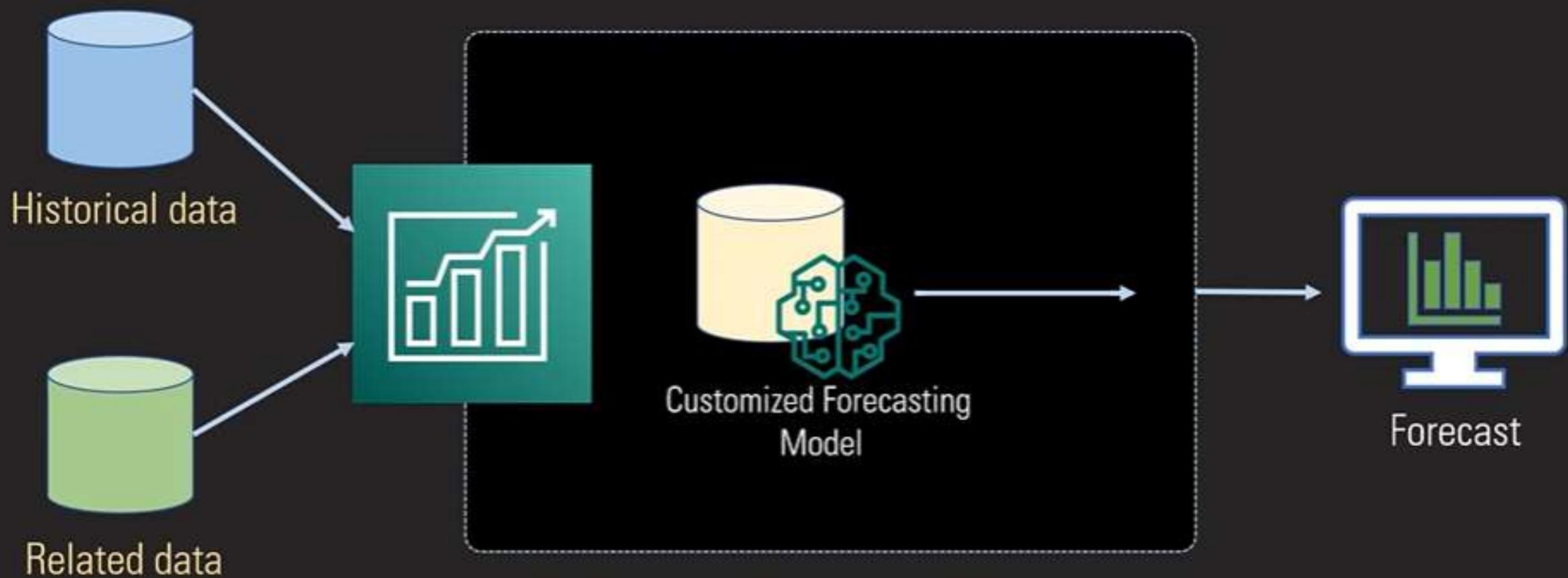
The [BMW Group](#), like many global automobile manufacturers, must manage operations in many different languages. Headquartered in Germany, the company oversees 120,000 employees and assembles cars and motorcycles in 31 manufacturing plants across 15 countries. “With suppliers and production all over the world, we have to provide our documents in numerous languages,” says Fabian Schrickel, group leader of technical translation, standardization, and product conformity at the BMW Group. For the company to effectively communicate with customers, retailers, suppliers, partners, and employees, it needed accurate, high-quality machine translations.

# Amazon Translate: Lab

<https://github.com/CloudSihmar/ai-bedrock/blob/main/translate.py>



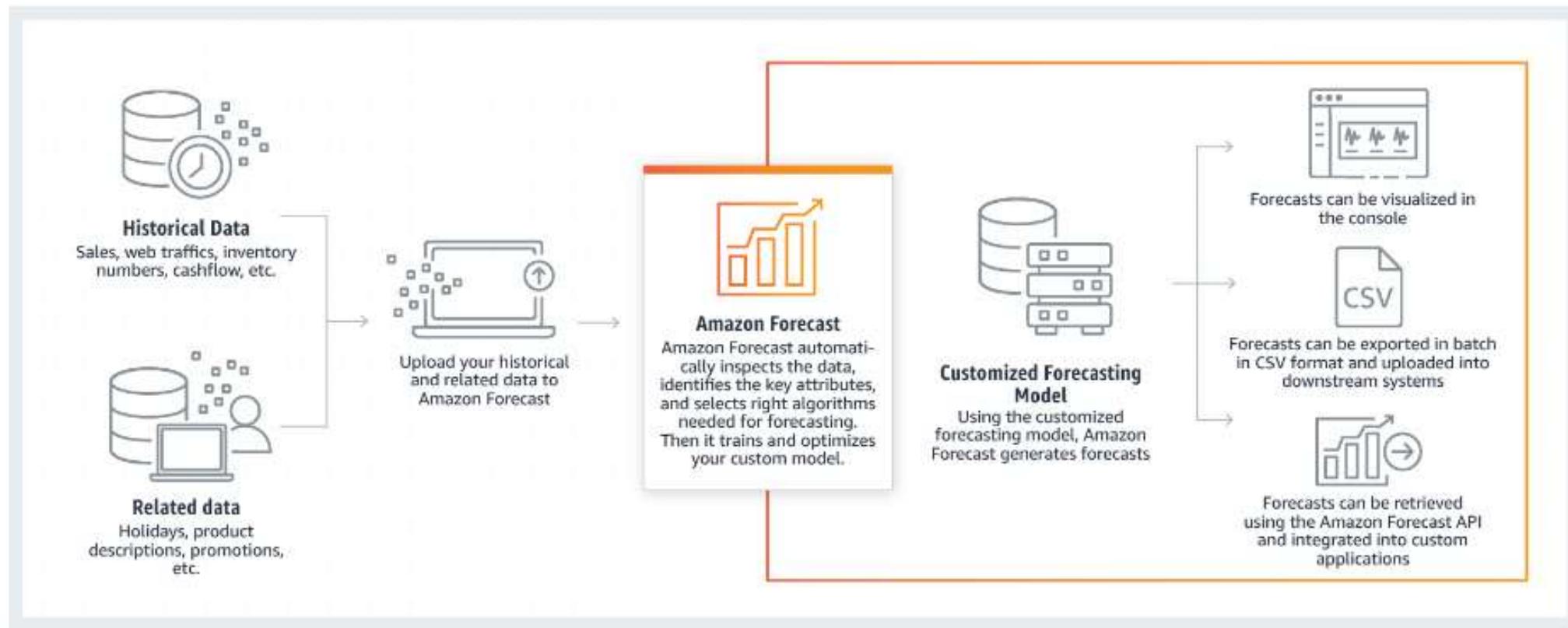
## Forecast business outcomes



# Amazon Forecast: Use cases

- Weather forecasting
- Loan/Credit card account open Forecasting
- Housing prices forecasting
- Retail and inventory forecasting
- Workforce planning
- Travel demand forecasting

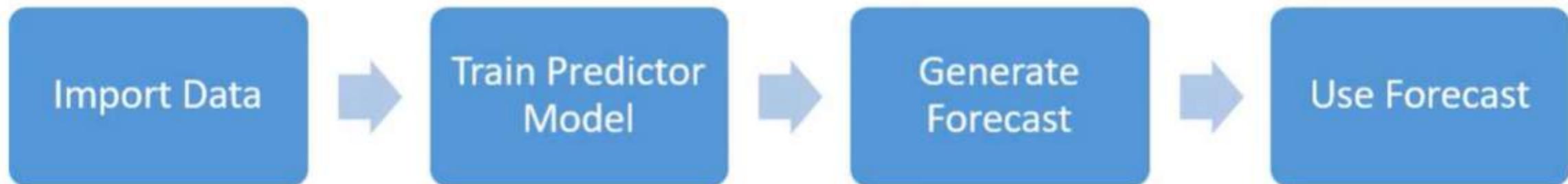
# Amazon Forecast: How it work?



# Forecast Implementation

## Forecast Implementation

- Dataset Groups
  - Import Data, metadata or related data from S3
- Create Predictor
- Create Forecast-Query Forecast
- Create Forecast Export



# Amazon Forecast – Dataset Domains

- **RETAIL**: For retail demand forecasting
- **INVENTORY\_PLANNING**: For supply chain and inventory planning
- **EC2 Capacity**: For forecasting Amazon Elastic Compute Cloud capacity
- **WORK\_FORCE**: For work force planning
- **WEB\_TRAFFIC**: For estimating future web traffic
- **METRICS**: For forecasting metrics, such as revenue and cash flow
- **CUSTOM**: For all other types of time-series forecasting

# Amazon Forecast: Choosing Algorithms

- **ARIMA:** Useful for simple dataset with under 100 time series
- **CNN-QR:** Works best with large datasets containing hundreds of time series.
- **DeepAR+:** Works best with large datasets containing hundreds of feature time series
- **ETS:** Useful for simple datasets with under 100 time series and datasets with seasonality patterns
- **NPTS:** Useful when working with sparse or intermittent time series.
- **Prophet:** Works best with time series with strong seasonal effects and several seasons of historical data

# Amazon Forecast: Dashboard



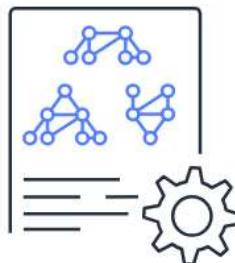
## Import your data

Datasets are required to train predictors, which are then used to generate forecasts.

Target time series data  Active View Edit

Item metadata data [Import](#)

Related time series data [Import](#)

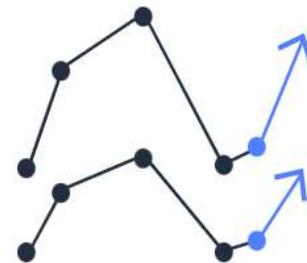


## Train a predictor

Train a predictor, a custom model with underlying infrastructure that Amazon Forecast trains on your datasets.

[View predictors](#)

[Train predictor](#)



## Generate forecasts

Generate forecasts using your trained predictors.

[Create a forecast](#)

[Query forecast](#)



## Explore insights

Identify the attributes impacting your forecasts with explainability and explore different scenarios in a what-if analysis.

[Explore explainability](#)

[Explore what-if analysis](#)

# Forecast Lookup

**Forecast lookup [Info](#)**

When you create a forecast, Amazon Forecast generates forecasts for each unique item in your target time-series dataset. Use the forecast lookup to find your forecasts.

**Forecast details**

Forecast type | Choose the forecast whose forecasts you want to view.  
my\_forecast

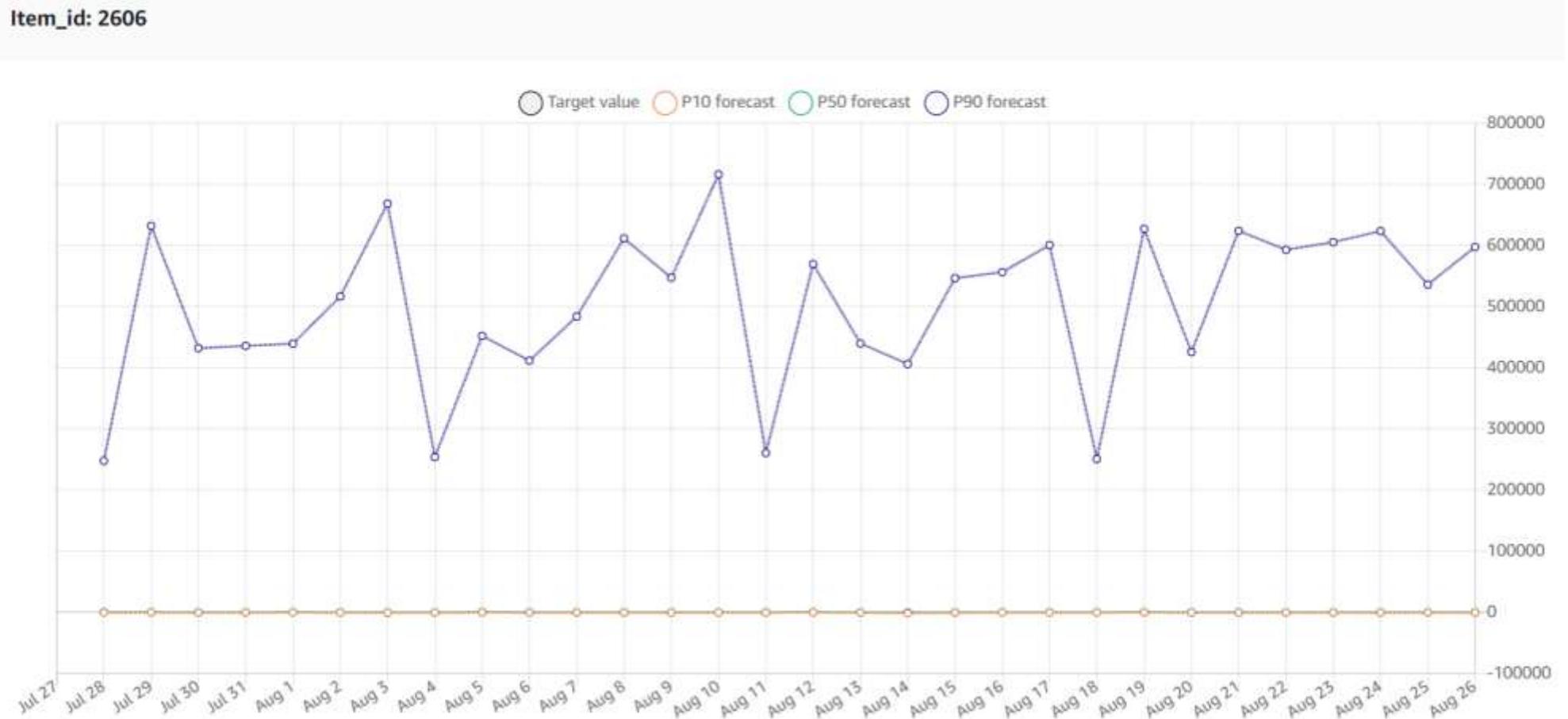
Start date | [Info](#) This is the start date for the historical demand you want to view.  
2019/07/27  00:00:00 Use 24-hour format.

End date | [Info](#) This is the end date for the forecast that you want to view.  
2019/08/26  00:00:00 Use 24-hour format.

Choose the keys you want to use to filter your forecasts:

Forecast key	Value	Remove forecast key
item_id	2606	<input type="button" value="Remove forecast key"/>

# Forecast Lookup



# Forecast Lookup: Case Study



## Meesho

Meesho is India's largest marketplace for longtail/unbranded products and our vision is to enable 100 million small businesses in India to succeed online. The Meesho marketplace provides micro, small and medium businesses, and individual entrepreneurs access to millions of customers, selection from over 100+ categories, pan-India logistics, payment services, and customer support capabilities to efficiently run their businesses on the Meesho ecosystem.

"At Meesho, we have a lot of products with short life spans and it is important for us to react to key metrics associated with product performance and manage our inventory optimally. By using Amazon Forecast, we were able to predict the demand forecasting of products at weekly/daily with an increase in forecasting accuracy of 20% in comparison to our existing solution. Amazon Forecast provides easy to use APIs that helped us to easily build an automated system in half the time it would have taken for an in-house model. We have achieved promising results so far with Amazon Forecast on our current inventory and we plan to continue leveraging it to improve forecasting accuracy on our ever increasing assortment of products."

**Ravindra Yadav, Director, Data Science - Meesho**

# Forecast Lookup: Case Study



## Foxconn

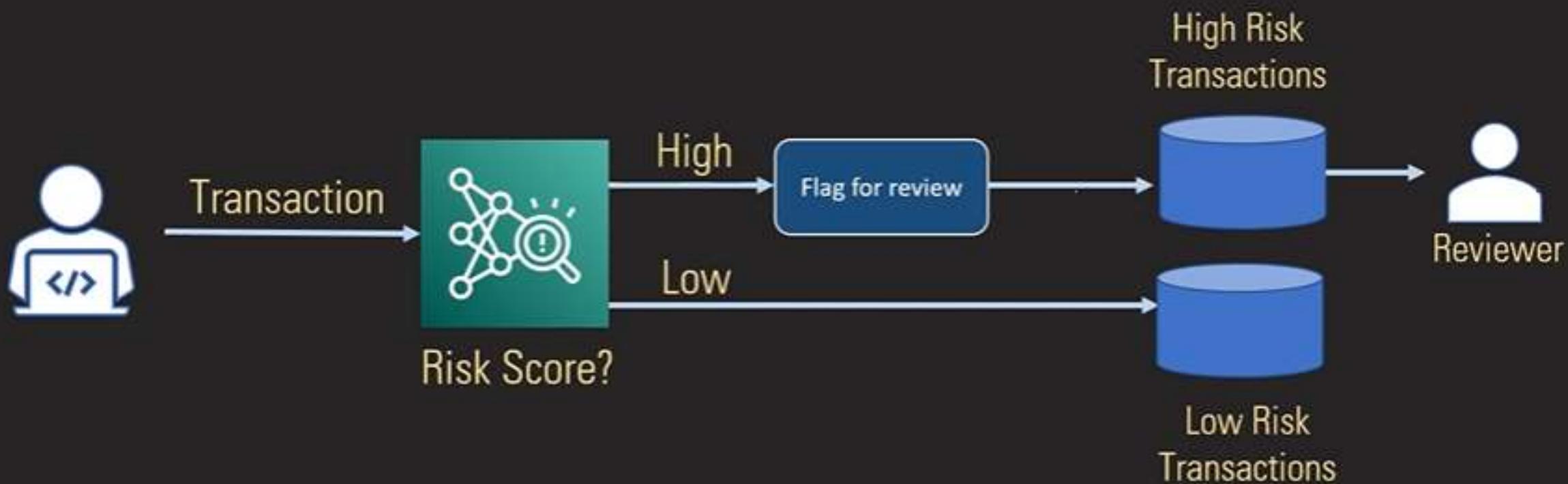
Hon Hai Technology Group (Foxconn) is the world's largest electronics manufacturer and technology solutions provider. During the COVID-19 Pandemic, Foxconn faced unprecedented volatility in customer demand, supplies, and capacity. The company collaborated with the Amazon Machine Learning Solutions Lab to develop a demand forecasting model for its factory in Mexico to generate accurate net order forecasts with a simple API call and input data.

"I was very impressed with the world class machine learning team at AWS. My team worked closely with the Amazon Machine Learning Solutions Lab to develop a demand forecast model using Amazon Forecast within a few weeks. Our solution increased our forecasting accuracy by 8%. We project \$553K annual savings using this solution for our factory in Mexico. As a bonus, it will be easy to integrate this solution into our cloud workflow once we migrate our data infrastructure to AWS. This collaboration with AWS helped minimize wasted labor costs and maximize customer satisfaction."

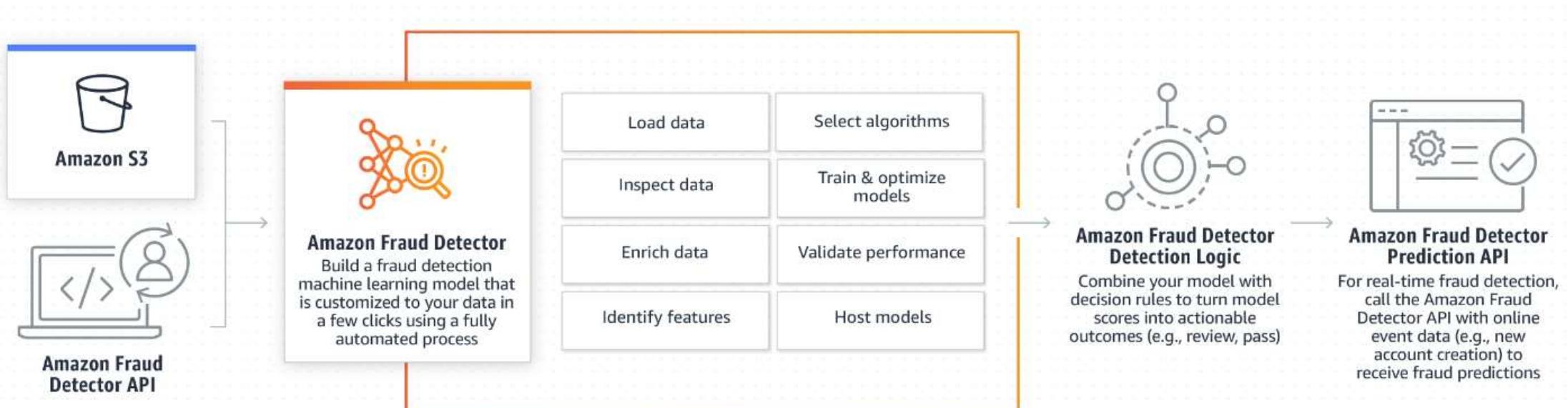
**Azim Siddique, Technical Advisor and CoE Architect - Foxconn**



# Detect online fraud



# Amazon Fraud Detector



# Amazon Fraud Detector: Use cases

- Identify suspicious online payments
- Detect new account fraud
- Prevent trial and loyalty program abuse
- Improve account takeover detection

# Amazon Fraud Detector



# Amazon Fraud Detector: Case Study



## GoDaddy

GoDaddy is the world's largest services platform for entrepreneurs around the globe and is on a mission to empower their worldwide community of 19+ million customers and entrepreneurs everywhere by giving them all the help and tools they need to grow online.

"GoDaddy is committed to preventing fraudulent accounts and we're continually bolstering our capabilities to automatically detect such accounts during sign-up. We recently began using Amazon Fraud Detector and we're pleased that it offers low cost of implementation and a self-service approach to building a machine learning model that is customized to our business. The model can be easily deployed and used in our new account process without impacting the signup experience for legitimate customers. The model we built with Amazon Fraud Detector is able to detect likely fraudulent sign-ups immediately, so we're very pleased with the results and look forward to accomplishing more."

**John Kercheval, Senior Director, Identity Services Group - GoDaddy**

# Amazon Fraud Detector: Case Study



## Standard Bank Insurance

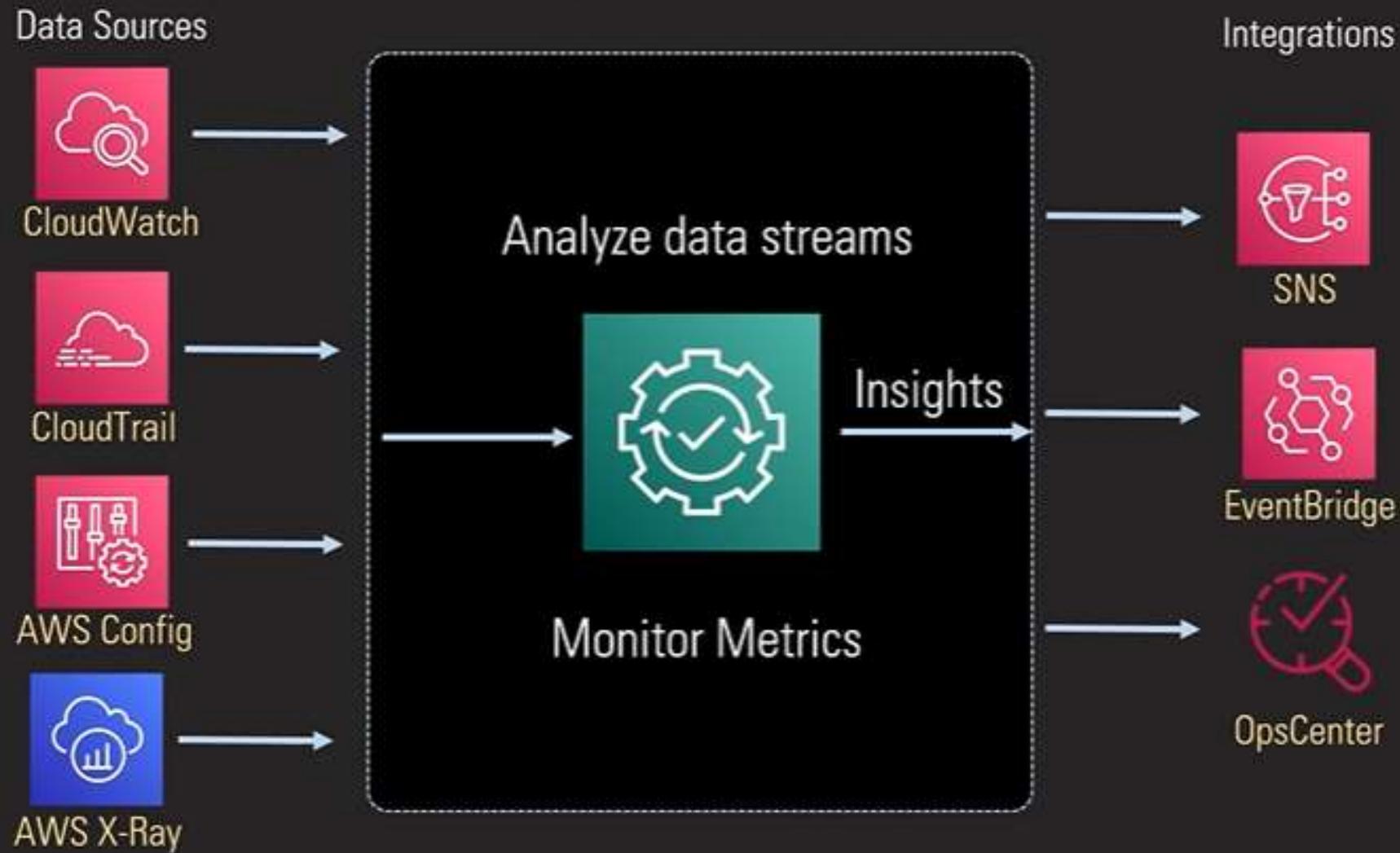
Standard Bank is a major South African financial services group that has been in business for over 160 years and is Africa's largest lender by assets. They offer a range of products and services which include investment solutions, home loans, vehicle and asset finance, and insurance.

"Since we launched Amazon Fraud Detector into production, our results have been great. Approximately 94% of our claims are typically rated as low risk and for these claims, the turnaround time has already reduced from 48 hours in February with our traditional manual process to less than 6 hours by the end of August. This has resulted in improved customer experience. We have also seen a 36% increase in our Net Promoter Survey (NPS) scores between February 2022 to August 2022 since going live in production. We attribute this to the faster payouts due to safely automating approvals for low-risk claims. For the 6% of claims rated as high risk, we now have more capacity to interrogate these better than before. As a result, we are able to thoroughly investigate suspected cases and stop more claims with actual fraud. Overall our confirmed fraudulent cases that we have been able to identify before paying out a claim has increased over 100%, which has greatly reduced the business' exposure to risk."

– Ashia Bowers, Head of Automation, Standard Bank Insurance



# Identify abnormal application behavior using ML



# Devops Services and Tools



# Key Challenges



Data volumes are large and disparate



When an issue occurs, users spend a lot of time and effort manually correlating across data sources and tools



As new services are adopted, deep DevOps/CloudOps expertise is required to configure new alarms and monitoring procedures

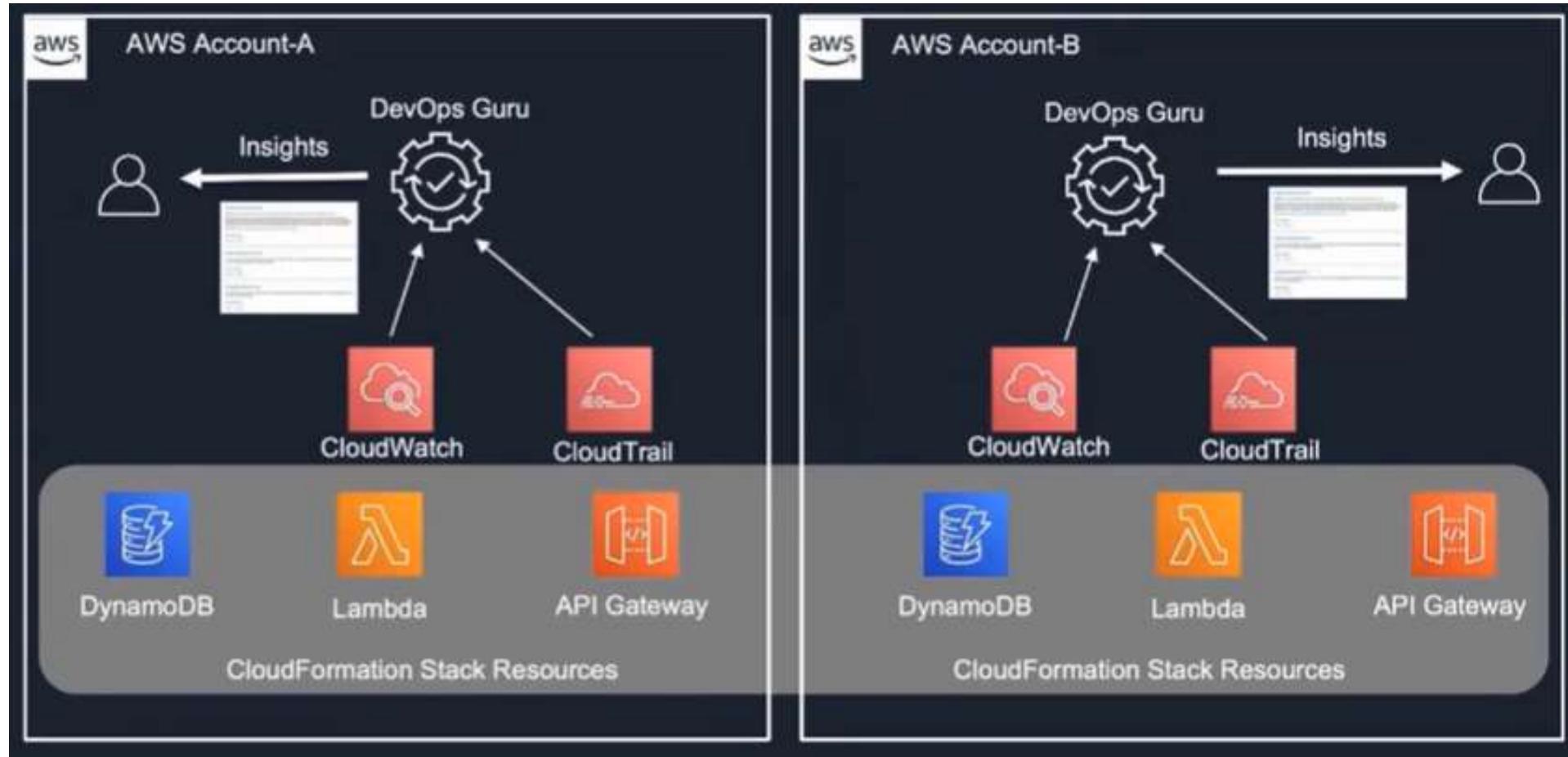


Alarms and notifications from multiple tools cause alarm fatigue and an inability to identify the most critical issue

# Devops Guru

- DevOps Guru is an ML-powered service that makes it easy for developers and operators to automatically detect issues to improve application availability and reduce expensive downtime
- No machine learning experience required
- Offloads operators from setting alarms, thresholds
- Prevent generation of tons of alarms
- Prescribed guidance via insights

# DevOps Guru Insights



# DevOps Guru: Why to use

- Easy to use: No configuration or ML required
- Auto detect operational issues: DevOps Guru continuously analyzes streams of disparate data and metrics to determine application behavior
- Quick resolution: Helps to resolve issues quickly with ML-powered insights and recommendations
- Scalable: Easily scales and maintains availability as new AWS workloads are added
- Reduce noise: Helps overcome alarm fatigue by automatically correlating and grouping related anomalies

# DevOps Guru: Case Study



HCL Technologies is a next-generation global technology company that helps enterprises reimagine their businesses for the digital age. Our technology products and services are built on four decades of innovation, with a world-renowned management philosophy, a strong culture of invention and risk-taking, and a relentless focus on customer relationships.

"We are always looking for ways to reduce the amount of time our teams spend on resolving operational issues, and we are now using Amazon DevOps Guru and leveraging its ML-powered insights to help us identify, correlate, and remediate operational issues quickly. With the insights Amazon DevOps Guru provides, our teams can now quickly find issues without having to start from scratch trying to root cause problems. Our IT team has significantly reduced our mean time to recovery (MTTR), and they are saving hours upon hours of time resolving issues—all the while ensuring our customers have the best end-user experience possible."

**Anchal Gupta, Senior Technical Lead, DevOps, HCL Technologies**

# DevOps Guru: Case Study



PagerDuty Inc. (NYSE:PD) is a leader in digital operations management.

"PagerDuty is further deepening our partnership with AWS with a new integration with Amazon DevOps Guru. PagerDuty's digital operations management platform was built to drive a shift to DevOps culture and we are delighted to continue this commitment with this integration. Harnessing DevOps Guru's ML capabilities, PagerDuty provides even more real-time signal-to-action capabilities to our joint customers. Through PagerDuty's ingestion of Amazon DevOps Guru's Amazon SNS, AWS customers can take real-time action on operational issues before they become customer-impacting outages."

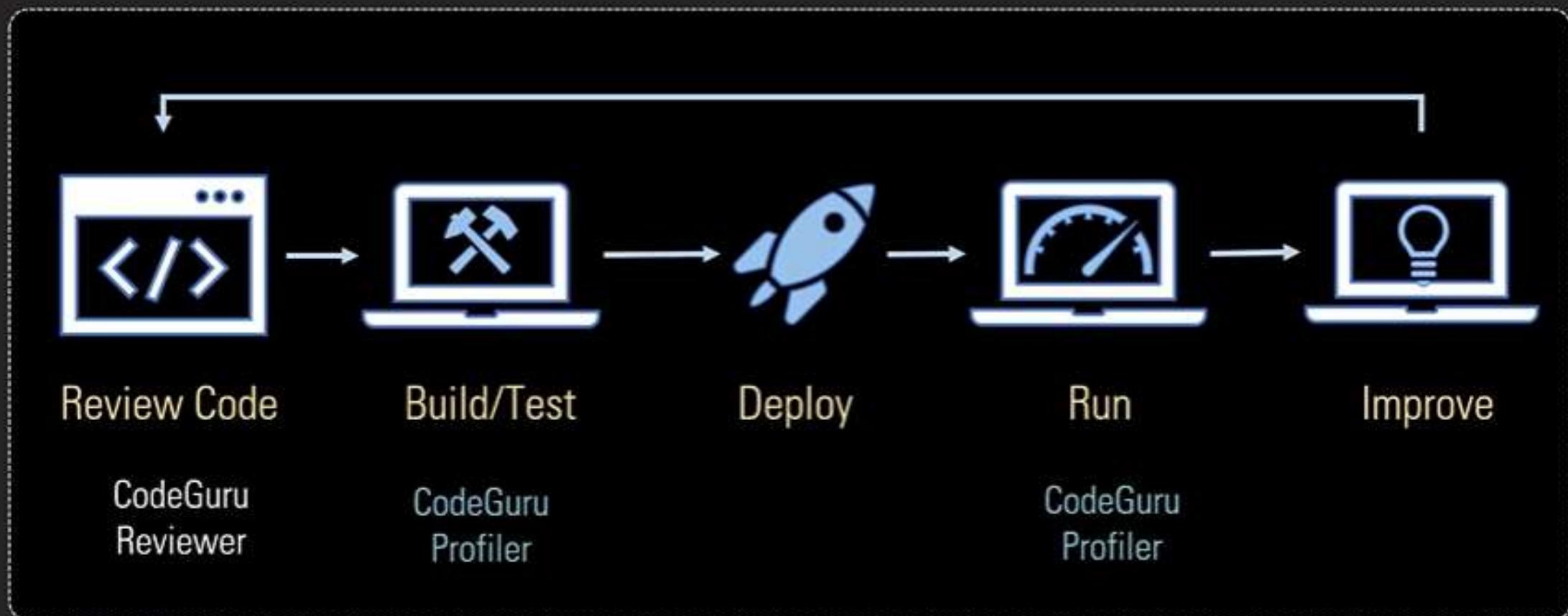
**Jonathan Rende, SVP of Product, PagerDuty**

# DevOps Guru: Demo

<https://github.com/CloudSihmar/amazon-devops-guru.git>



# Recommendations on performance issues



# Amazon CodeGuru

- Code quality is a growing concern for enterprises building cloud-native applications.
- Amazon CodeGuru is a developer tool from AWS that lets developers identify their application's most expensive lines.
- The tool then provides recommendations for remediation.
- CodeGuru also monitors application performance in production.
- It then delivers recommendations to developers about how they can further improve code quality and application performance while helping reduce your cloud costs.

# Amazon CodeGuru: Services

## 1. CodeGuru Profiler

- CodeGuru Profiler lets your DevOps team visualize **application performance** from a centralized dashboard.
- Here, they can **troubleshoot application latency** and **CPU utilization** problems. It also allows your team to learn where they can reduce the infrastructure costs for an application.

# Amazon CodeGuru: Services

## 2. CodeGuru Reviewer

- CodeGuru Reviewer lets your DevOps team detect defects in **software code** using **machine learning**.
- It offers suggestions for improving your Java and Python code as part of its analysis and guidelines on addressing the defects.
- CodeGuru Reviewer works with the following Java and Python code repositories: AWS CodeCommit, Amazon S3, Bitbucket, GitHub, GitHub Enterprise Cloud and GitHub Enterprise Server.

# Amazon CodeGuru: Services

## **3. CodeGuru Security**

- CodeGuru Security is a static application security testing (SAST) tool that combines Machine Learning and Automated Reasoning to identify vulnerabilities in your code, provide recommendations on how to fix the identified vulnerabilities, and track the status of the vulnerabilities until closure.

# CodeGuru Benefits and Features

- Reducing the time you spend reviewing and triaging findings and receive in-context code patches for certain classes of vulnerabilities.
- See where your application spends its time, and get recommendations to improve the performance and cost of your applications.
- Run automated code reviews and get recommendations to improve the quality of your source code.

# CodeGuru: Case Study



Cognizant is one of the world's leading professional services companies, transforming clients' business, operating and technology models for the digital era.

"Amazon CodeGuru helps Cognizant development teams deliver mission critical software for our customer's digital transformation programs. Incorporating CodeGuru in our development workflows improves and automates code reviews, helps our DevOps teams proactively identify and fix functional and non-functional issues and ensures that the deployments exceeds the performance, security and compliance requirements of our customers across industries and regions."

**Todd Carey, Global Head, Cognizant AWS Business Group**

# Responsible AI

# Responsible AI

- Developing responsible AI requires an understanding of the possible issues, limitations or unintended consequences.
- Responsible Artificial Intelligence (Responsible AI) is an approach to developing, assessing, and deploying AI systems in a safe, trustworthy, and ethical way.
- AI systems are the product of many decisions made by those who develop and deploy them. From system purpose to how people interact with AI systems, Responsible AI can help proactively guide these decisions toward more beneficial and equitable outcomes.
- That means keeping people and their goals at the center of system design decisions and respecting enduring values like fairness, reliability, and transparency.

# Responsible AI



Fairness



Reliability  
& Safety



Privacy &  
Security



Inclusiveness



Transparency



Accountability

# Responsible AI Principles

- Accurate & reliable
- Accountable & transparent
- Fair & human-centric
- Safe & ethical
- Secure & resilient
- Interpretable & documented
- Privacy-enhanced & data governed

# Future Potential

# Future Potential & Jobs Perspective

- **Job Displacement:** The widespread adoption of genAI may lead to the displacement of certain jobs, particularly those involving repetitive tasks or routine decision-making processes.
- **New Job Opportunities:** However, genAI also creates new job opportunities, such as AI ethicists, data privacy specialists, AI trainers, and AI system auditors.
- **Skill Requirements:** The rise of genAI emphasizes the importance of acquiring skills in data science, machine learning, and AI ethics for professionals across various industries.
- **Human-AI Collaboration:** Rather than replacing human workers entirely, genAI is likely to augment human capabilities, leading to more collaboration between humans and AI systems in various domains.

# Ethical Considerations and Future Trends

- **Transparency:** It's crucial to ensure that genAI systems are transparent in their operations, allowing users to understand how they function and make decisions.
- **Fairness:** GenAI algorithms should be designed and trained to avoid biases and discrimination, especially in sensitive areas like hiring, lending, and criminal justice.
- **Accountability:** There should be mechanisms in place to hold developers and users of genAI accountable for any unethical or harmful outcomes resulting from its use.
- **Privacy:** Safeguarding individuals' privacy is paramount when using genAI, especially in applications involving personal data generation and manipulation.
- **Security:** GenAI systems must be resilient against attacks and misuse to prevent unauthorized access, manipulation, or malicious activities.

# Launching a Generative AI for your business involves several key steps to ensure its successful integration and utilization

## 1. Identify Business Areas

Start by identifying the areas of your business where Generative AI can have the most impact. This could include:

- **Marketing and advertising:** Creating personalized content, generating ad copy, or designing visuals.
- **Product development:** Generating new product ideas, designs, or prototypes.
- **Customer service:** Automating responses to common inquiries or generating personalized recommendations.
- **Data analysis:** Generating insights from large datasets or predicting trends.
- Analyze your business processes and determine where AI-driven generative capabilities can streamline operations, improve efficiency, or enhance customer experiences.

## 2. Creating a Roadmap

Develop a roadmap that outlines the implementation of Generative AI within your organization. Consider the following steps:

- **Define objectives:** Clearly articulate the goals you aim to achieve with Generative AI.
- **Assess readiness:** Evaluate your organization's readiness for adopting Generative AI, including infrastructure, resources, and skillsets.
- **Identify use cases:** Prioritize use cases based on their potential impact and feasibility of implementation.
- **Develop a timeline:** Establish a timeline for each phase of implementation, considering factors like development, testing, and deployment.
- **Allocate resources:** Determine the resources required, including budget, personnel, and technology.
- **Risk assessment:** Identify potential risks and challenges associated with implementation and develop mitigation strategies.
- **Monitoring and evaluation:** Plan for ongoing monitoring and evaluation to track progress and make necessary adjustments.

### 3. Identify Core Components

When implementing Generative AI, consider the following core components:

- **Data:** Ensure access to high-quality, relevant data that will be used to train and refine the Generative AI models.
- **Algorithms:** Select or develop algorithms that are suited to the specific use cases identified for your business.
- **Infrastructure:** Establish the necessary infrastructure to support Generative AI, including computing resources and storage capabilities.
- **Integration:** Integrate Generative AI into existing systems and workflows, ensuring compatibility and seamless operation.
- **Governance and compliance:** Implement measures to ensure ethical use of Generative AI and compliance with relevant regulations and standards.
- **Training and support:** Provide training to employees on how to effectively use Generative AI tools and offer ongoing support to address any issues or challenges that arise.

**Human capital** plays a crucial role in the successful adoption and implementation of Generative AI (GenAI) within an organization.

## 1. Importance of the Right People for GenAI Adoption

- Expertise: GenAI adoption requires individuals with expertise in various domains, including data science, machine learning, software engineering, and domain-specific knowledge related to the business.
- Innovation mindset: The right people should possess an innovative mindset and be open to exploring new technologies and approaches to problem-solving.
- Collaboration: Collaboration between different teams and departments is essential for successful GenAI adoption. Individuals who can work effectively in interdisciplinary teams and communicate across different levels of the organization are valuable.
- Adaptability: GenAI technology is rapidly evolving, so individuals who are adaptable and willing to continuously learn and upgrade their skills are essential for long-term success.
- Ethical considerations: Given the potential ethical implications of GenAI, having individuals who are knowledgeable about ethical considerations and can ensure responsible use of AI is crucial.

## 2. Roles and Responsibilities in GenAI Adoption

- Data scientists: Responsible for developing and training Generative AI models using relevant data and algorithms.
- Software engineers: Design and implement the infrastructure required for deploying and integrating Generative AI solutions into existing systems.
- Domain experts: Provide domain-specific knowledge and insights to guide the development and application of Generative AI in relevant business areas.
- Project managers: Oversee the planning, execution, and monitoring of GenAI adoption projects, ensuring alignment with organizational goals and timelines.
- Ethicists and compliance specialists: Ensure that GenAI adoption adheres to ethical standards and regulatory requirements, mitigating risks associated with data privacy, bias, and fairness.
- Change management specialists: Facilitate organizational change and adoption by addressing cultural and behavioral barriers, providing training, and fostering a supportive environment for GenAI integration.

### 3. Organizational Models for GenAI Adoption

- Centralized model: In this model, a central team or department is responsible for overseeing all aspects of GenAI adoption, including development, deployment, and governance. This approach ensures consistency and centralized expertise but may lead to silos and slower decision-making.
- Decentralized model: In a decentralized model, individual business units or departments have autonomy over GenAI adoption within their respective domains. This approach allows for flexibility and customization but may result in duplication of efforts and inconsistent standards.
- Hybrid model: Combining elements of both centralized and decentralized models, the hybrid model strikes a balance between standardization and autonomy. Centralized governance and expertise support decentralized implementation, fostering collaboration and knowledge sharing across the organization.

# Process for adopting Generative AI

## 1. Understanding Key Elements

- Data: Identify and collect relevant data sources necessary for training Generative AI models.
- Algorithms: Select appropriate algorithms and techniques for generating AI-driven outputs based on the desired outcomes and use cases.
- Infrastructure: Establish the necessary computing resources, storage capabilities, and software tools to support the development and deployment of Generative AI solutions.
- Expertise: Assemble a multidisciplinary team with expertise in data science, machine learning, software engineering, domain knowledge, and ethics to drive GenAI adoption.

## 2. GenAI Strategy

- Define Objectives: Clearly articulate the business goals and objectives that GenAI adoption aims to achieve, such as improving efficiency, enhancing customer experiences, or driving innovation.
- Use Case Prioritization: Identify and prioritize use cases based on their potential impact, feasibility, and alignment with organizational priorities.
- Roadmap Development: Develop a roadmap outlining the implementation timeline, resource allocation, and milestones for GenAI adoption, ensuring alignment with broader organizational strategies.

### 3. Responsible AI

- Ethical Considerations: Evaluate the ethical implications of GenAI adoption, including issues related to privacy, bias, fairness, transparency, and accountability.
- Governance Framework: Establish governance mechanisms and policies to ensure responsible and ethical use of Generative AI, including data governance, model governance, and compliance with regulatory requirements.
- Ethical AI Education: Provide training and education to employees on ethical considerations and responsible AI practices to promote awareness and adherence to ethical standards throughout the organization.

## 4. Risk Assessments

- Identify Risks: Conduct a thorough risk assessment to identify potential risks and challenges associated with GenAI adoption, including technical, ethical, legal, and societal risks.
- Risk Mitigation: Develop mitigation strategies to address identified risks, such as implementing safeguards to prevent bias in AI models, enhancing data security measures, and ensuring compliance with regulatory requirements.

## 5. Development Process and Management

- Iterative Development: Adopt an iterative development approach to continuously refine and improve Generative AI models based on feedback and real-world performance.
- Project Management: Implement project management practices to plan, execute, and monitor GenAI adoption projects, including defining deliverables, timelines, and resource allocation.
- Collaboration and Communication: Foster collaboration and communication among cross-functional teams involved in GenAI development and deployment to ensure alignment with organizational goals and objectives.