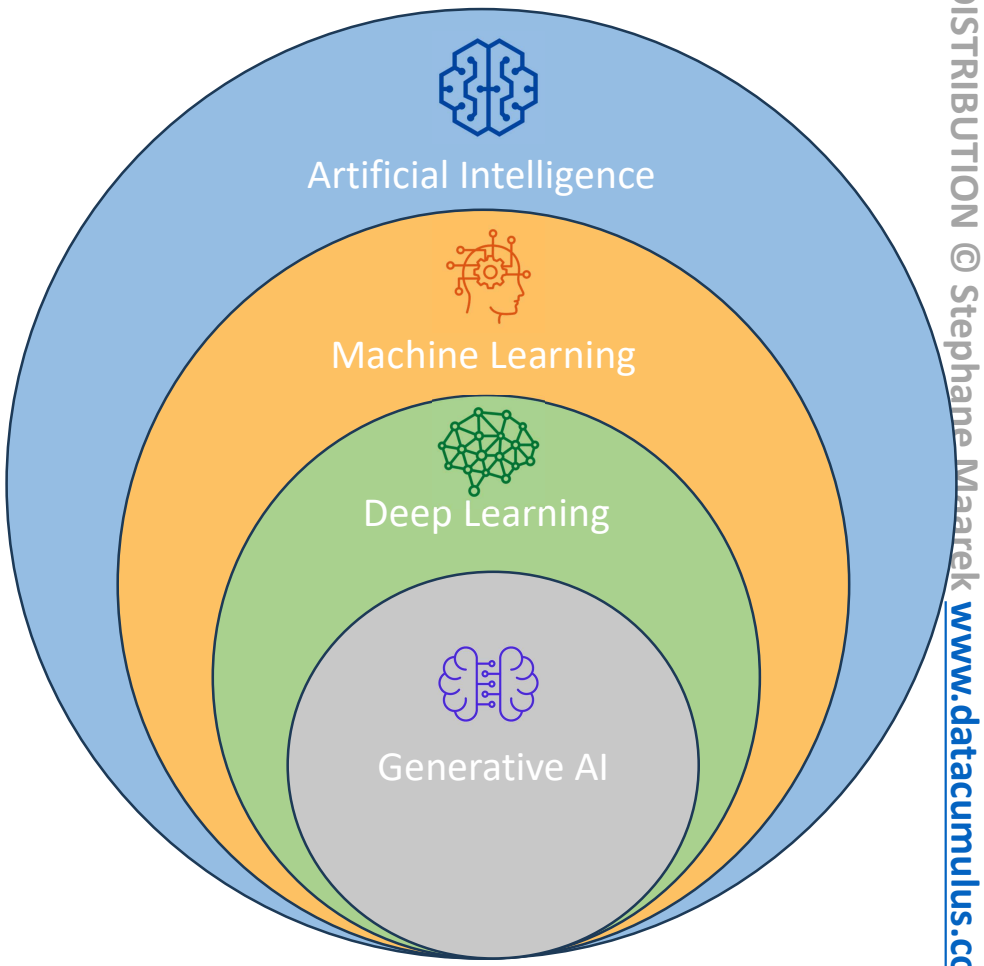
AI and Machine Learning Overview

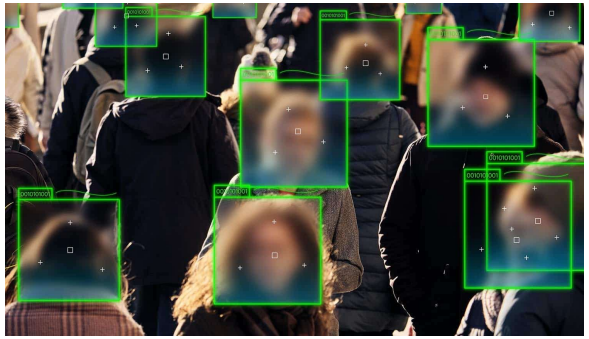
What is Artificial Intelligence (AI)?

- AI is a broad field for the development of intelligent systems capable of performing tasks that typically require human intelligence:
  - Perception
  - Reasoning
  - Learning
  - Problem solving
  - Decision-making
- Umbrella-term for various techniques

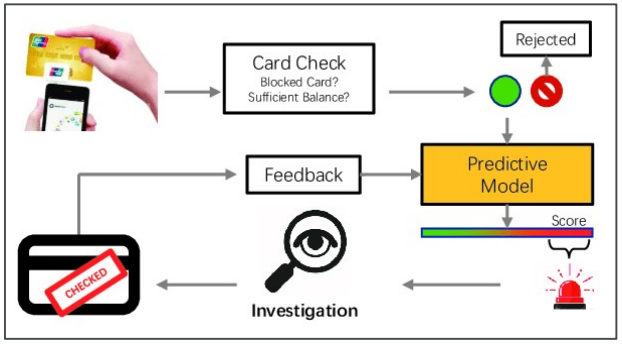


Artificial Intelligence – Use Cases

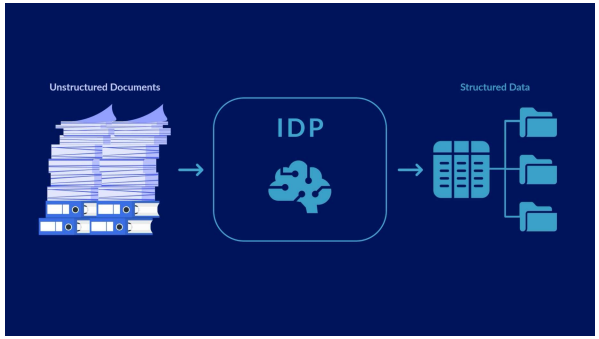
Computer Vision



Facial Recognition



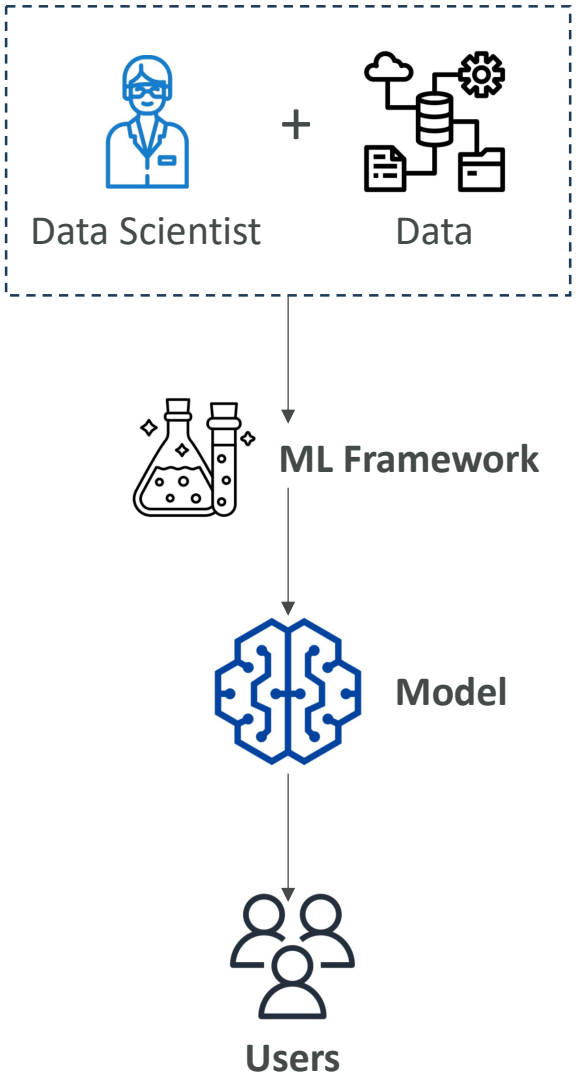
Fraud Detection



Intelligent Document Processing (IDP)

AI Components

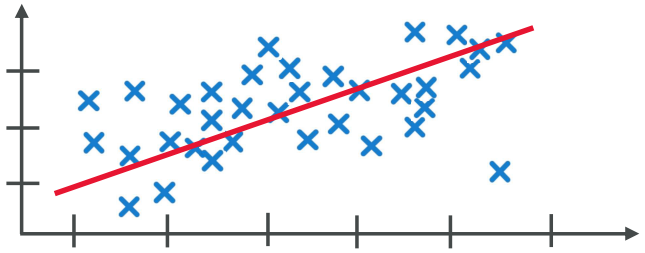
- **Data Layer** – collect vast amount of data
- **ML Framework and Algorithm Layer** – data scientists and engineer work together to understand use cases, requirements, and frameworks that can solve them
- **Model Layer** – implement a model and train it, we have the structure, the parameters and functions, optimizer function
- **Application Layer** – how to serve the model, and its capabilities for your users



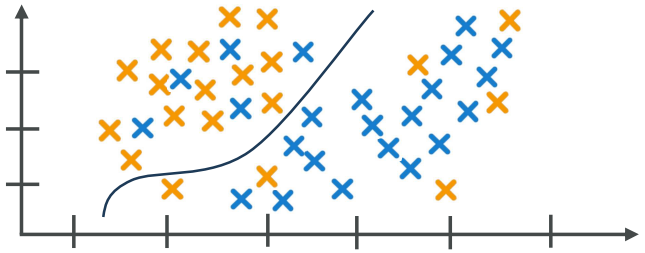
What is Machine Learning (ML)?

- ML is a type of AI for building methods that allow machines to learn
- **Data** is leveraged to improve computer performance on a set of task
- Make predictions based on data used to train the model
- No explicit programming of rules

Regression

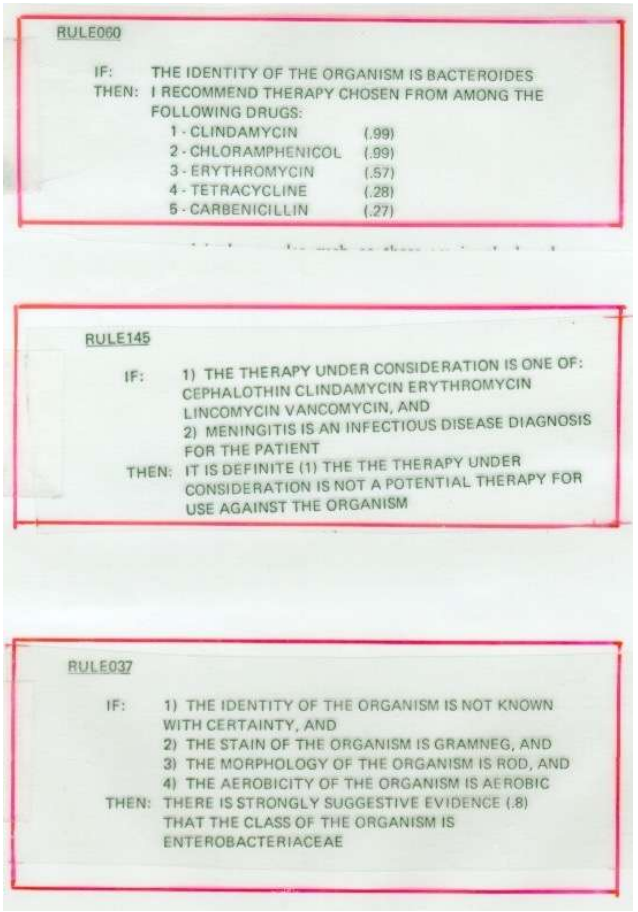


Classification



AI != ML
Ex: MYCIN Expert System

- System developed in 1970s to diagnose patients based on reported symptoms and medical test results
- Collection of over 500 rules
- Simple yes/no or textual questions
- It provides a list of culprit bacteria ranked from high to low based on the probability of diagnosis, the reason behind the diagnosis, and a potential dosage for the cure
- Never really used in production as personal computers didn't exist yet



RULE060

IF: THE IDENTITY OF THE ORGANISM IS BACTEROIDES
THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE FOLLOWING DRUGS:
1 - CLINDAMYCIN (.99)
2 - CHLORAMPHENICOL (.99)
3 - ERYTHROMYCIN (.57)
4 - TETRACYCLINE (.28)
5 - CARBENICILLIN (.27)

RULE145

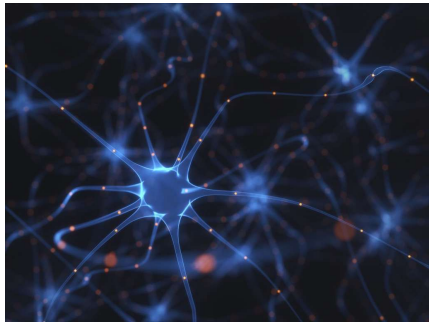
IF: 1) THE THERAPY UNDER CONSIDERATION IS ONE OF: CEPHALOTHIN CLINDAMYCIN ERYTHROMYCIN LINCOMYCIN VANCOMYCIN, AND
2) MENINGITIS IS AN INFECTIOUS DISEASE DIAGNOSIS FOR THE PATIENT
THEN: IT IS DEFINITE (1) THE THE THERAPY UNDER CONSIDERATION IS NOT A POTENTIAL THERAPY FOR USE AGAINST THE ORGANISM

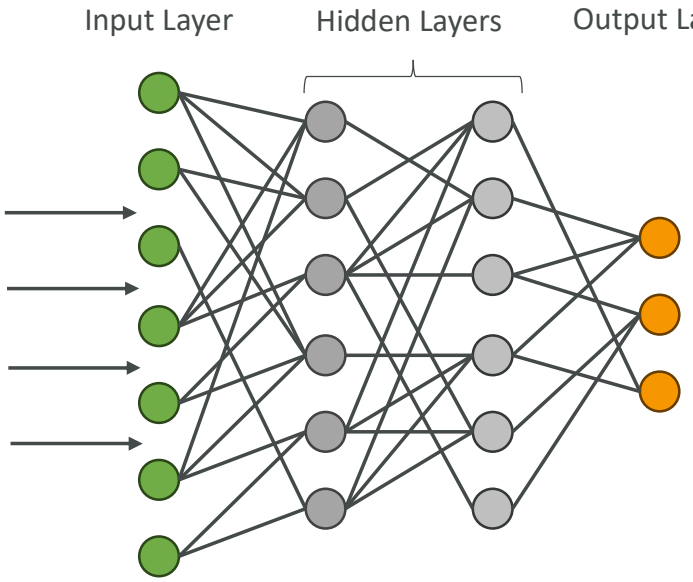
RULE037

IF: 1) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND
2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
4) THE AEROBICITY OF THE ORGANISM IS AEROBIC
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE CLASS OF THE ORGANISM IS ENTEROBACTERIACEAE

What is Deep Learning (DL)?

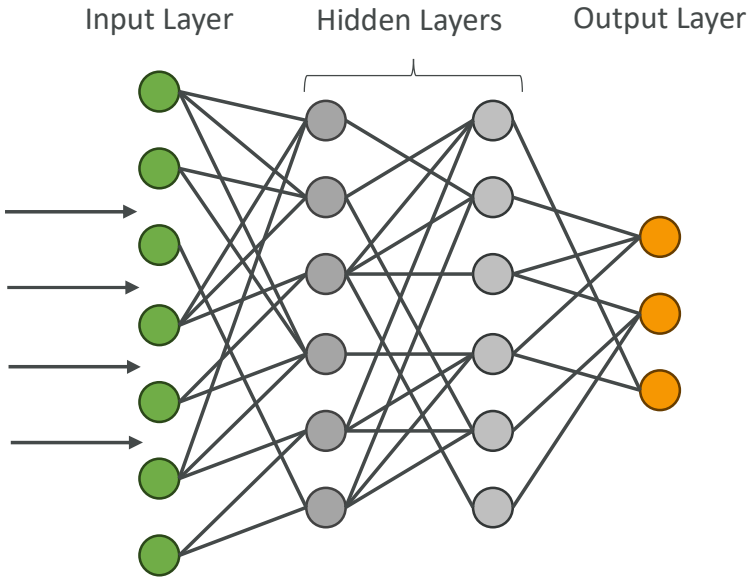
- Uses neurons and synapses (like our brain) to train a model
- Process more complex patterns in the data than traditional ML
- **Deep** Learning because there's more than one layer of learning
- **Ex: Computer Vision** – image classification, object detection, image segmentation
- **Ex: Natural Language Processing (NLP)** – text classification, sentiment analysis, machine translation, language generation
- Large amount of input data
- Requires GPU (Graphical Processing Unit)



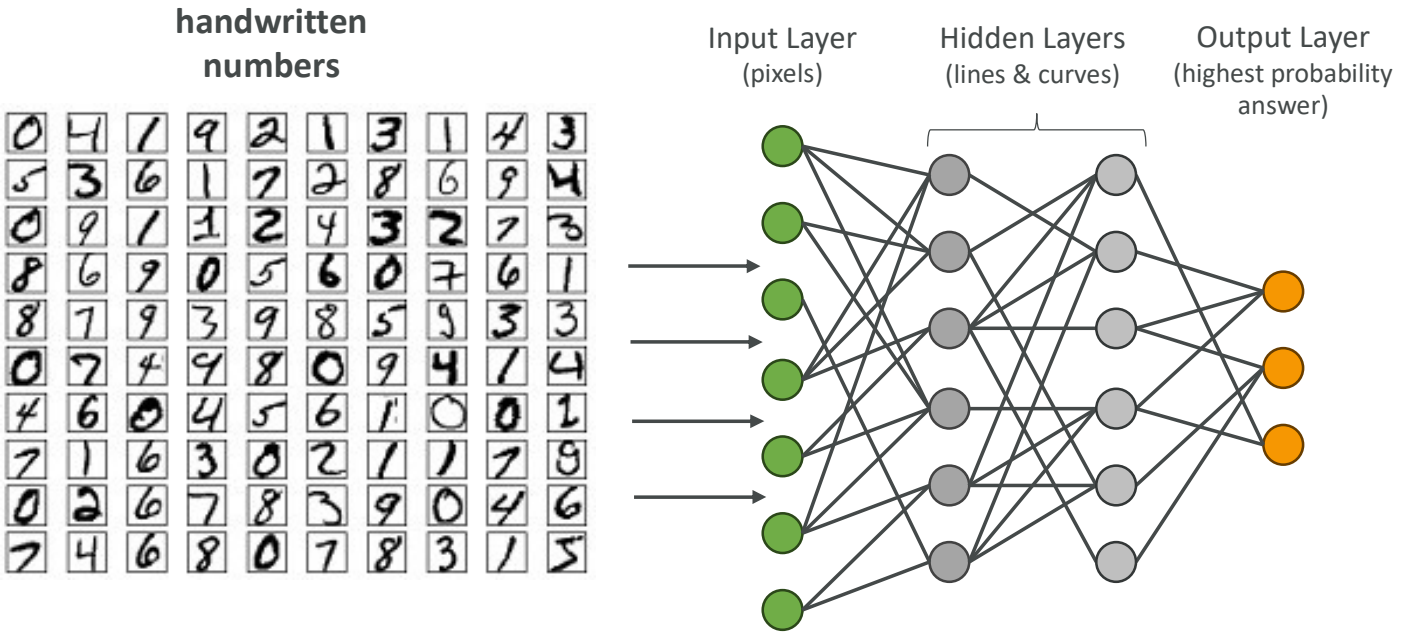


Neural Networks – how do they work?

- Nodes (tiny units) are connected together
- Nodes are organized in layers
- When the neural network sees a lot of data, it identifies patterns and changes the connections between the nodes
- Nodes are "talking" to each other, by passing on (or not) data to the next layer
- The math and parameters tuning behind it is beyond the level of this course
- Neural networks may have billions of nodes



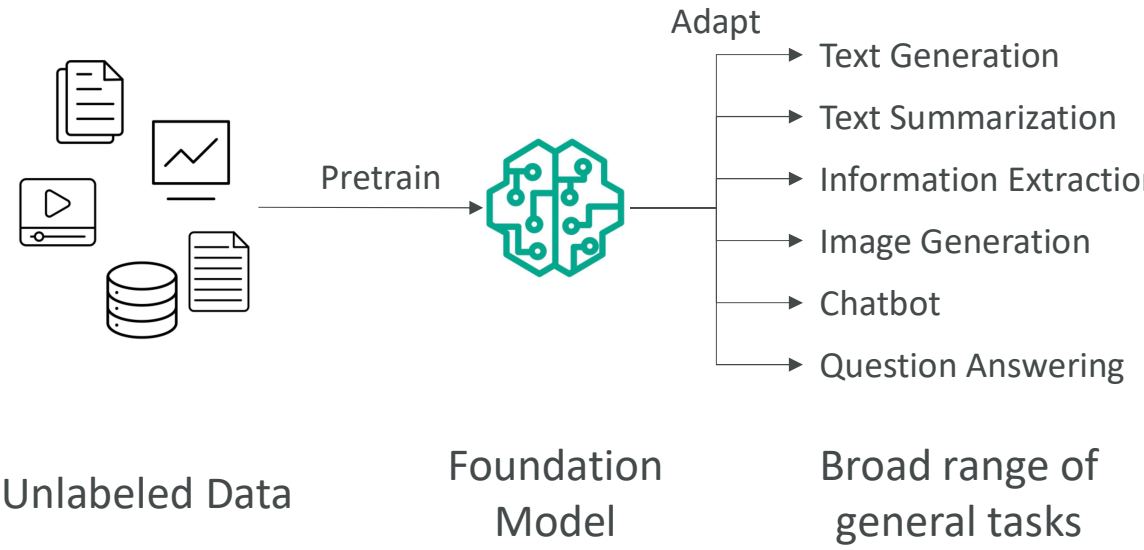
Deep Learning Example
Recognizing hand-written digits



- Intuitively: each layer will learn about a "pattern" in the data
- Example: vertical lines for a 1, 4, 7
- Example: curved bottom for 6, 8, 0
- But this is all "learned" by the Neural Network

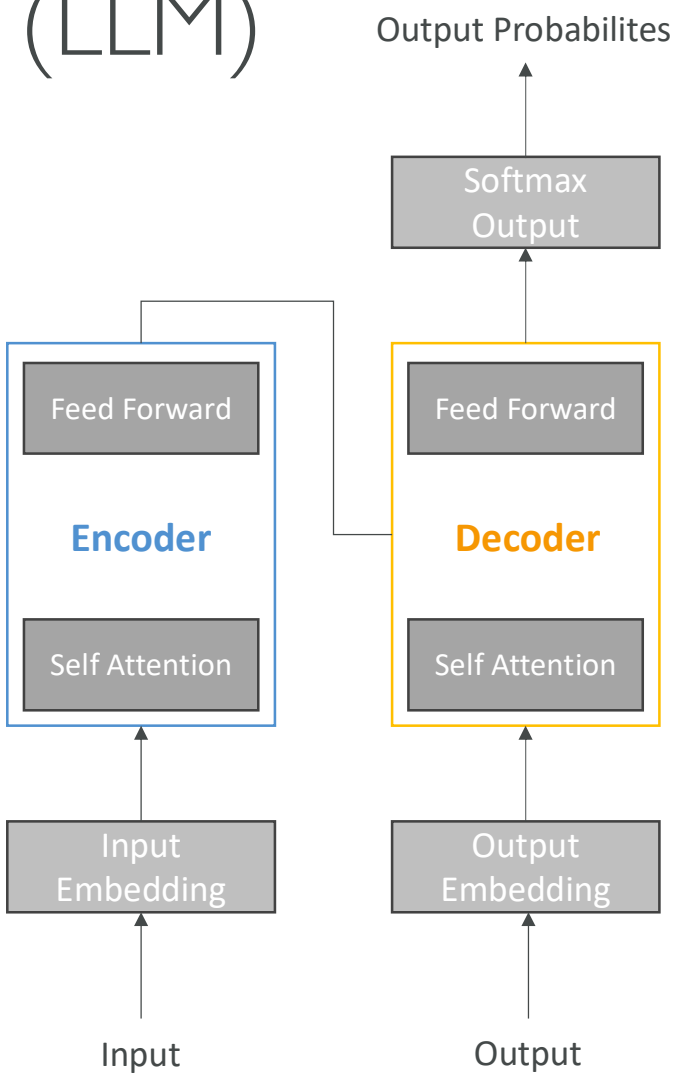
What is Generative AI (Gen-AI)?

- Subset of Deep Learning
- Multi-purpose foundation models backed by neural networks
- They can be fine-tuned if necessary to better fit our use-cases



What is the Transformer Model? (LLM)

- Able to process a sentence as a whole instead of word by word
- Faster and more efficient text processing (less training time)
- It gives relative importance to specific words in a sentence (more coherent sentences)
- Transformer-based LLMs
  - Powerful models that can understand and generate human-like text
  - Trained on vast amounts of text data from the internet, books, and other sources, and learn patterns and relationships between words and phrases
  - Example: Google BERT, OpenAI ChatGPT
  - (ChatGPT = Chat Generative Pretrained Transformer)



Diffusion Models (ex: Stable Diffusion)

Training: Forward diffusion process

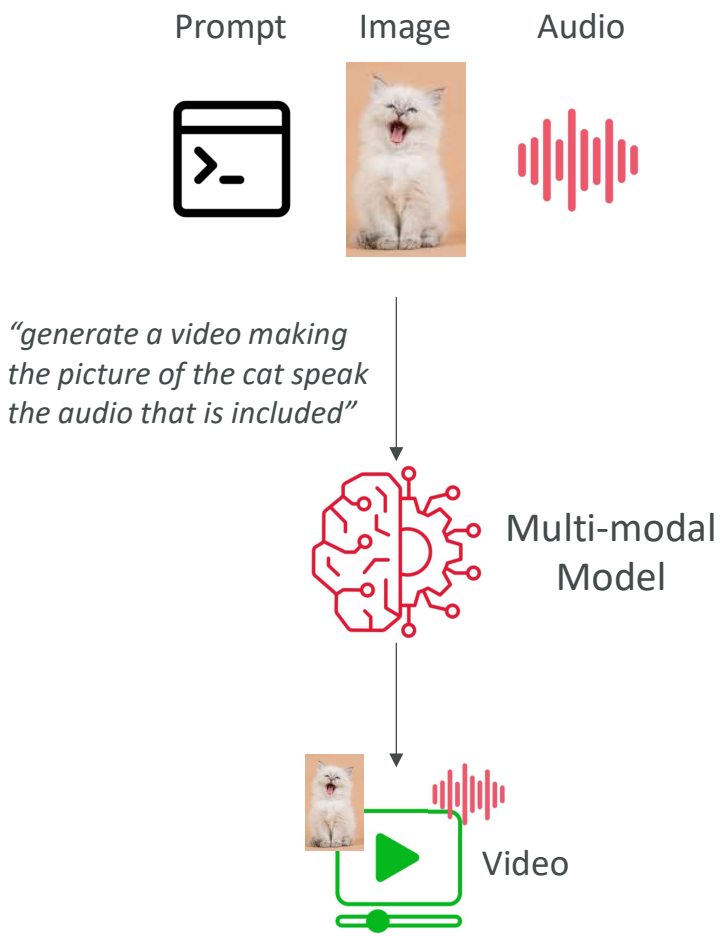
Picture

Noise

Generating: Reverse diffusion process
"a cat with a computer"

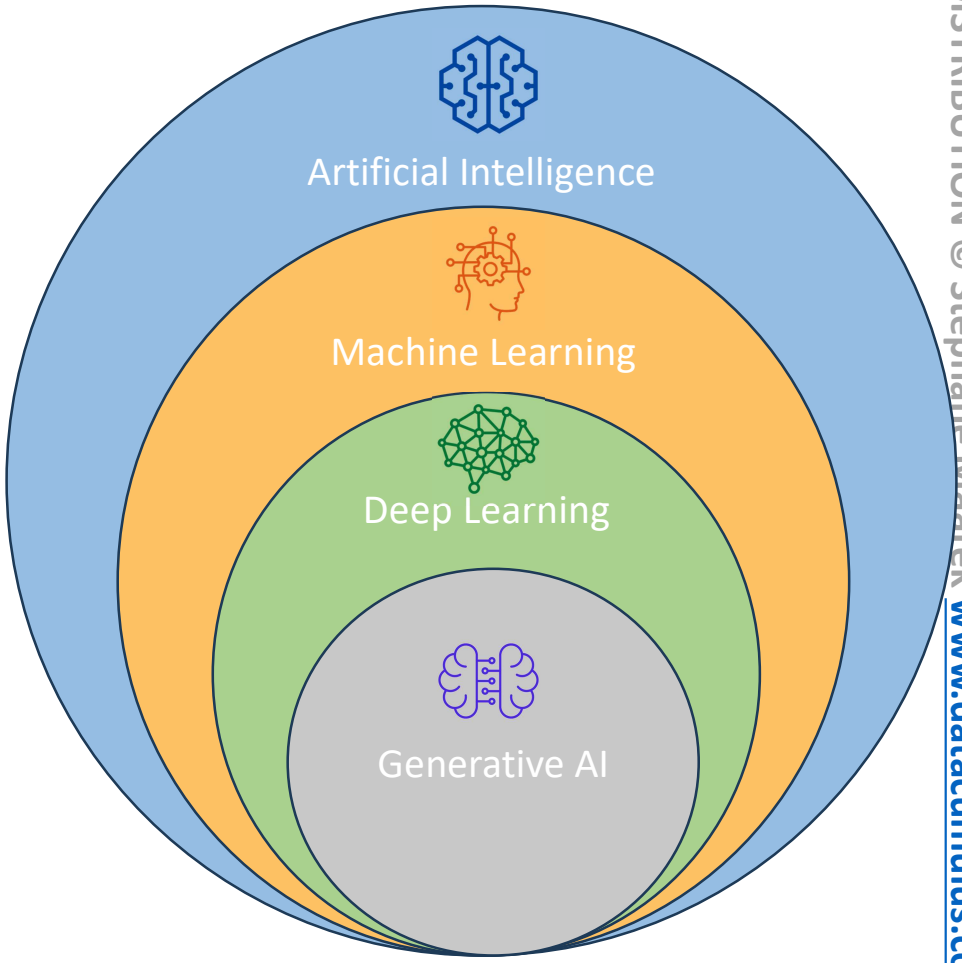
Multi-modal Models (ex: GPT-4o)

- Does NOT rely on a single type of input (text, or images, or audio only)
- Does NOT create a single type of output
- Example: a multi-modal can take a mix of audio, image and text and output a mix of video, text for example



Humans are a mix of AI

- Sometimes we know "if this happens, then do that" (AI)
- Sometimes we've seen a lot of similar things before, and we classify them (Machine Learning)
- Sometimes we haven't seen something before, but we have "learned" a lot of similar concepts, so we can make a decision (Deep Learning)
- Sometimes, we get creative, and based on what we've learned, we can generate content: Gen AI



ML Terms You May Encounter in the Exam

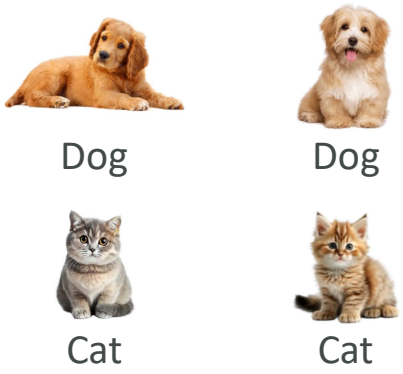
- **GPT (Generative Pre-trained Transformer)** – generate human text or computer code based on input prompts
- **BERT (Bidirectional Encoder Representations from Transformers)** – similar intent to GPT, but reads the text in two directions
- **RNN (Recurrent Neural Network)** – meant for sequential data such as time-series or text, useful in speech recognition, time-series prediction
- **ResNet (Residual Network)** – Deep Convolutional Neural Network (CNN) used for image recognition tasks, object detection, facial recognition
- **SVM (Support Vector Machine)** – ML algorithm for classification and regression
- **WaveNet** – model to generate raw audio waveform, used in Speech Synthesis
- **GAN (Generative Adversarial Network)** – models used to generate synthetic data such as images, videos or sounds that resemble the training data. Helpful for data augmentation
- **XGBoost (Extreme Gradient Boosting)** – an implementation of gradient boosting

Training Data

- **To train our model we must have good data**
- Garbage in => Garbage out
- Most critical stage to build a good model
- Several options to model our data, which will impact the types of algorithms we can use to train our models
- **Labeled** vs. **Unlabeled** Data
- **Structured** vs. **Unstructured** Data

Labeled vs. Unlabeled Data

- Labeled Data
  - Data includes both input features and corresponding output labels
  - Example: dataset with images of animals where each image is labeled with the corresponding animal type (e.g., cat, dog)
  - Use case: **Supervised Learning**, where the model is trained to map inputs to known outputs



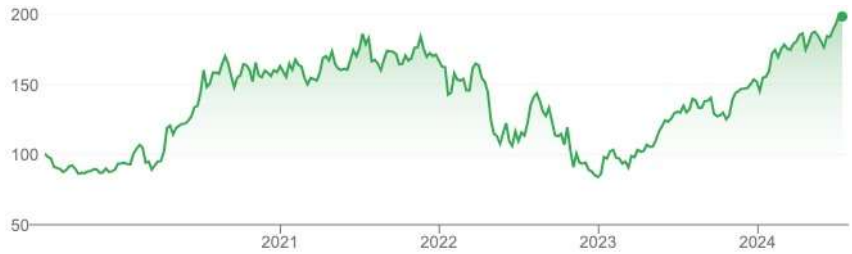
- Unlabeled Data
  - Data includes only input features without any output labels
  - Example: a collection of images without any associated labels
  - Use case: **Unsupervised Learning**, where the model tries to find patterns or structures in the data

Structured Data

- Data is organized in a structured format, often in rows and columns (like Excel)
- Tabular Data
  - Data is arranged in a table with rows representing records and columns representing features
  - Example: customers database with fields such as name, age, and total purchase amount

Customer_ID	Name	Age	Purchase_Amount
1	Alice	30	$200
2	Bob	45	$300

- Time Series Data
  - Data points collected or recorded at successive points in time
  - Example: Stock prices recorded daily over a year



Date	Stock Price
01-07-2024	$197.20
02-07-2024	$200

Unstructured Data

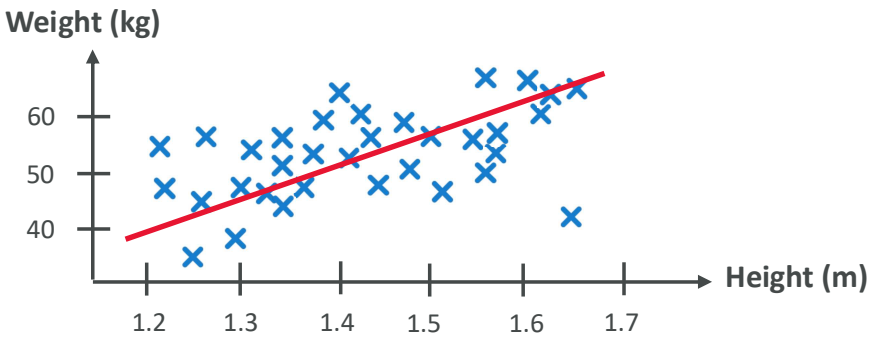
- Data that doesn't follow a specific structure and is often text-heavy or multimedia content
- Text Data
  - Unstructured text such as articles, social media posts, or customer reviews
  - Example: a collection of product reviews from an e-commerce site
- Image Data
  - Data in the form of images, which can vary widely in format and content
  - Example: images used for object recognition tasks

Review: Attended a yoga class at the new studio. The instructor was excellent, and the facility was well-maintained. Loved the variety of classes offered. Only downside was the parking situation.

ML Algorithms – Supervised Learning

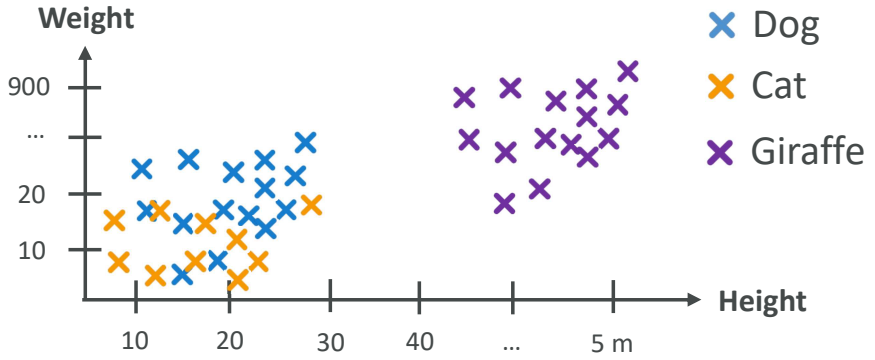
- Learn a mapping function that can predict the output for new unseen input data
- Needs labeled data: very powerful, but difficult to perform on millions of datapoints

Regression



What's the weight of a person which is 1.6m tall ?
=> Based on linear regression: 60kg

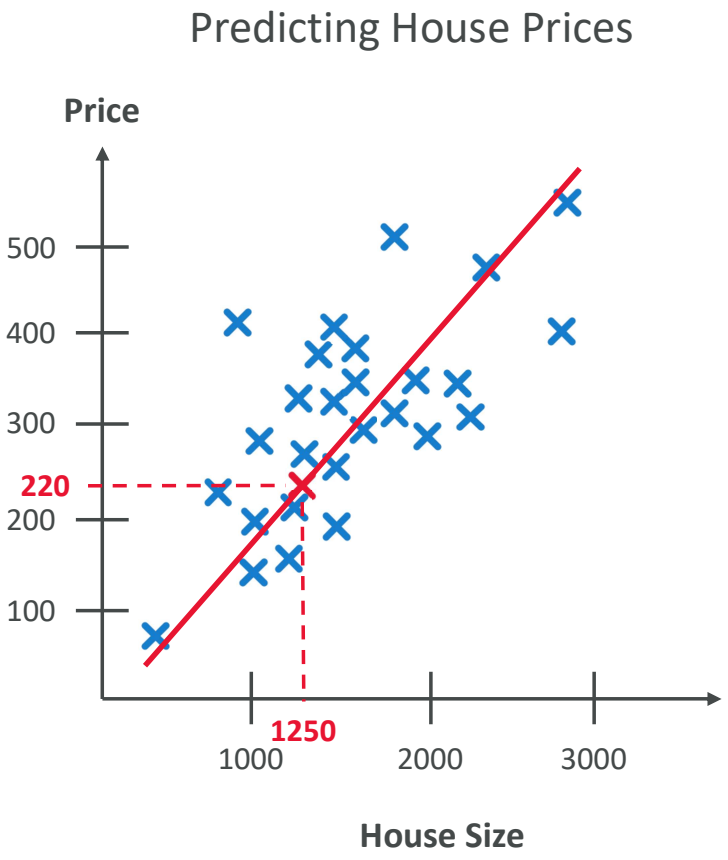
Classification



Which animal is this?
Height: 4.5m
Weight: 800kg

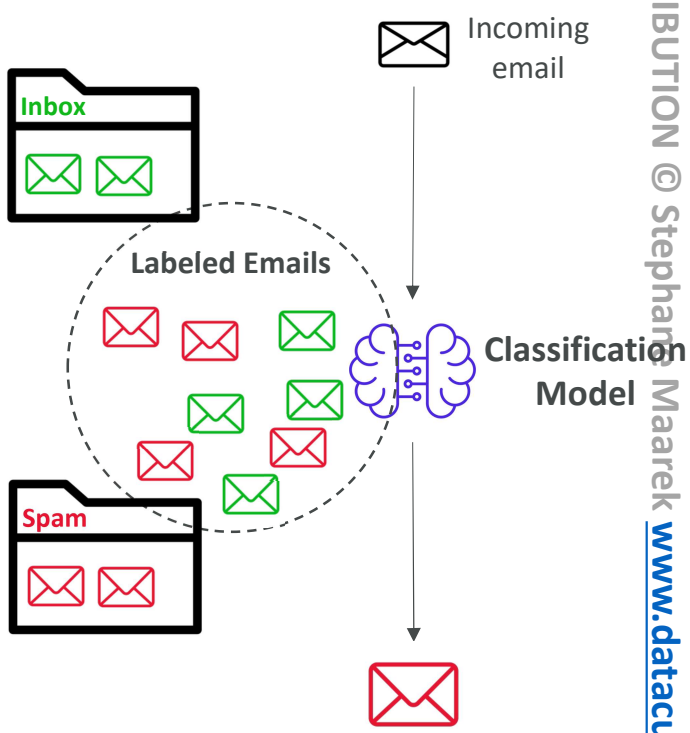
Supervised Learning – Regression

- Used to predict a numeric value based on input data
- The output variable is **continuous**, meaning it can take any value within a range
- Use cases: used when the goal is to predict a quantity or a real value
- Examples:
  - **Predicting House Prices** – based on features like size, location, and number of bedrooms
  - **Stock Price Prediction** – predicting the future price of a stock based on historical data and other features
  - **Weather Forecasting** – predicting temperatures based on historical weather data



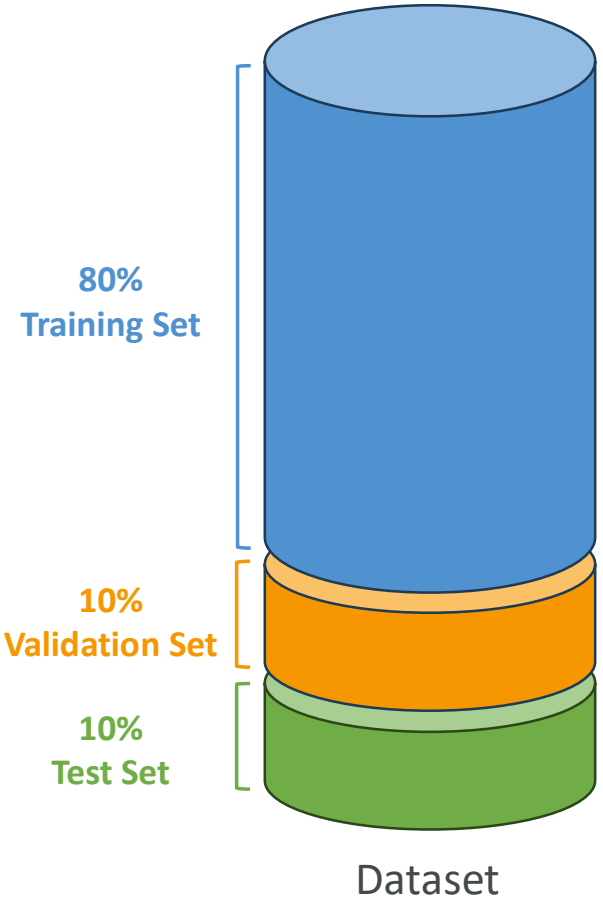
Supervised Learning – Classification

- Used to predict the categorical label of input data
- The output variable is **discrete**, which means it falls into a specific category or class
- Use cases: scenarios where decisions or predictions need to be made between distinct categories (fraud, image classification, customer retention, diagnostics)
- Examples:
  - **Binary Classification** – classify emails as "spam" or "not spam"
  - **Multiclass Classification** – classify animals in a zoo as "mammal," "bird," "reptile"
  - **Multi-label Classification** – assign multiple labels to a movie, like "action" and "comedy
- Key algorithm: K-nearest neighbors (k-NN) model



Training vs. Validation vs. Test Set

- Training Set
  - Used to train the model
  - Percentage: typically, **60-80%** of the dataset
  - Example: 800 labeled images from a dataset of 1000 images
- Validation Set
  - Used to tune model parameters and validate performance
  - Percentage: typically, **10-20%** of the dataset
  - Example: 100 labeled images for hyperparameter tuning (tune the settings of the algorithm to make it more efficient)
- Test Set
  - Used to evaluate the final model performance
  - Percentage: typically, **10-20%** of the dataset
  - Example: 100 labeled images to test the model's accuracy



Feature Engineering

- The process of using domain knowledge to select and transform raw data into meaningful features
- Helps enhancing the performance of machine learning models
- Techniques
  - **Feature Extraction** – extracting useful information from raw data, such as deriving age from date of birth
  - **Feature Selection** – selecting a subset of relevant features, like choosing important predictors in a regression model
  - **Feature Transformation** – transforming data for better model performance, such as normalizing numerical data
- Particularly meaningful for **Supervised Learning**

Before
Feature Engineering

Customer_ID	Name	BirthDate	Purchase_Amount
1	Alice	**15-05-1993**	$200
2	Bob	**22-08-1978**	$300

After
Feature Engineering

Customer_ID	Name	Age	Purchase_Amount
1	Alice	**30**	$200
2	Bob	**45**	$300

Feature Engineering on **Structured** Data

- Structured Data (Tabular Data)
- Example: Predicting house prices based on features like size, location, and number of rooms

- Feature Engineering Tasks
  - **Feature Creation** – deriving new features like "price per square foot"
  - **Feature Selection** – identifying and retaining important features such as location or number of bedrooms
  - **Feature Transformation** – normalizing features to ensure they are on a similar scale, which helps algorithms like gradient descent converge faster

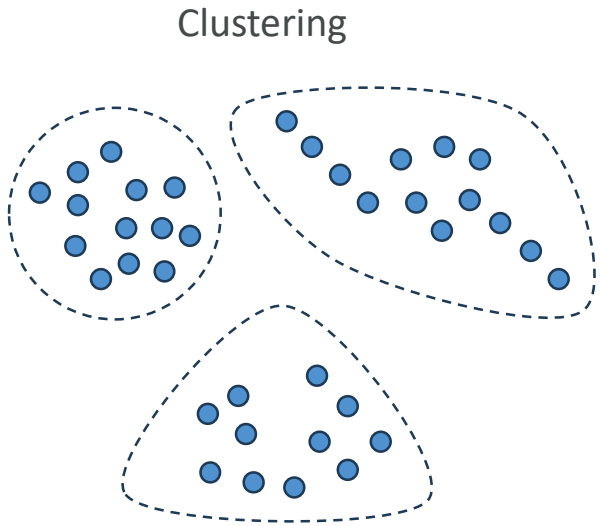
Feature Engineering on **Unstructured** Data

- Unstructured Data (Text, Images)
- Example: sentiment analysis of customer reviews

- Feature Engineering Tasks
  - **Text Data** – converting text into numerical features using techniques like TF-IDF or word embeddings
  - **Image Data** – extracting features such as edges or textures using techniques like convolutional neural networks (CNNs)

ML Algorithms – Unsupervised Learning

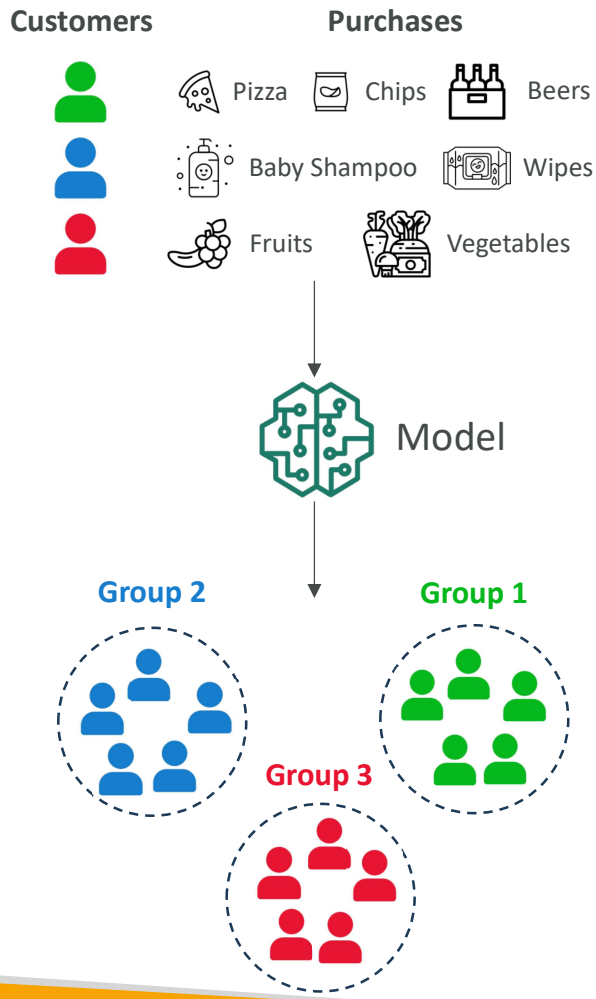
- The goal is to discover inherent patterns, structures, or relationships within the input data
- The machine must uncover and create the groups itself, but humans still put labels on the output groups
- Common techniques include **Clustering**, **Association Rule Learning** , and **Anomaly Detection**
- Clustering use cases: customer segmentation, targeted marketing, recommender systems
- **Feature Engineering** can help improve the quality of the training



Unsupervised Learning

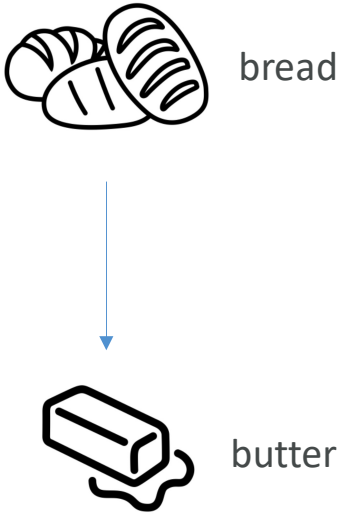
Unsupervised Learning – Clustering Technique

- Used to group similar data points together into clusters based on their features
- Example: Customer Segmentation
  - Scenario: e-commerce company wants to segment its customers to understand different purchasing behaviors
  - Data: A dataset containing customer purchase history (e.g., purchase frequency, average order value)
  - Goal: Identify distinct groups of customers based on their purchasing behavior
  - Technique: K-means Clustering
- Outcome: The company can target each segment with tailored marketing strategies



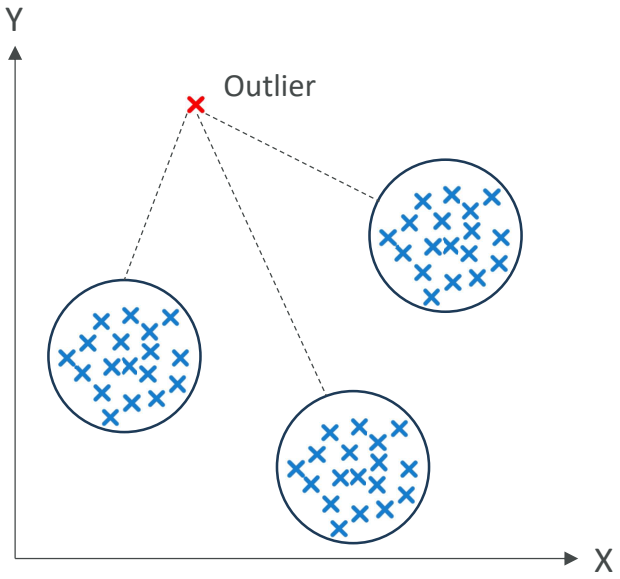
Unsupervised Learning – Association Rule Learning Technique

- Example: Market Basket Analysis
  - **Scenario:** supermarket wants to understand which products are frequently bought together
  - **Data:** transaction records from customer purchases
  - **Goal:** Identify associations between products to optimize product placement and promotions
  - **Technique: Apriori** algorithm
- Outcome: the supermarket can place associated products together to boost sales



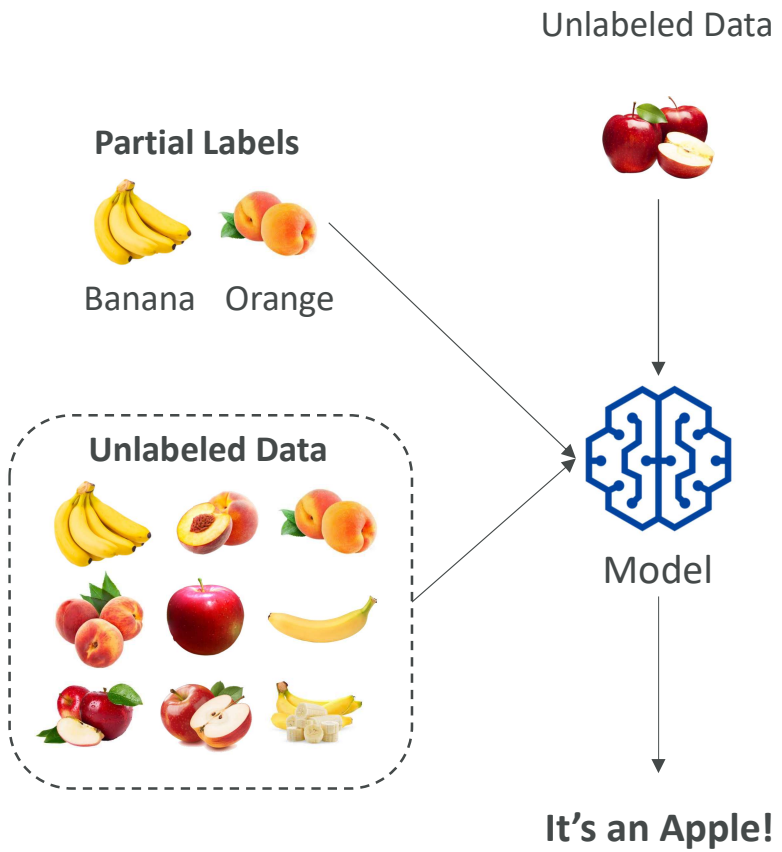
Unsupervised Learning – Anomaly Detection Technique

- Example: Fraud Detection
  - **Scenario:** detect fraudulent credit card transactions
  - **Data:** transaction data, including amount, location, and time
  - **Goal:** identify transactions that deviate significantly from typical behavior
  - **Technique: Isolation Forest**
- Outcome: the system flags potentially fraudulent transactions for further investigation



Semi-supervised Learning

- Use a small amount of labeled data and a large amount of unlabeled data to train systems
- After that, the partially trained algorithm itself labels the unlabeled data
- This is called pseudo-labeling
- The model is then re-trained on the resulting data mix without being explicitly programmed



Self-Supervised Learning

- Have a model generate pseudo-labels for its own data without having humans label any data first
- Then, using the pseudo labels, solve problems traditionally solved by Supervised Learning
- Widely used in NLP (to create the BERT and GPT models for example) and in image recognition tasks

Huge amount of text data

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Clients will often use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, compute, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware and operating systems. One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk (HDD)/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM)…

Self-supervised learning

Learn the English language, grammar, meaning of words, and relationships between words

Self-Supervised Learning: Intuitive example

- Create "pre-text tasks" to have the model solve simple tasks and learn patterns in the dataset.
- Pretext tasks are not "useful" as such, but will teach our model to create a "representation" of our dataset
  - Predict any part of the input from any other part
  - Predict the future from the past
  - Predict the masked from the visible
  - Predict any occluded part from all available parts
- After solving the pre-text tasks, we have a model trained that can solve our end goal: "downstream tasks"

Unlabeled Data

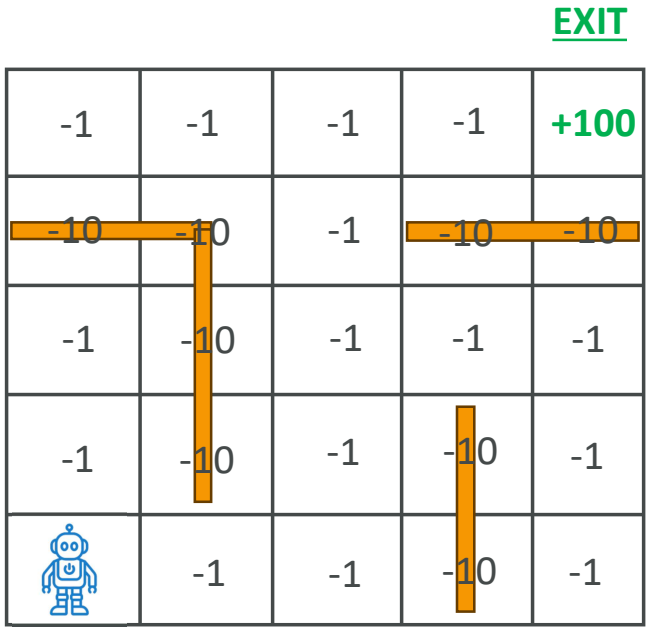
Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis

Pretext Task

X	Y
Amazon Web	Services
that provides on-demand cloud	computing
APIs to individuals, _____, and governments, on a metered, pay-as-you-go	companies

What is Reinforcement Learning (RL)?

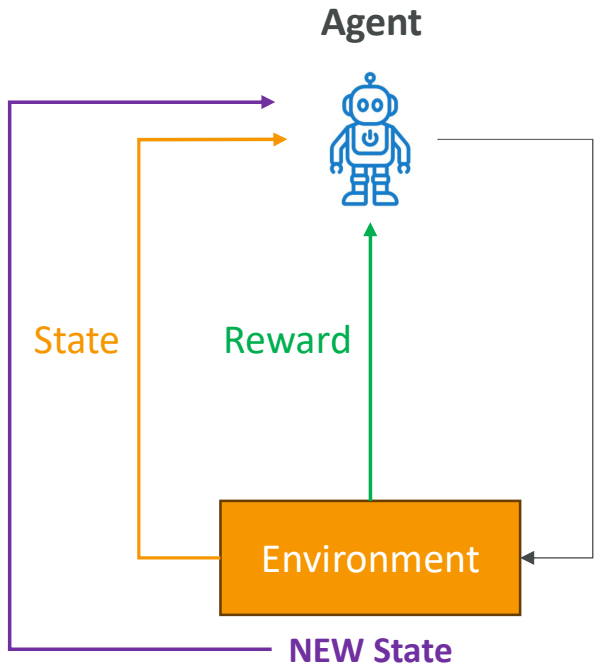
- A type of Machine Learning where an agent learns to make decisions by performing actions in an environment to maximize cumulative rewards
- Key Concepts
  - **Agent** – the learner or decision-maker
  - **Environment** – the external system the agent interacts with
  - **Action** – the choices made by the agent
  - **Reward** – the feedback from the environment based on the agent's actions
  - **State** – the current situation of the environment
  - **Policy** – the strategy the agent uses to determine actions based on the state



Simulate many times
Learn from mistakes
Learn from successes

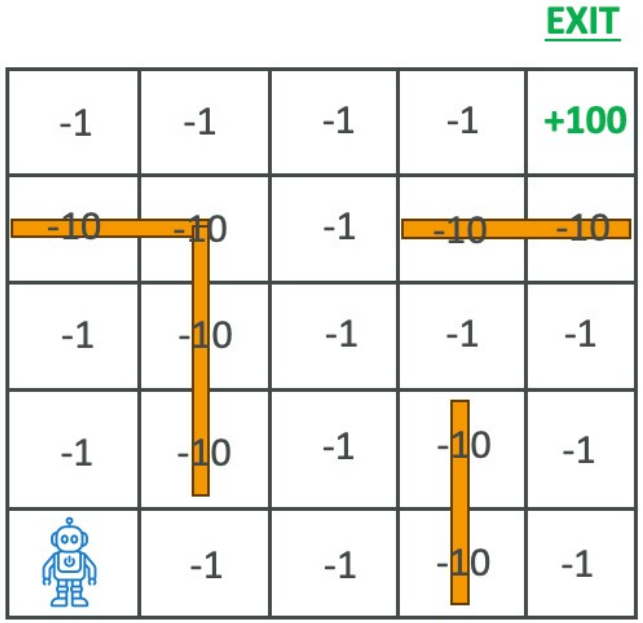
How Does Reinforcement Learning Work?

- Learning Process
  - The Agent observes the current State of the Environment
  - It selects an Action based on its Policy
  - The environment transitions to a new State and provides a Reward
  - The Agent updates its Policy to improve future decisions
- **Goal:** Maximize cumulative reward over time



Example: Reinforcement Learning in Action

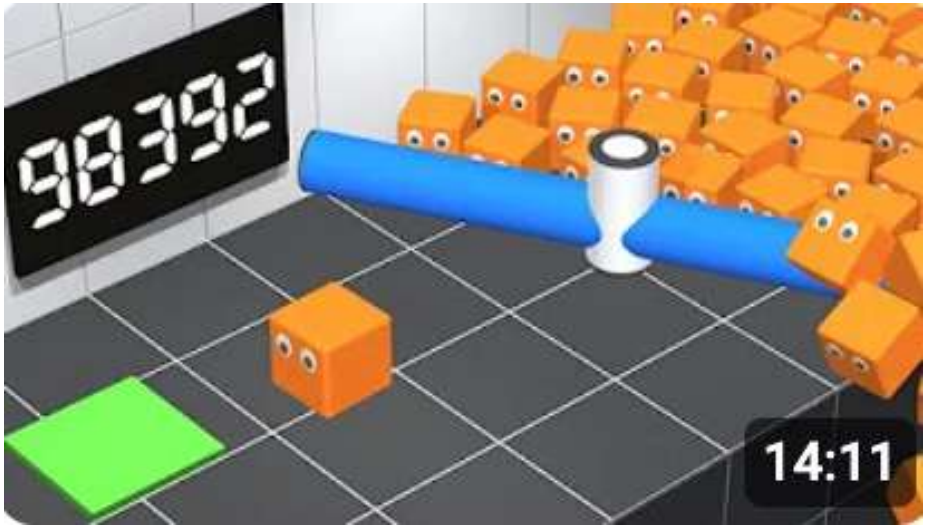
- Scenario: training a robot to navigate a maze
- Steps: robot (Agent) observes its position (State)
  - Chooses a direction to move (Action)
  - Receives a reward (-1 for taking a step, -10 for hitting a wall, +100 for going to the exit)
  - Updates its Policy based on the Reward and new position
- Outcome: the robot learns to navigate the maze efficiently over time



Reinforcement learning - YouTube Channel

- Check out:
- https://www.youtube.com/@aiwarehouse

- For example:
  ”AI Learns to Escape”
  https://youtu.be/2tamH76Tjvw



Applications of Reinforcement Learning

- **Gaming** – teaching AI to play complex games (e.g., Chess, Go)
- **Robotics** – navigating and manipulating objects in dynamic environments
- **Finance** – portfolio management and trading strategies
- **Healthcare** – optimizing treatment plans
- **Autonomous Vehicles** – path planning and decision-making

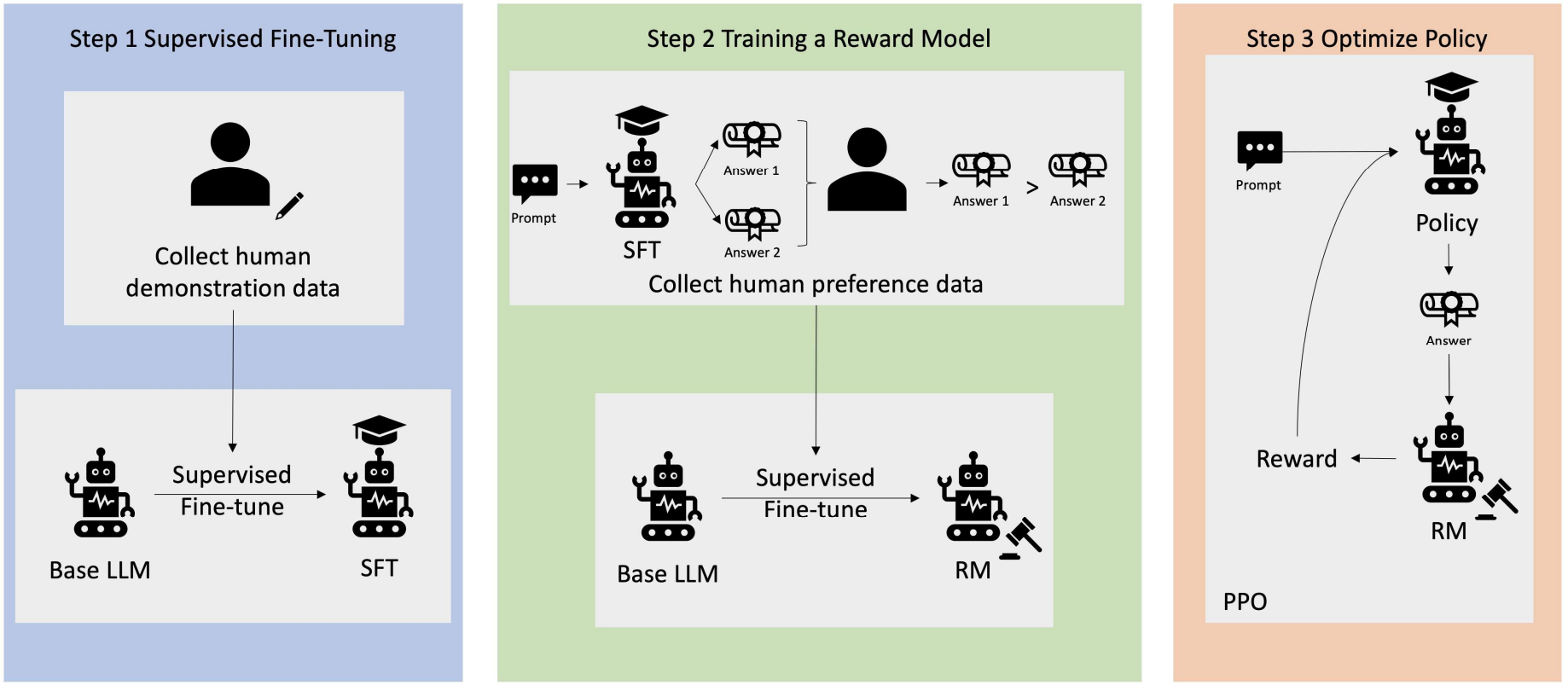
What is RLHF?

- RLHF = Reinforcement Learning from Human Feedback
- Use human feedback to help ML models to self-learn more efficiently
- In Reinforcement Learning there's a reward function
- RLHF incorporates human feedback in the reward function, to be more aligned with human goals, wants and needs
  - First, the model's responses are compared to human's responses
  - Then, a human assess the quality of the model's responses
- RLHF is used throughout GenAI applications including LLM Models
- RLHF significantly enhances the model performance
- Example: grading text translations from "technically correct" to "human"

How does RLHF work?
Example: internal company knowledge chatbot

- Data collection
  - Set of human-generated prompts and responses are created
  - "Where is the location of the HR department in Boston?"
- Supervised fine-tuning of a language model
  - Fine-tune an existing model with internal knowledge
  - Then the model creates responses for the human-generated prompts
  - Responses are mathematically compared to human-generated answers
- Build a separate *reward model*
  - Humans can indicate which response they prefer from the same prompt
  - The reward model can now estimate how a human would prefer a prompt response
- Optimize the language model with the reward-based model
  - Use the *reward model* as a reward function for RL
  - This part can be fully automated

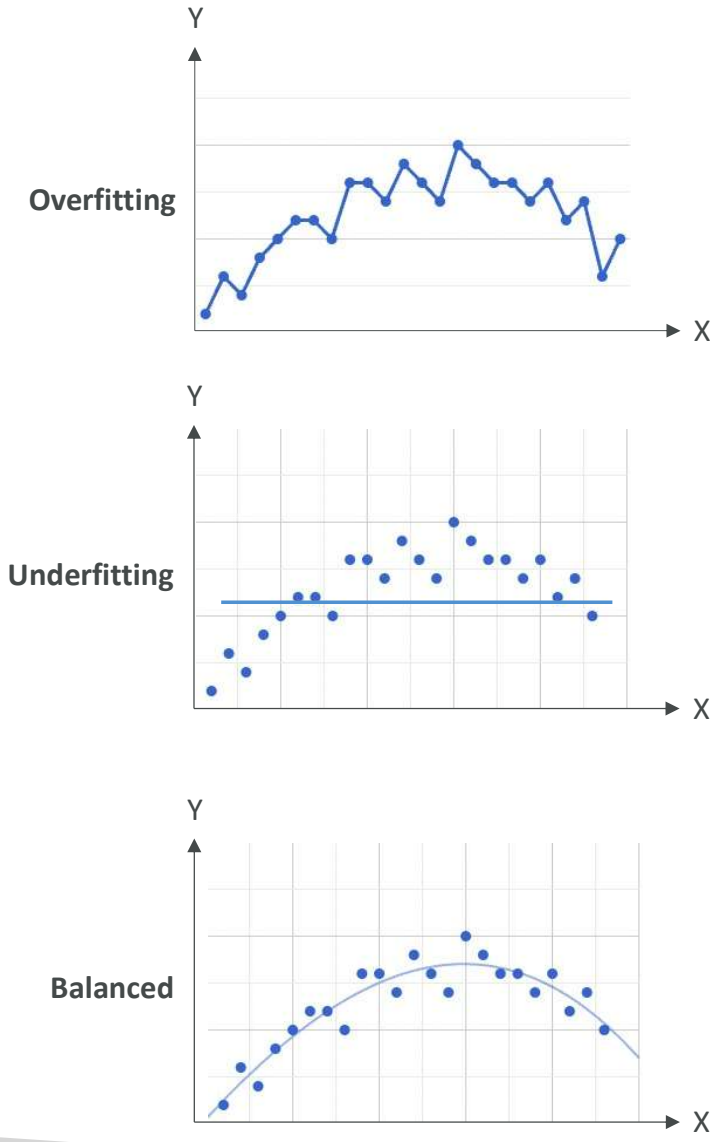
RLHF Process



https://aws.amazon.com/what-is/reinforcement-learning-from-human-feedback/

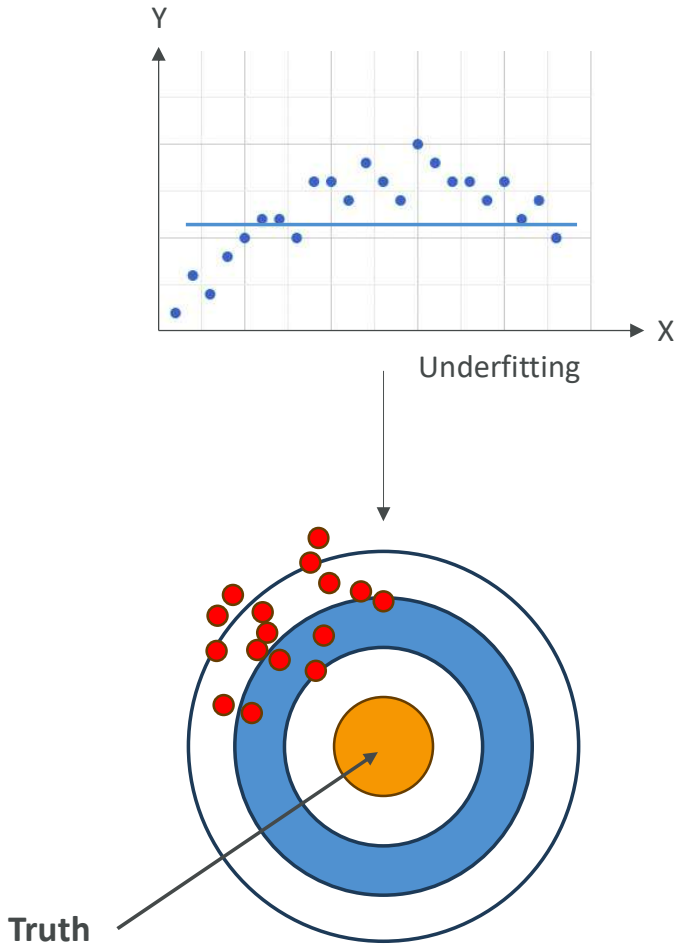
Model Fit

- In case your model has poor performance, you need to look at its fit
- Overfitting
  - Performs well on the training data
  - Doesn't perform well on evaluation data
- Underfitting
  - Model performs poorly on training data
  - Could be a problem of having a model too simple or poor data features
- Balanced
  - Neither overfitting or underfitting



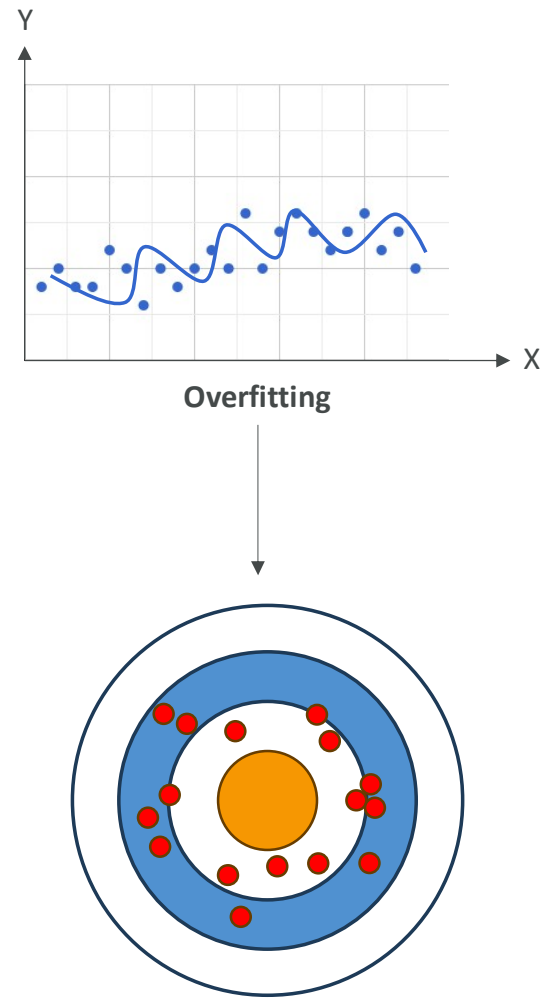
Bias and Variance

- Bias
  - Difference or error between predicted and actual value
  - Occurs due to the wrong choice in the ML process
- High Bias
  - The model doesn't closely match the training data
  - Example: linear regression function on a non-linear dataset
  - Considered as underfitting
- Reducing the Bias
  - Use a more complex model
  - Increase the number of features



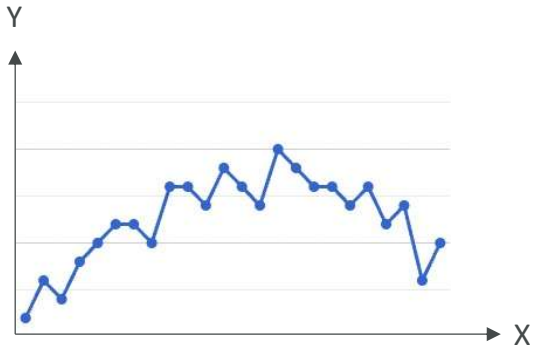
Bias and Variance

- Variance
  - How much the performance of a model changes if trained on a different dataset which has a similar distribution
- High Variance
  - Model is very sensitive to changes in the training data
  - This is the case when overfitting: performs well on training data, but poorly on unseen test data
- Reducing the Variance
  - Feature selection (less, more important features)
  - Split into training and test data sets multiple times



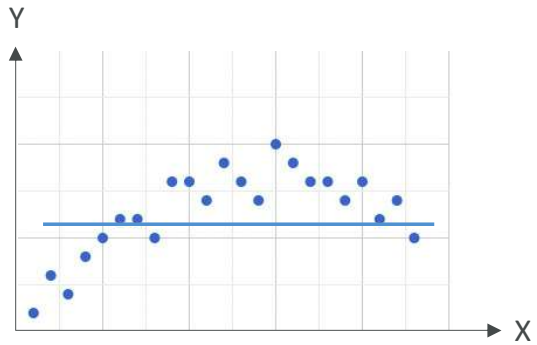
Bias and Variance

High variance



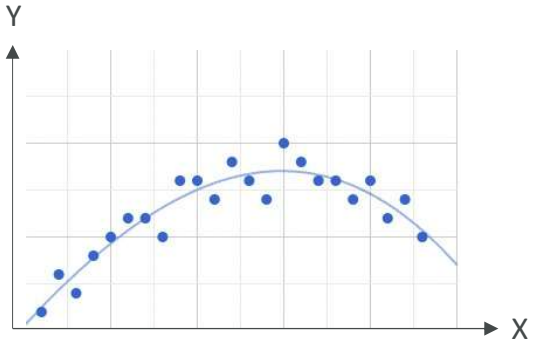
Overfitting

High bias



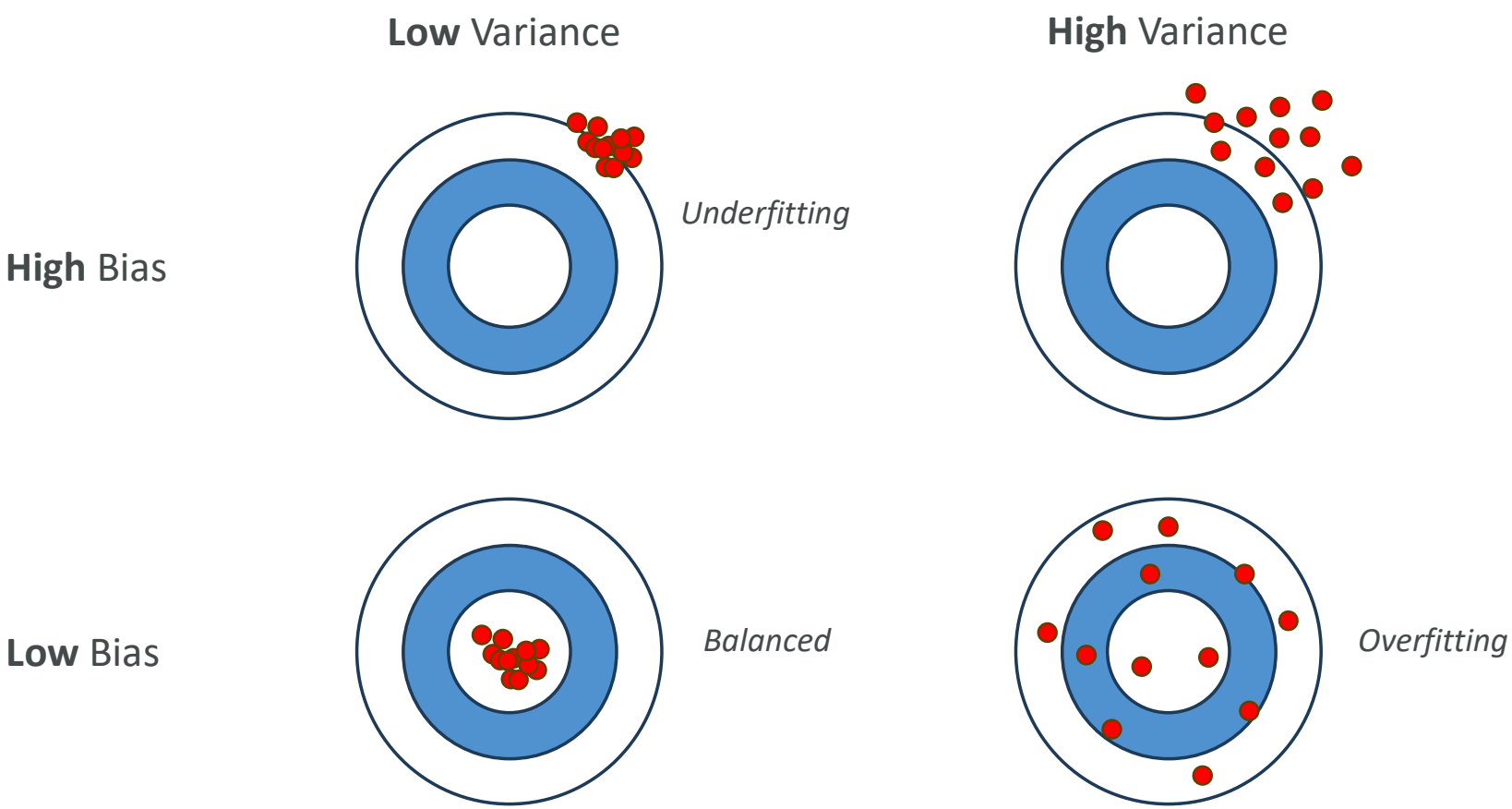
Underfitting

Low bias, low variance

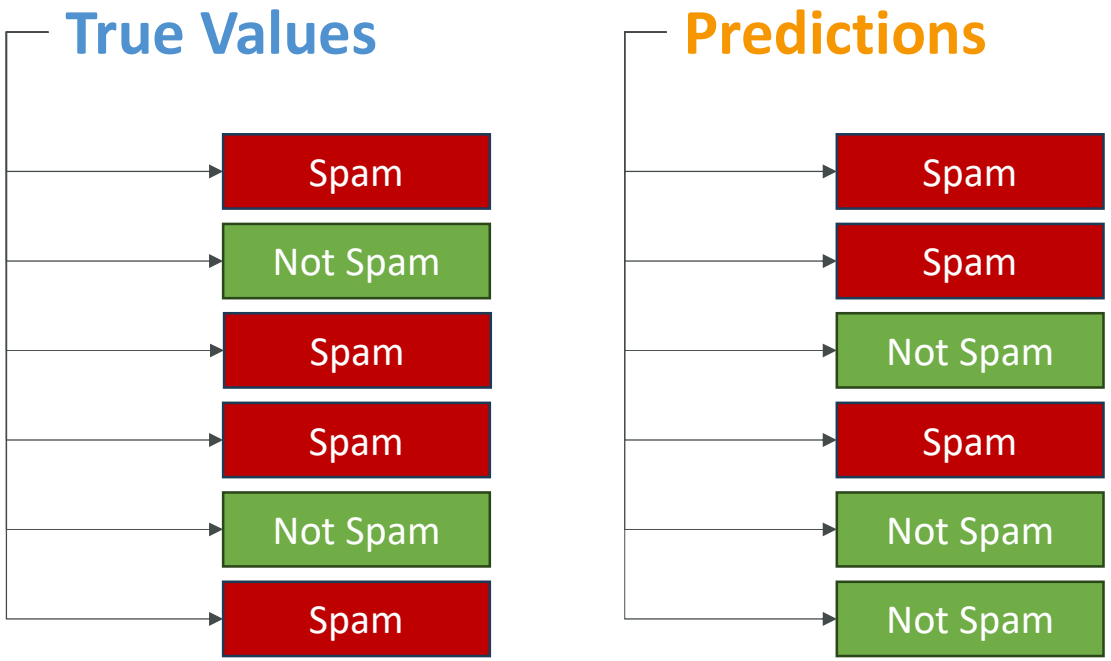


Balanced

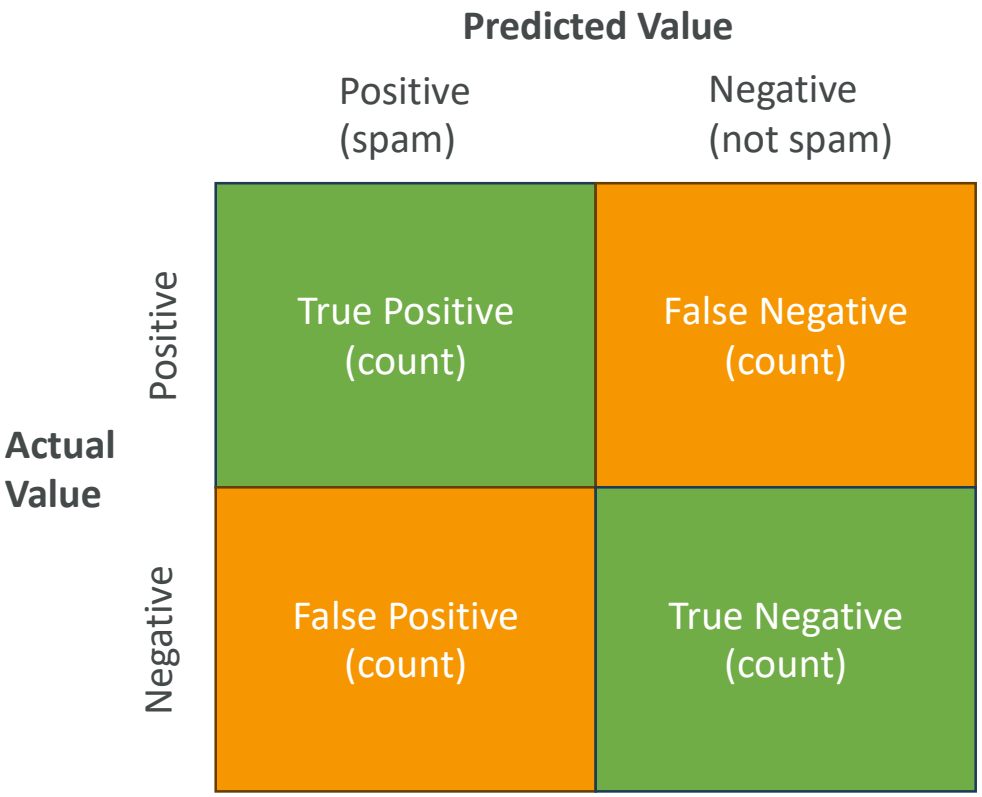
Bias and Variance



Binary Classification Example



Confusion Matrix



$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

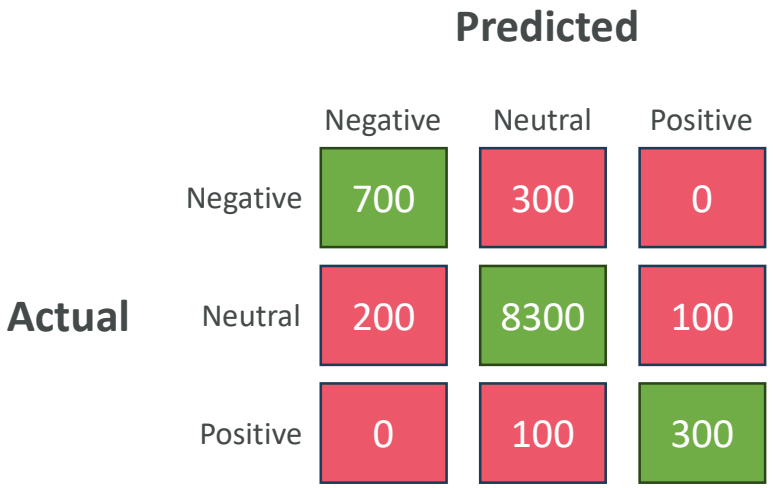
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ **(Rarely used)**

Confusion Matrix - continued

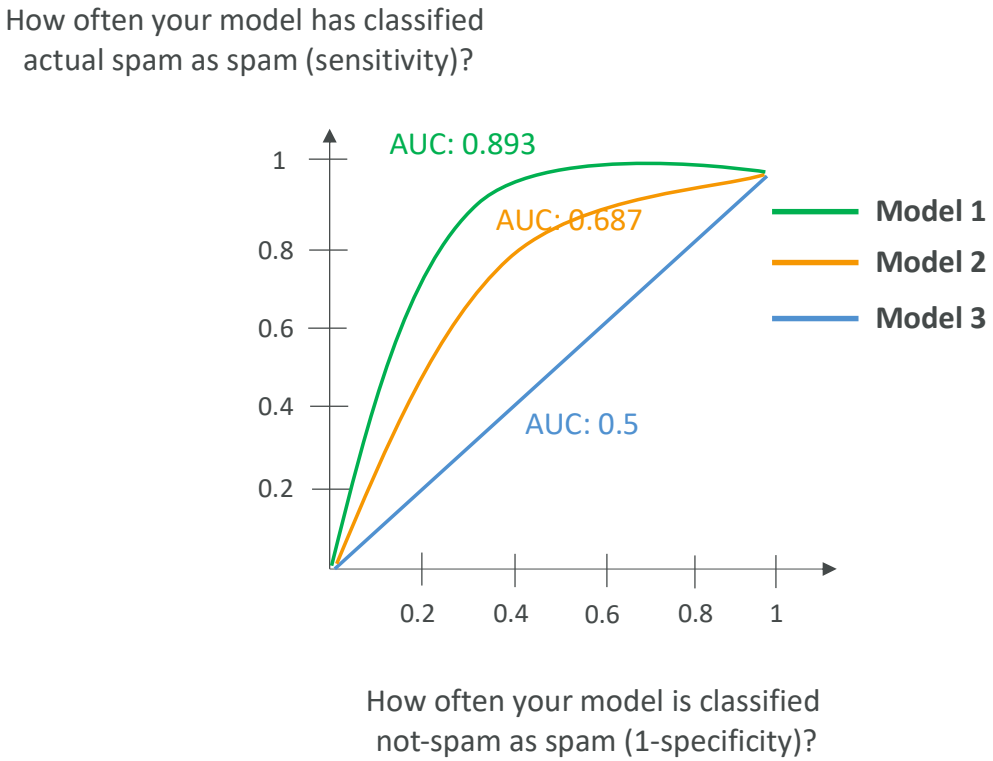
- Confusion Matrixes be multi-dimension too
- Best way to evaluate the performance of a model that does classifications

- Metrics
  - **Precision** – Best when false positives are costly
  - **Recall** – Best when false negatives are costly
  - **F1 Score** – Best when you want a balance between precision and recall, especially in imbalanced datasets
  - **Accuracy** – Best for balanced datasets

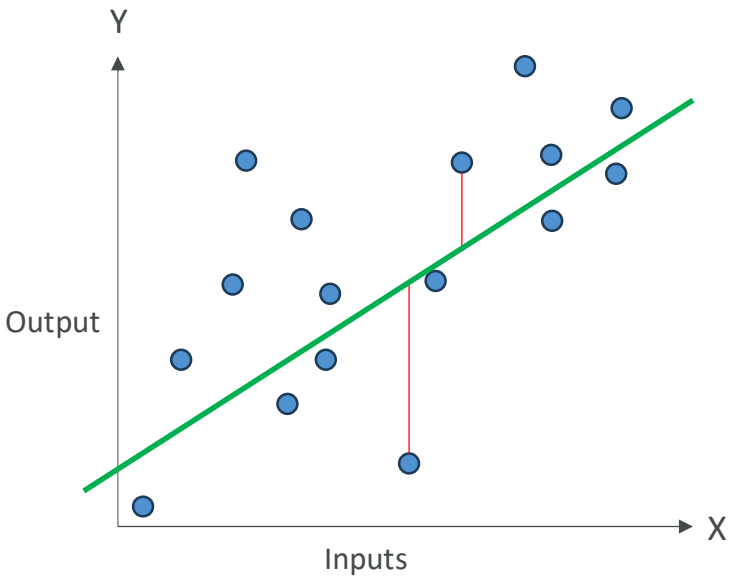


AUC-ROC
Area under the curve-receiver operator curve

- Value from 0 to 1 (perfect model)
- Uses sensitivity (true positive rate) and "1-specificity" (false positive rate)
- AUC-ROC shows what the curve for true positive compared to false positive looks like at various thresholds, with multiple confusion matrixes
- You compare them to one another to find out the threshold you need for your business use case.



Model Evaluation – Regressions Metrics



MAE = Mean Absolute Error
between predicted and actual values

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAPE =
Mean Absolute Percentage Error

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}|$$

RMSE =
Root mean squared error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

R^2 (R Squared): explains variance in your model
R^2 close to 1 means predictions are good

Model Evaluation – Regression Metrics

- MAE, MAPE, RMSE, R^2 (R Squared) are used for evaluating models that predict a **continuous** value (i.e., regressions)
- Example: Imagine you're trying to predict how well students do on a test based on how many hours they study.

- **MAE, MAPE, RMSE** – measure the error: how "accurate" the model is
  - if RMSE is 5, this means that, on average, your model's prediction of a student's score is about **5 points off** from their actual score

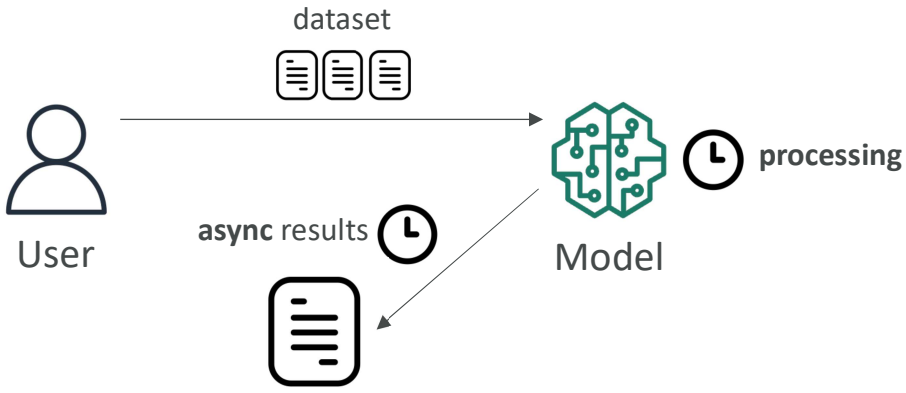
- **R^2 (R Squared)** – measures the variance
  - If R^2 is 0.8, this means that 80% of the changes in test scores can be explained by how much students studied, and the remaining 20% is due to other factors like natural ability or luck

Machine Learning – Inferencing

- Inferencing is when a model is making prediction on new data

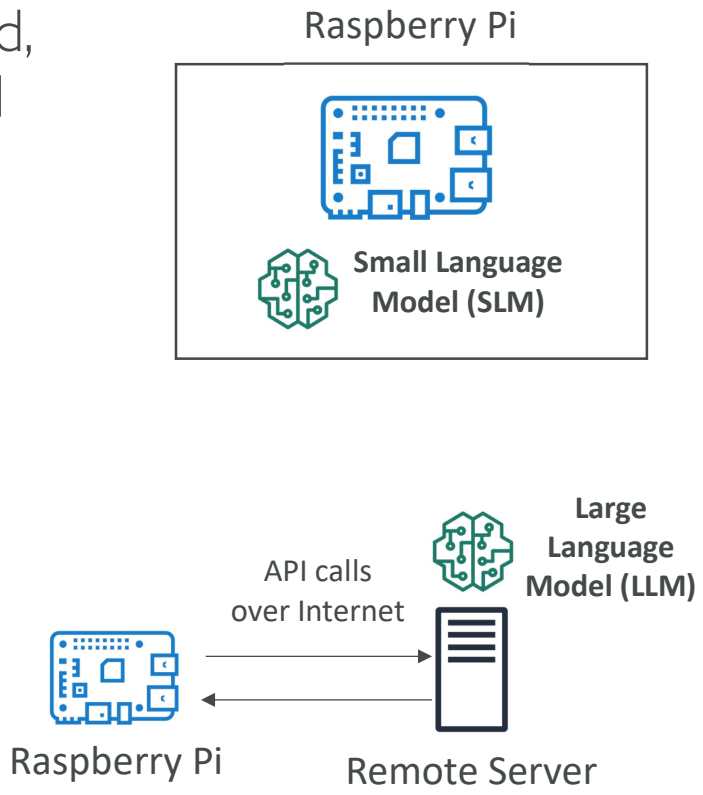
- Real Time
  - Computers have to make decisions quickly as data arrives
  - Speed is preferred over perfect accuracy
  - Example: chatbots

- Batch
  - Large amount of data that is analyzed all at once
  - Often used for data analysis
  - Speed of the results is usually not a concern, and accuracy is

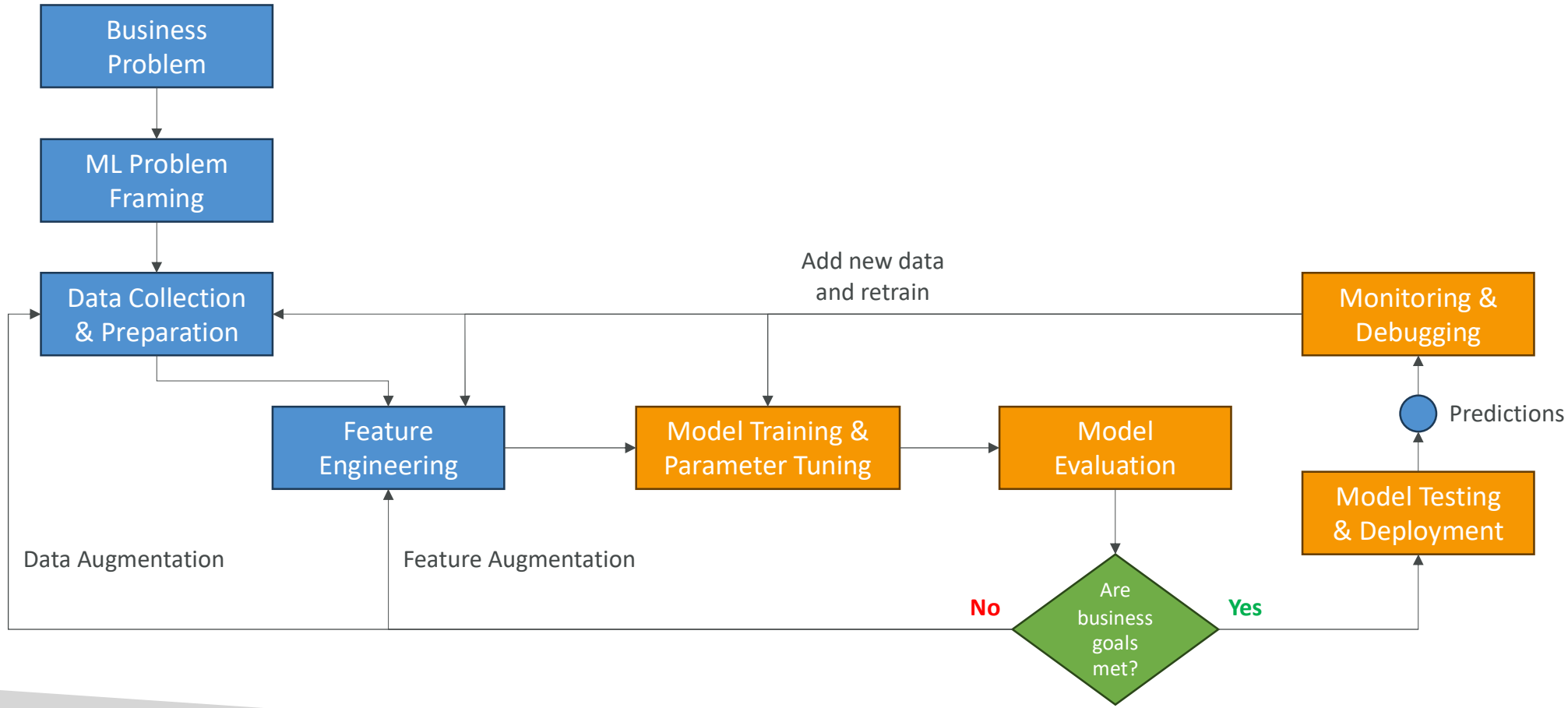


Inferencing at the Edge

- Edge devices are usually devices with less computing power that are close to where the data is generated, in places where internet connections can be limited
- **Small Language Model (SLM)** on the edge device
  - Very low latency
  - Low compute footprint
  - Offline capability, local inference
- **Large Language Model (LLM)** on a remote server
  - More powerful model
  - Higher latency
  - Must be online to be accessed



Phases of Machine Learning Project



Phases of Machine Learning Project

- Define business goals
  - Stakeholders define the value, budget and success criteria
  - Defining KPI (Key Performance Indicators) is critical

- ML problem framing
  - Convert the business problem and into a machine learning problem
  - Determine if ML is appropriate
  - Data scientist, data engineers and ML architects and subject matter experts (SME) collaborate

Phases of Machine Learning Project

- Data processing
  - Convert the data into a usable format
  - Data collection and integration (make it centrally accessible)
  - Data preprocessing and data visualization (understandable format)
  - Feature engineering: create, transform and extract variables from data

- Model development
  - Model training, tuning, and evaluation
  - Iterative process
  - Additional feature engineering and tune model hyperparameters

Exploratory Data Analysis

- Visualize the data with graphs

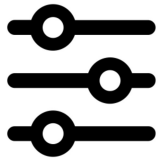
- Correlation Matrix:
  - Look at correlations between variables (how "linked" they are)
  - Helps you decide which features can be important in your model

	Hours Studied	Test Score	Sleep Hours	Distractions
Hours Studied	1	0.85	0.4	-0.6
Test Score	0.85	1	0.3	-0.5
Sleep Hours	0.4	0.3	1	-0.2
Distractions	-0.6	-0.5	-0.2	1

Phases of Machine Learning Project

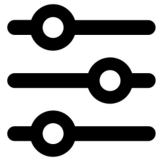
- Retrain
  - Look at data and features to improve the model
  - Adjust the model training hyperparameters
- Deployment
  - If results are good, the model is deployed and ready to make inferences
  - Select a deployment model (real-time, serverless, asynchronous, batch, on-premises…)
- Monitoring
  - Deploy a system to check the desired level of performance
  - Early detection and mitigation
  - Debug issues and understand the model's behavior
- Iterations
  - Model is continuously improved and refined as new data become available
  - Requirements may change
  - Iteration is important to keep the model accurate and relevant over time

Hyperparameter Tuning



- Hyperparameter:
  - Settings that define the model structure and learning algorithm and process
  - Set before training begins
  - Examples: learning rate, batch size, number of epochs, and regularization
- Hyperparameter tuning:
  - Finding the best hyperparameters values to optimize the model performance
  - Improves model accuracy, reduces overfitting, and enhances generalization
- How to do it?
  - Grid search, random search
  - Using services such as SageMaker Automatic Model Tuning (AMT)

Important Hyperparameters



- Learning rate
  - How large or small the steps are when updating the model's weights during training
  - High learning rate can lead to faster convergence but risks overshooting the optimal solution, while a low learning rate may result in more precise but slower convergence.
- Batch size
  - Number of training examples used to update the model weights in one iteration
  - Smaller batches can lead to more stable learning but require more time to compute, while larger batches are faster but may lead to less stable updates.
- Number of Epochs
  - Refers to how many times the model will iterate over the entire training dataset.
  - Too few epochs can lead to underfitting, while too many may cause overfitting
- Regularization
  - Adjusting the balance between simple and complex model
  - Increase regularization to reduce overfitting

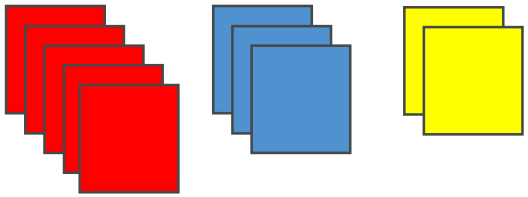
What to do if overfitting?

- Overfitting is when the model gives good predictions for training data but not for the new data
- It occurs due to:
  - Training data size is too small and does not represent all possible input values
  - The model trains too long on a single sample set of data
  - Model complexity is high and learns from the "noise" within the training data
- How can you prevent overfitting?
  - Increase the training data size
  - Early stopping the training of the model
  - Data augmentation (to increase diversity in the dataset)
  - Adjust hyperparameters (but you can't "add" them)

When is Machine Learning NOT appropriate?

- Imagine a well-framed problem like this one:
- A deck contains five red cards, three blue cards, and two yellow cards. What is the probability of drawing a blue card?

Blue probability is **3 out of 10**



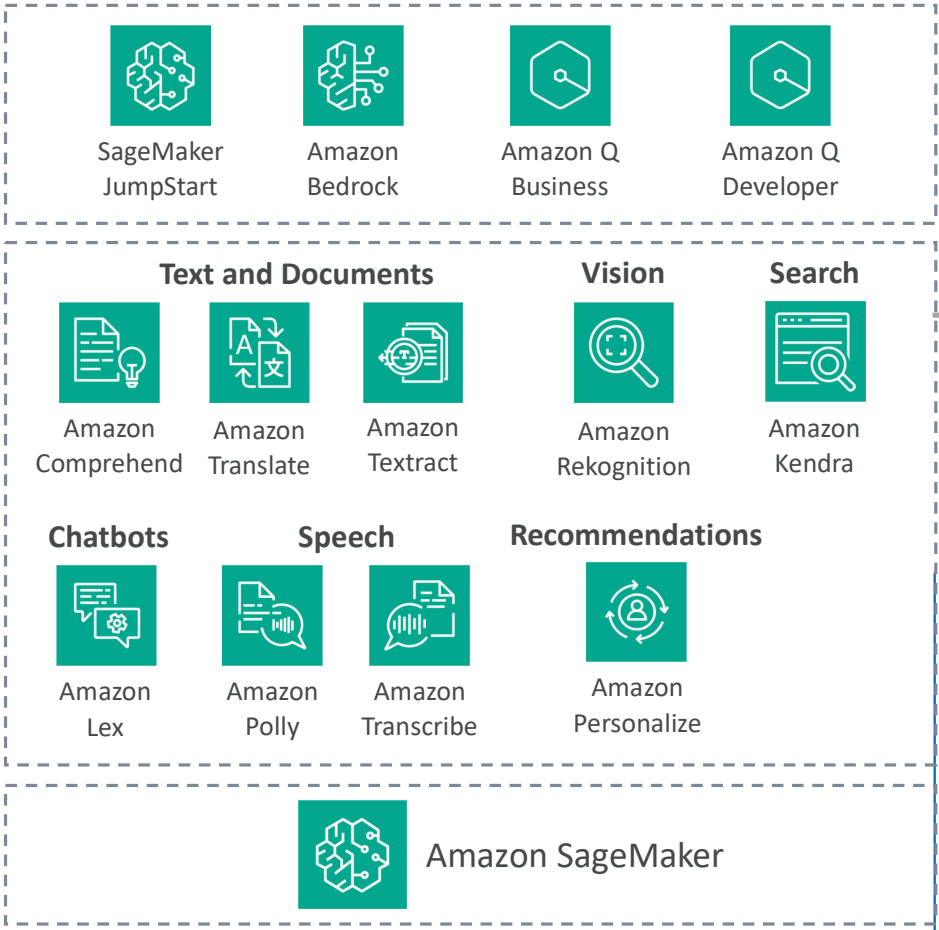
- For deterministic problems (the solution can be computed), it is better to write **computer code** that is adapted to the problem
- If we use Supervised Learning, Unsupervised Learning or Reinforcement Learning, we may have an "approximation" of the result
- Even though nowadays LLMs have reasoning capabilities, they are not perfect and therefore a "worse" solution

```
# Define the number of each card type
red_cards = 5
blue_cards = 3
yellow_cards = 2

# Calculate the total number of cards
total_cards = red_cards + blue_cards + yellow_cards

# Define the color of card we want to find the probability for
target_color_cards = blue_cards

# Calculate probability of drawing the target color card
probability_of_target = target_color_cards / total_cards

# Output the probability
print("The probability of drawing a blue card is:", probability_of_target)
```

AWS Managed AI Services

Why AWS AI Managed Services?

- AWS AI Services are pre-trained ML services for your use case
- Responsiveness and Availability
- **Redundancy and Regional Coverage:** deployed across multiple Availability Zones and AWS regions
- **Performance:** specialized CPU and GPUs for specific use-cases for cost saving
- **Token-based pricing:** pay for what you use
- **Provisioned throughput:** for predictable workloads, cost savings and predictable performance

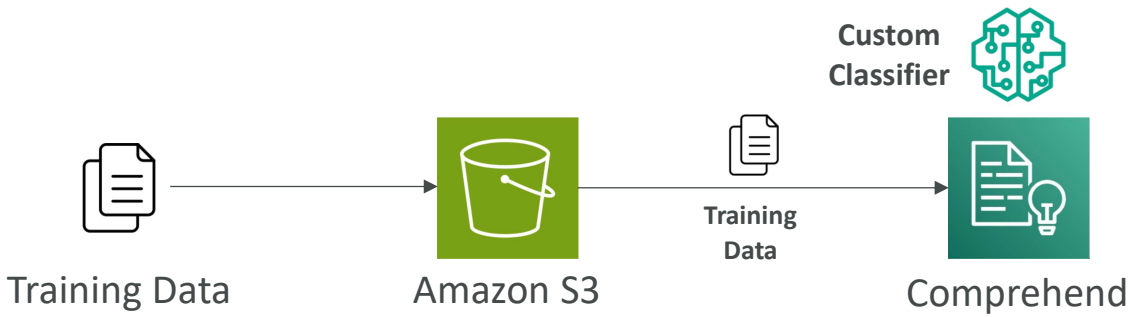


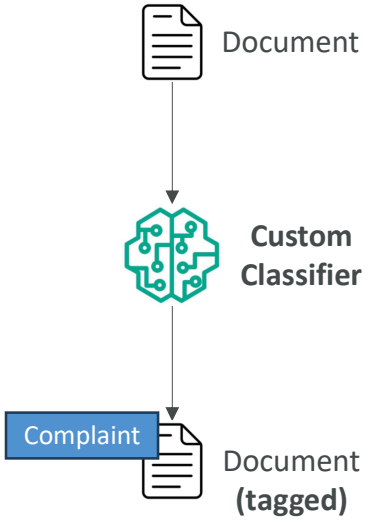
Amazon Comprehend

- For **Natural Language Processing – NLP**
- Fully managed and serverless service
- Uses machine learning to find insights and relationships in text
  - Language of the text
  - Extracts key phrases, places, people, brands, or events
  - Understands how positive or negative the text is
  - Analyzes text using tokenization and parts of speech
  - Automatically organizes a collection of text files by topic
- Sample use cases:
  - analyze customer interactions (emails) to find what leads to a positive or negative experience
  - Create and groups articles by topics that Comprehend will uncover

Comprehend – Custom Classification

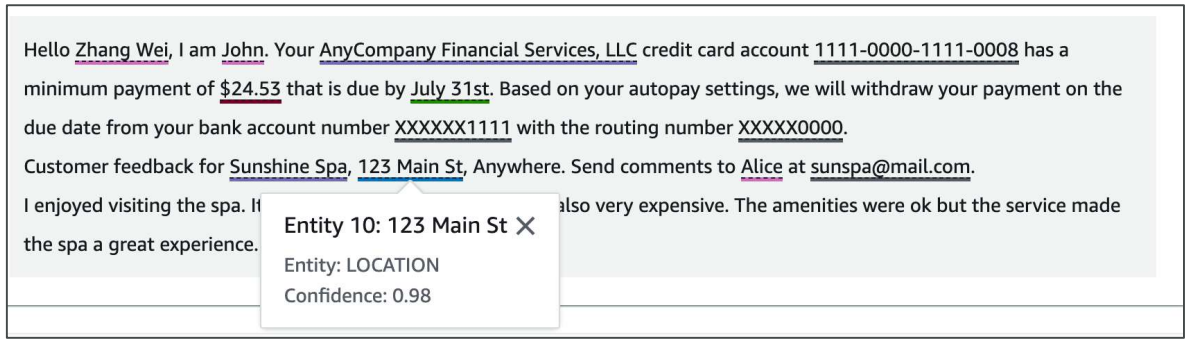
- Organize documents into categories (classes) that you define
- Example: categorize customer emails so that you can provide guidance based on the type of the customer request
- Supports different document types (text, PDF, Word, images...)
- **Real-time Analysis** – single document, synchronous
- **Async Analysis** – multiple documents (batch), Asynchronous





Named Entity Recognition (NER)

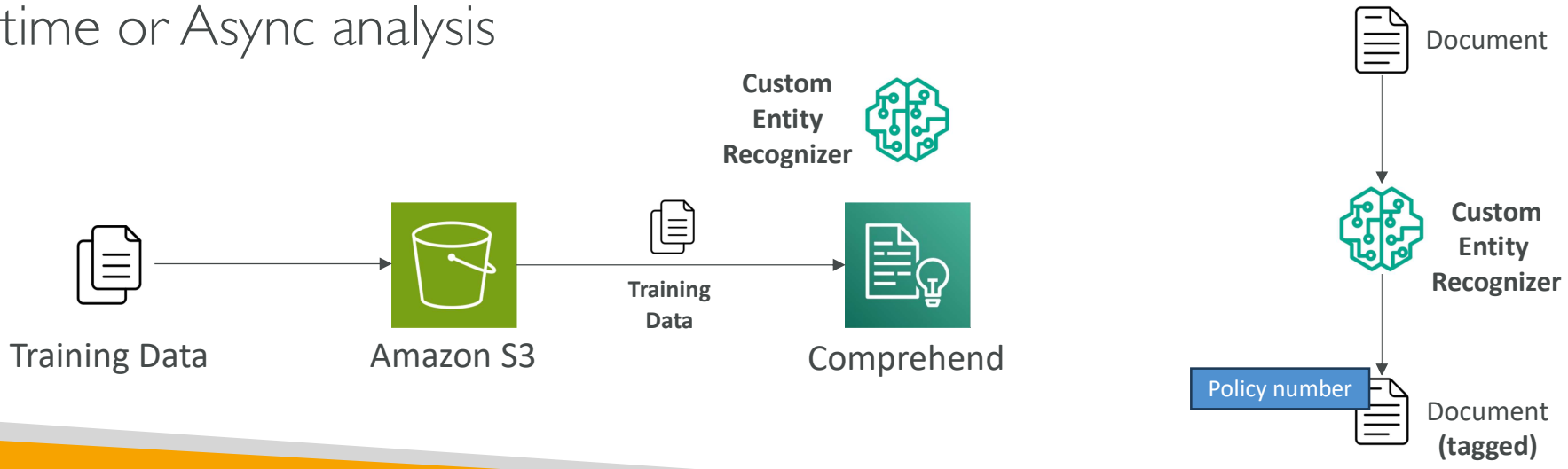
- NER – Extracts predefined, general-purpose entities like people, places, organizations, dates, and other standard categories, **from text**



Entity	Type
Zhang Wei	Person
John	Person
AnyCompany Financial Services, LLC	Organization
1111-0000-1111-0008	Other
$24.53	Quantity
July 31st	Date
XXXXXX1111	Other
XXXXX0000	Other
Sunshine Spa	Organization
123 Main St	Location

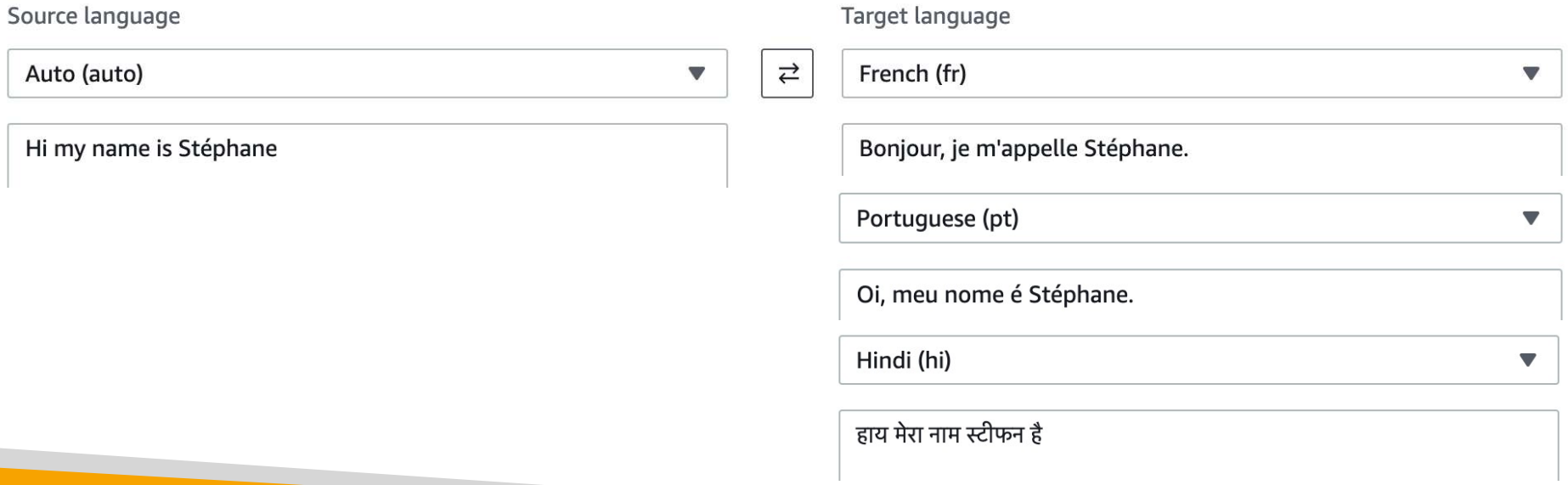
Comprehend – Custom Entity Recognition

- Analyze text for specific terms and noun-based phrases
- Extract terms like policy numbers, or phrases that imply a customer escalation, anything specific to your business
- Train the model with custom data such as a list of the entities and documents that contain them
- Real-time or Async analysis



Amazon Translate

- Natural and accurate **language translation**
- Amazon Translate allows you to **localize content** - such as websites and applications - for **international users**, and to easily translate large volumes of text efficiently.



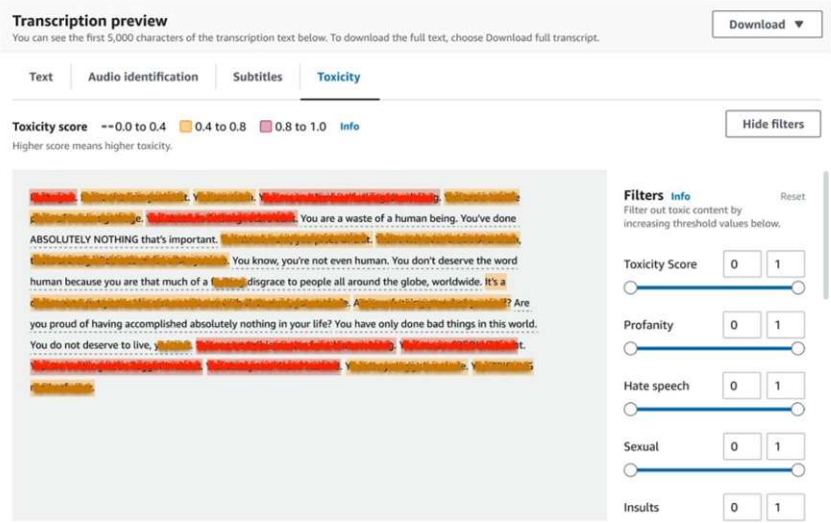
Amazon Transcribe

- Automatically **convert speech to text**
- Uses a **deep learning process** called **automatic speech recognition** (ASR) to convert speech to text quickly and accurately
- **Automatically remove Personally Identifiable Information (PII) using Redaction**
- **Supports Automatic Language Identification for multi-lingual audio**
- Use cases:
  - transcribe customer service calls
  - automate closed captioning and subtitling
  - generate metadata for media assets to create a fully searchable archive

"Hello my name is Stéphane.
I hope you're enjoying the course!

Transcribe – Toxicity Detection

- ML-powered, voice-based toxicity detection capability
- Leverages speech cues: tone and pitch, and text-based cues
- Toxicity categories: sexual harassment, hate speech, threat, abuse, profanity, insult, and graphic….



Toxicity Categories

Profanity: Speech that contains words, phrases, or acronyms that are impolite, vulgar, or offensive.

Hate speech: Speech that criticizes, insults, denounces, or dehumanizes a person or group on the basis of an identity (such as race, ethnicity, gender, religion, sexual orientation, ability, and national origin).

Sexual: Speech that indicates sexual interest, activity, or arousal using direct or indirect references to body parts, physical traits, or sex.

Insults: Speech that includes demeaning, humiliating, mocking, insulting, or belittling language. This type of language is also labeled as bullying.

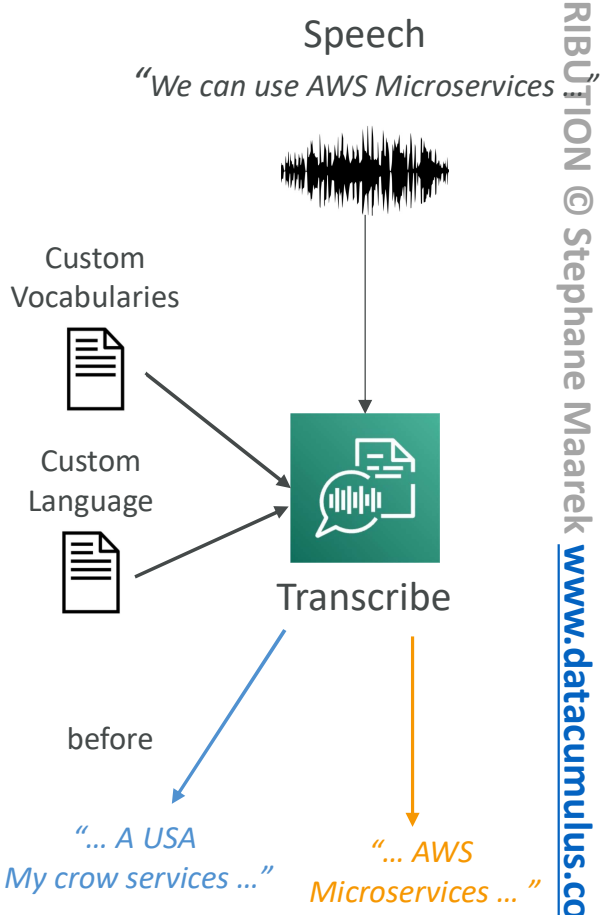
Violence or threat: Speech that includes threats seeking to inflict pain, injury, or hostility toward a person or group.

Graphic: Speech that uses visually descriptive and unpleasantly vivid imagery. This type of language is often intentionally verbose to amplify a recipient's discomfort.

Harassment or abusive: Speech intended to affect the psychological well-being of the recipient, including demeaning and objectifying terms.

Amazon Transcribe – Improving Accuracy

- Allows Transcribe to capture domain-specific or non-standard terms (e.g., technical words, acronyms, jargon…)
- Custom Vocabularies (for words)
  - Add specific words, phrases, domain-specific terms
  - Good for brand names, acronyms…
  - Increase recognition of a new word by providing hints (such as pronunciation..)
- Custom Language Models (for context)
  - Train Transcribe model on your own domain-specific text data
  - Good for transcribing large volumes of domain-specific speech
  - Learn the context associated with a given word
- Note: use both for the highest transcription accuracy



Amazon Polly

- Turn text into lifelike speech using deep learning
- Allowing you to create applications that talk

Hi! My name is Stéphane
and this is a demo of Amazon Polly

Polly – Advanced Features

- Lexicons
  - Define how to read certain specific pieces of text
  - AWS => "Amazon Web Services"
  - W3C => "World Wide Web Consortium"
- SSML - Speech Synthesis Markup Language
  - Markup for your text to indicate how to pronounce it
  - Example: "Hello, <break> how are you?"
- **Voice engine:** generative, long-form, neural, standard…
- Speech mark:
  - Encode where a sentence/word starts or ends in the audio
  - Helpful for lip-syncing or highlight words as they're spoken

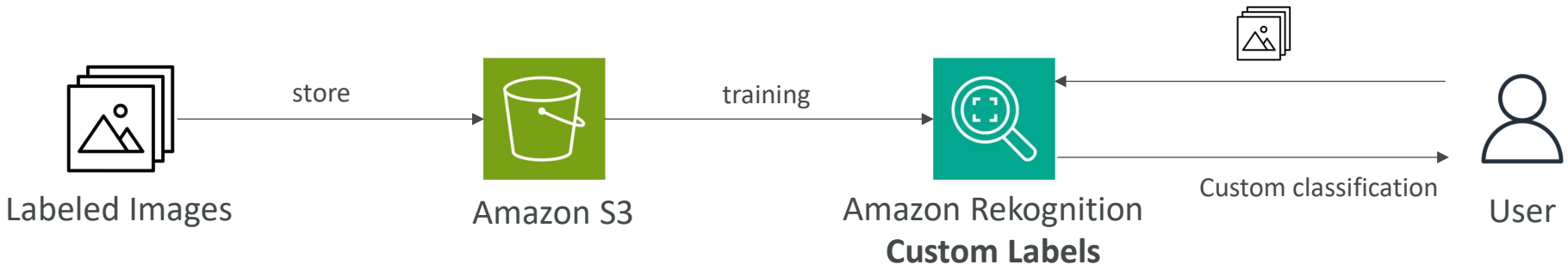
Action	SSML tag
Adding a pause	<break>
Emphasizing words	<emphasis>
Specifying another language for specific words	<lang>
Placing a custom tag in your text	<mark>
Adding a pause between paragraphs	<p>
Using phonetic pronunciation	<phoneme>
Controlling volume, speaking rate, and pitch	<prosody>
Setting a maximum duration for synthesized speech	<prosody amazon:max-duration>
Adding a pause between sentences	<s>
Controlling how special types of words are spoken	<say-as>
Identifying SSML-enhanced text	<speak>
Pronouncing acronyms and abbreviations	<sub>
Improving pronunciation by specifying parts of speech	<w>
Adding the sound of breathing	<amazon:auto-breaths>
Newscaster speaking style	<amazon:domain name="news">
Adding dynamic range compression	<amazon:effect name="drc">
Speaking softly	<amazon:effect

Amazon Rekognition

- Find **objects, people, text, scenes** in **images and videos** using ML
- **Facial analysis** and **facial search** to do user verification, people counting
- Create a database of "familiar faces" or compare against celebrities
- Use cases:
  - Labeling
  - Content Moderation
  - Text Detection
  - Face Detection and Analysis (gender, age range, emotions…)
  - Face Search and Verification
  - Celebrity Recognition
  - Pathing (ex: for sports game analysis)

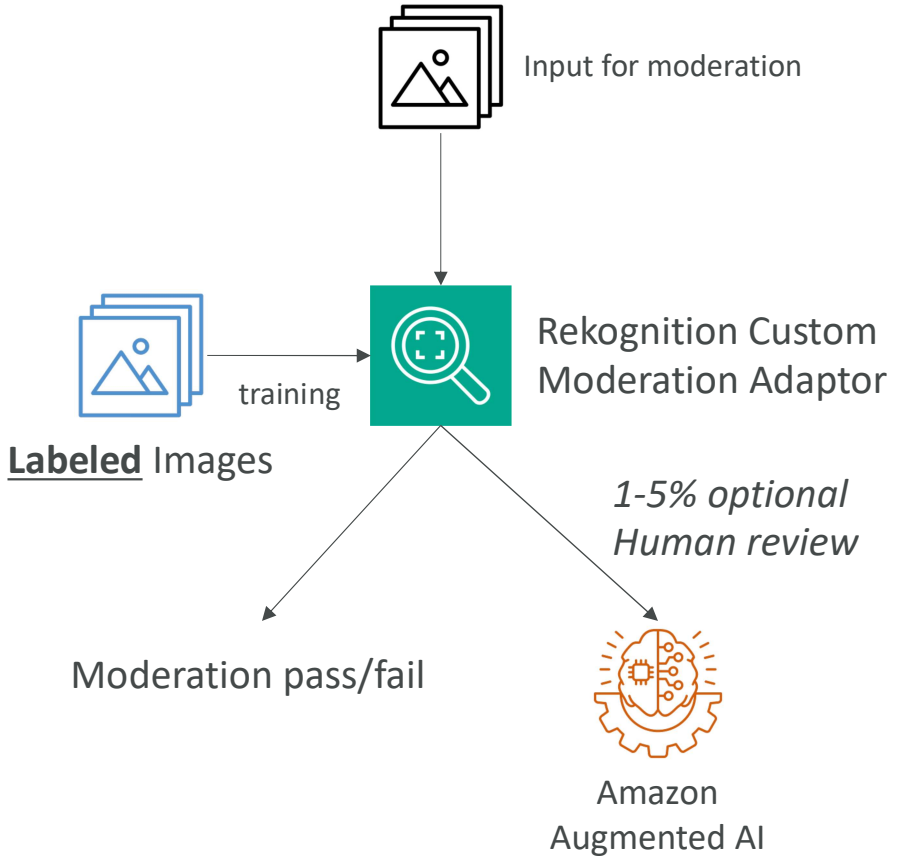
Amazon Rekognition – Custom Labels

- Examples: find your logo in social media posts, identify your products on stores shelves (National Football League – NFL – uses it to find their logo in pictures)
- Label your training images and upload them to Amazon Rekognition
- Only needs a few hundred images or less
- Amazon Rekognition creates a custom model on your images set
- New subsequent images will be categorized the custom way you have defined

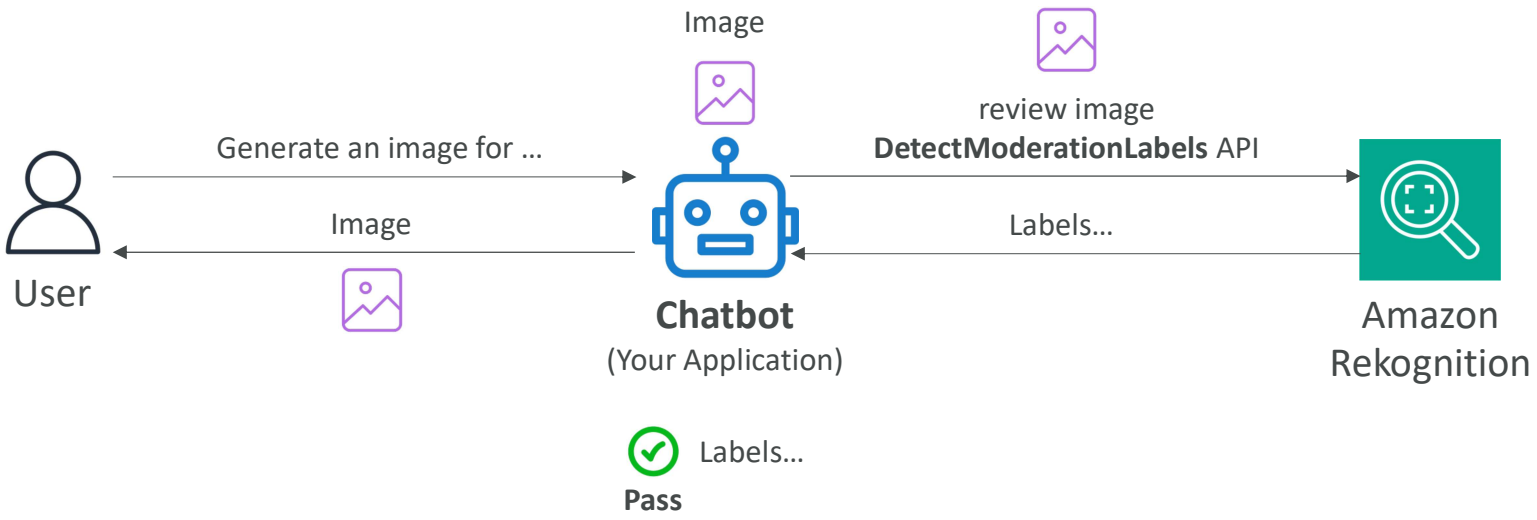


Amazon Rekognition – Content Moderation

- Automatically detect inappropriate, unwanted, or offensive content
- Example: filter out harmful images in social media, broadcast media, advertising…
- Bring down human review to 1-5% of total content volume
- Integrated with Amazon Augmented AI (Amazon A2I) for human review
- Custom Moderation Adaptors
  - Extends Rekognition capabilities by providing your own labeled set of images
  - Enhances the accuracy of Content Moderation or create a specific use case of Moderation

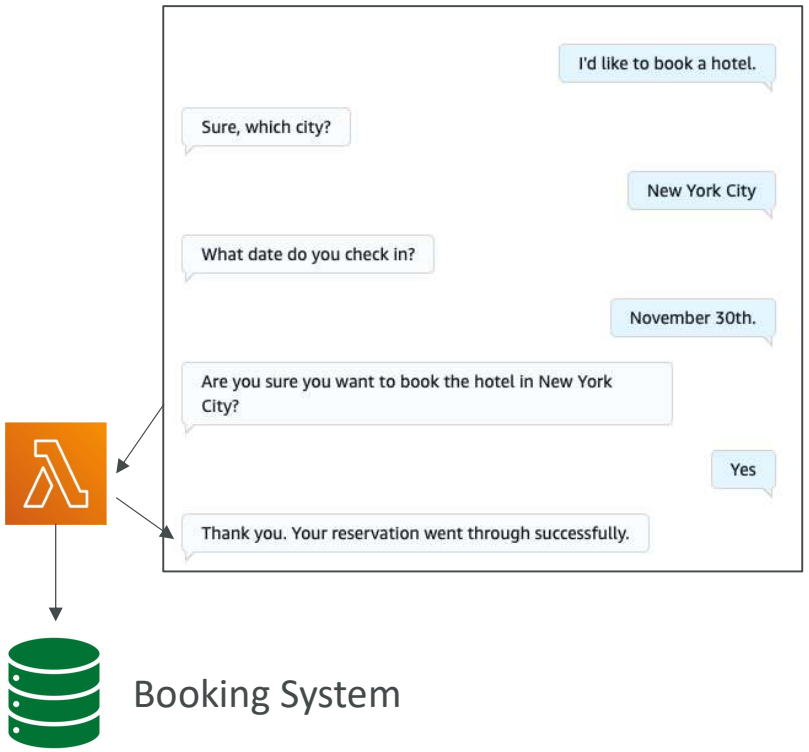


Content Moderation API – Diagram



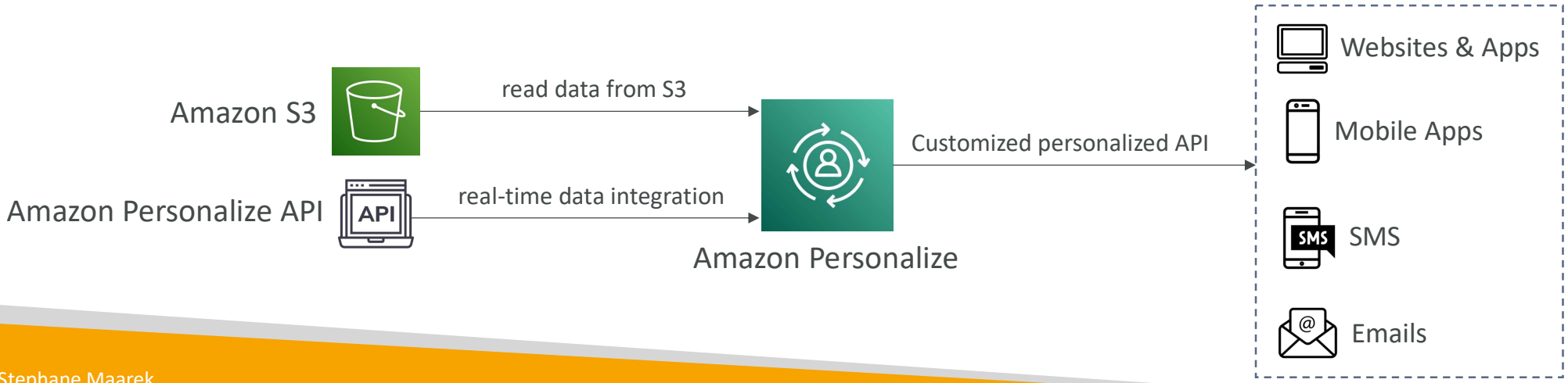
Amazon Lex

- Build chatbots quickly for your applications using voice and text
- Example: a chatbot that allows your customers to order pizzas or book a hotel
- Supports multiple languages
- Integration with AWS Lambda, Connect, Comprehend, Kendra
- The bot automatically understands the user intent to invoke the correct Lambda function to "fulfill the intent"
- The bot will ask for "Slots" (input parameters) if necessary



Amazon Personalize

- Fully managed ML-service to build apps with real-time personalized recommendations
- Example: personalized product recommendations/re-ranking, customized direct marketing
  - Example: User bought gardening tools, provide recommendations on the next one to buy
- Same technology used by Amazon.com
- Integrates into existing websites, applications, SMS, email marketing systems, …
- Implement in days, not months (you don't need to build, train, and deploy ML solutions)
- Use cases: retail stores, media and entertainment…



Amazon Personalize – Recipes

- Algorithms that are prepared for specific use cases
- You must provide the training configuration on top of the recipe
- Example recipes:
  - Recommending items for users (USER_PERSONALIZATION recipes)
    - User-Personalization-v2
  - Ranking items for a user (PERSONALIZED_RANKING recipes)
    - Personalized-Ranking-v2
  - Recommending trending or popular items (POPULAR_ITEMS recipes)
    - Trending-Now, Popularity-Count
  - Recommending similar items (RELATED_ITEMS recipes)
    - Similar-Items
  - Recommending the next best action (PERSONALIZED_ACTIONS recipes)
    - Next-Best-Action
  - Getting user segments (USER_SEGMENTATION recipes)
    - Item-Affinity
- NOTE: recipes and personalize are for recommendations

Amazon Textract

- Automatically extracts text, handwriting, and data from any scanned documents using AI and ML

analyze

Amazon Textract

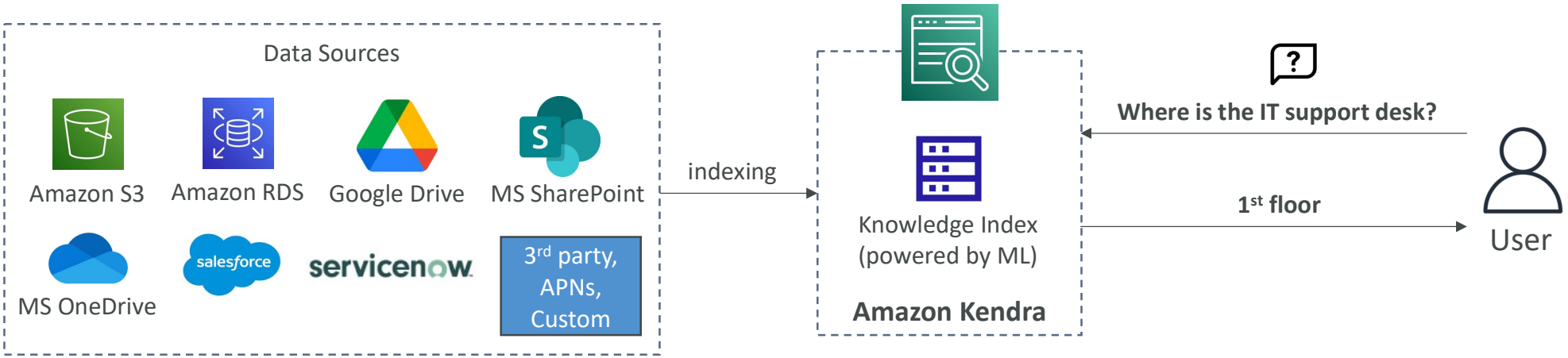
result

```
{
   "Document ID": "123456789-005",
   "Name": "",
   "SEX": "F",
   "DOB": "23.05.1997",
   ...
}
```

- Extract data from forms and tables
- Read and process any type of document (PDFs, images, …)
- Use cases:
  - Financial Services (e.g., invoices, financial reports)
  - Healthcare (e.g., medical records, insurance claims)
  - Public Sector (e.g., tax forms, ID documents, passports)

Amazon Kendra

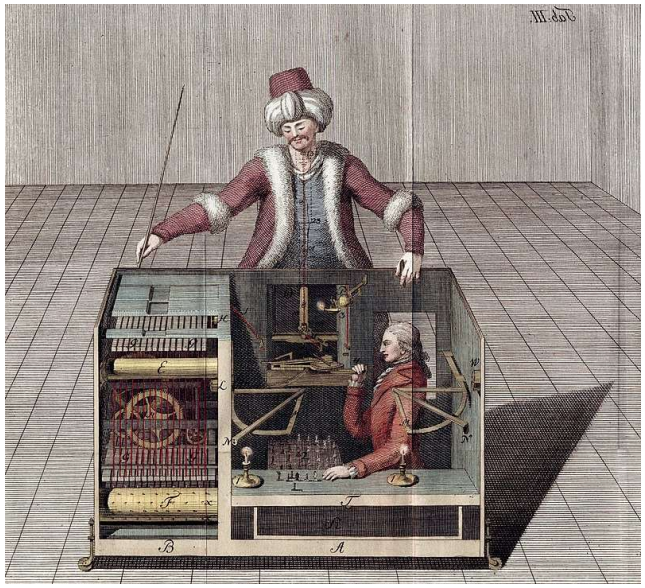
- Fully managed **document search service** powered by Machine Learning
- Extract answers from within a document (text, pdf, HTML, PowerPoint, MS Word, FAQs…)
- Natural language search capabilities
- Learn from user interactions/feedback to promote preferred results **(Incremental Learning)**
- Ability to manually fine-tune search results (importance of data, freshness, custom, …)



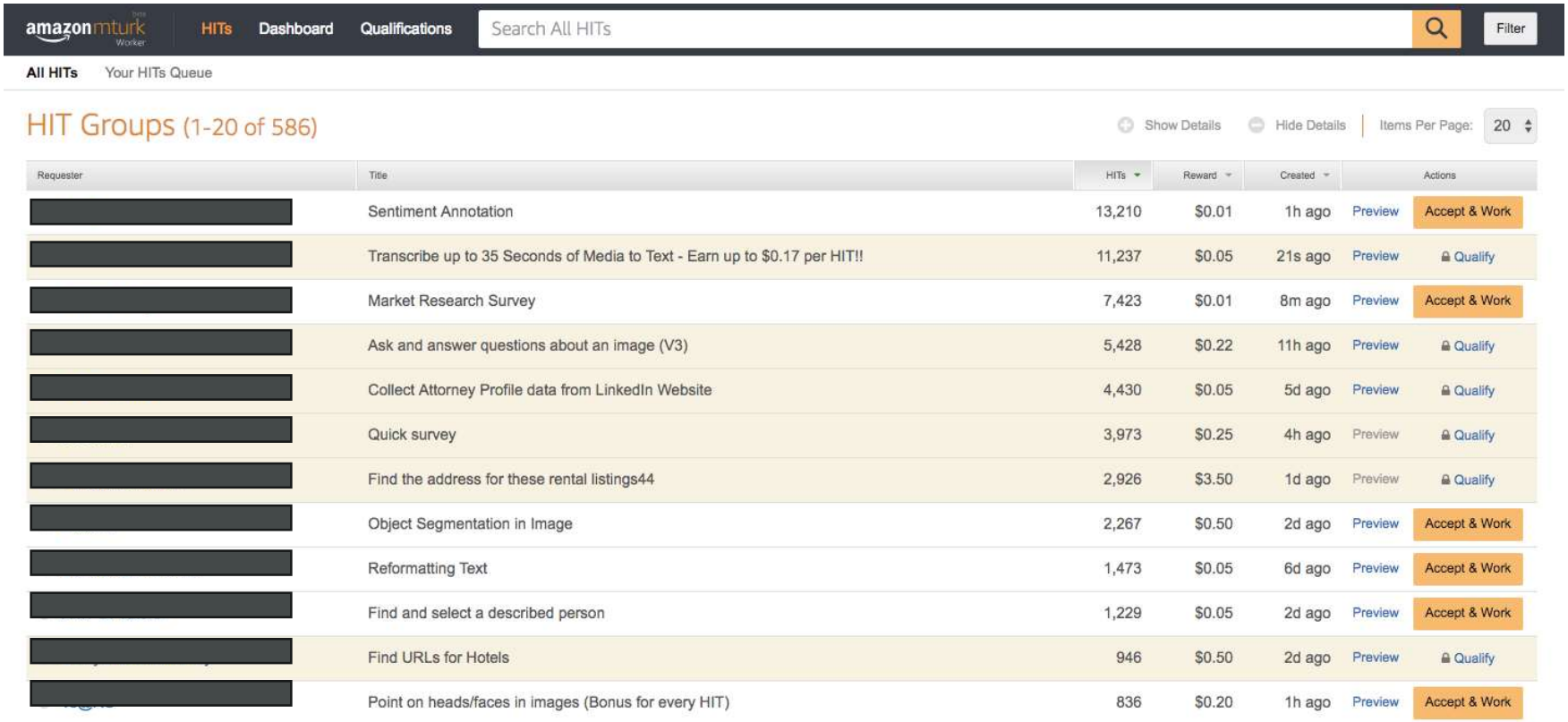
Amazon Mechanical Turk

- Crowdsourcing marketplace to perform simple human tasks
- Distributed virtual workforce

- Example:
  - You have a dataset of 10,000,000 images and you want to labels these images
  - You distribute the task on Mechanical Turk and **humans** will tag those images
  - You set the reward per image (for example $0.10 per image)
- Use cases: image classification, data collection, business processing
- **Integrates with Amazon A2I, SageMaker Ground Truth…**

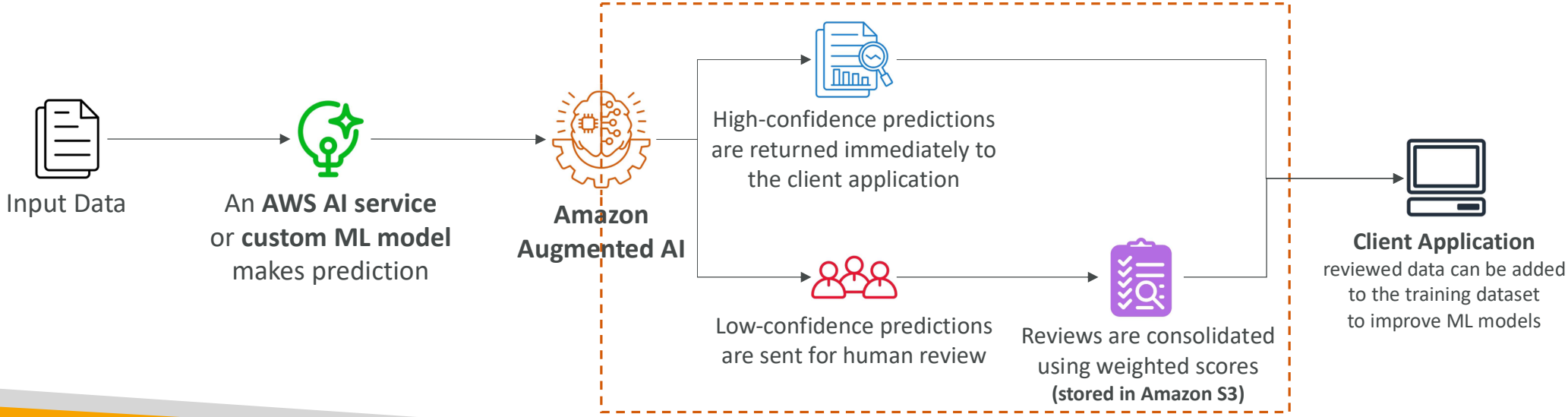


Amazon Mechanical Turk



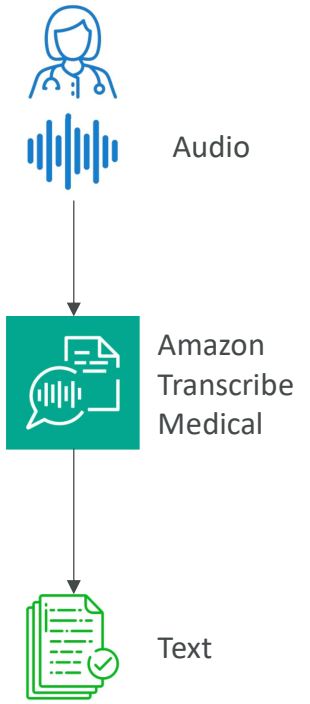
Amazon Augmented AI (A2I)

- Human oversight of Machine Learning predictions in production
  - Can be your own employees, over 500,000 contractors from AWS, or AWS Mechanical Turk
  - Some vendors are pre-screened for confidentiality requirements
- The ML model can be built on AWS or elsewhere (SageMaker, Rekognition…)



Amazon Transcribe Medical

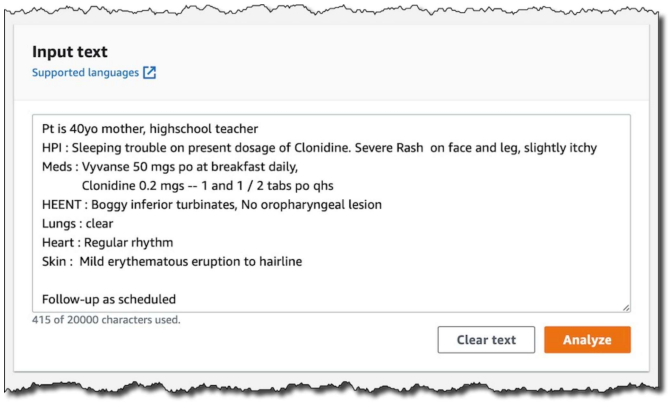
- Automatically convert medical-related speech to text (HIPAA compliant)
- Ability to transcribes medical terminologies such as:
  - Medicine names
  - Procedures
  - Conditions and diseases
- Supports both real-time (microphone) and batch (upload files) transcriptions
- Use cases:
  - Voice applications that enable physicians to dictate medical notes
  - Transcribe phone calls that report on drug safety and side effects



Amazon Comprehend Medical

- Amazon Comprehend Medical detects and returns useful information in unstructured clinical text:
  - Physician's notes
  - Discharge summaries
  - Test results
  - Case notes
- **Uses NLP to detect Protected Health Information (PHI)** – *DetectPHI* API
- Store your documents in Amazon S3
- Analyze real-time data with Kinesis Data Firehose
- Use Amazon Transcribe to transcribe patient narratives into text that can be analyzed by Amazon Comprehend Medical

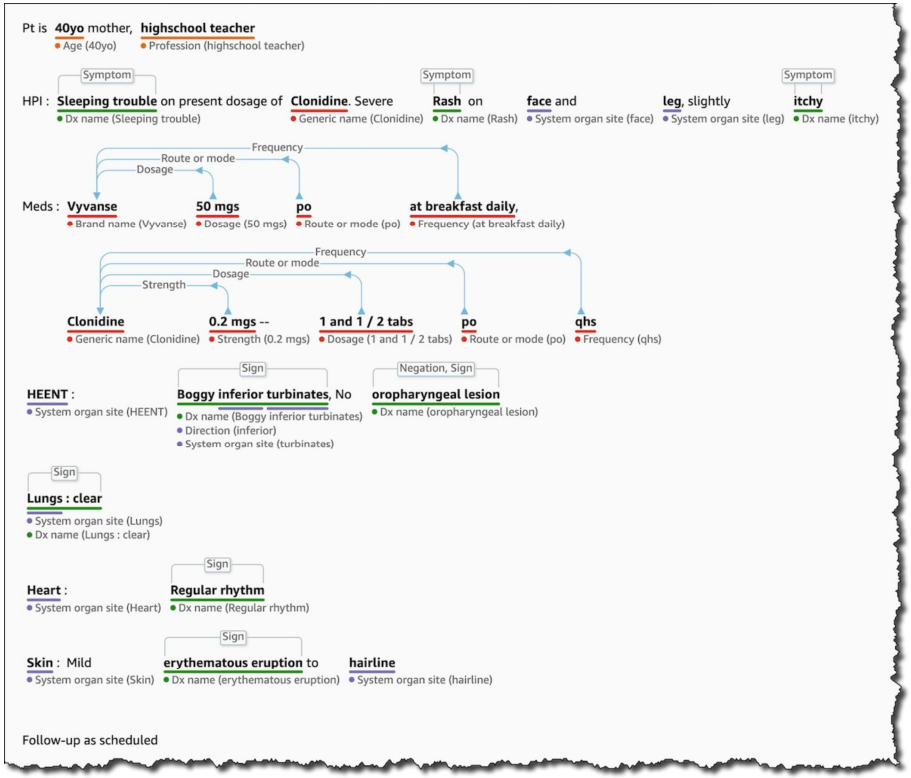
Amazon Comprehend Medical



OR

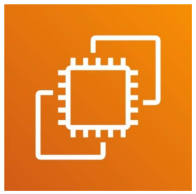
Amazon S3

Comprehend Medical



https://aws.amazon.com/blogs/aws/new-amazon-comprehend-medical-adds-ontology-linking/

Amazon EC2



- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- It mainly consists in the capability of :
  - Renting virtual machines (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across machines (ELB)
  - Scaling the services using an auto-scaling group (ASG)

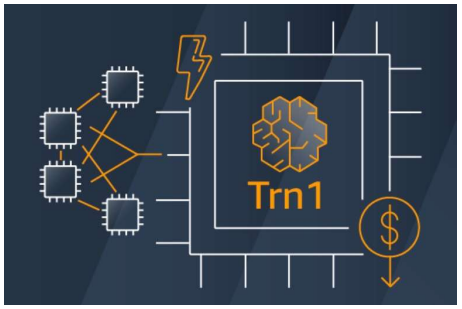
- Knowing EC2 is fundamental to understand how the Cloud works

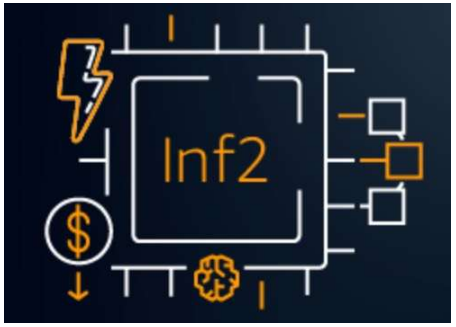
EC2 sizing & configuration options

- Operating System (**OS**): Linux, Windows or Mac OS
- How much compute power & cores (**CPU**)
- How much random-access memory (**RAM**)
- How much storage space:
  - Network-attached (**EBS & EFS**)
  - hardware (**EC2 Instance Store**)
- Network card: speed of the card, Public IP address
- Firewall rules: **security group**
- Bootstrap script (configure at first launch): EC2 User Data

Amazon's Hardware for AI

- GPU-based EC2 Instances (P3, P4, P5…, G3…G6…)
- AWS Trainium
  - ML chip built to perform Deep Learning on 100B+ parameter models
  - Trn1 instance has for example 16 Trainium Accelerators
  - 50% cost reduction when training a model
- AWS Inferentia
  - ML chip built to deliver inference at high performance and low cost
  - Inf1, Inf2 instances are powered by AWS Inferentia
  - Up to 4x throughput and 70% cost reduction
- Trn & Inf have the lowest environmental footprint

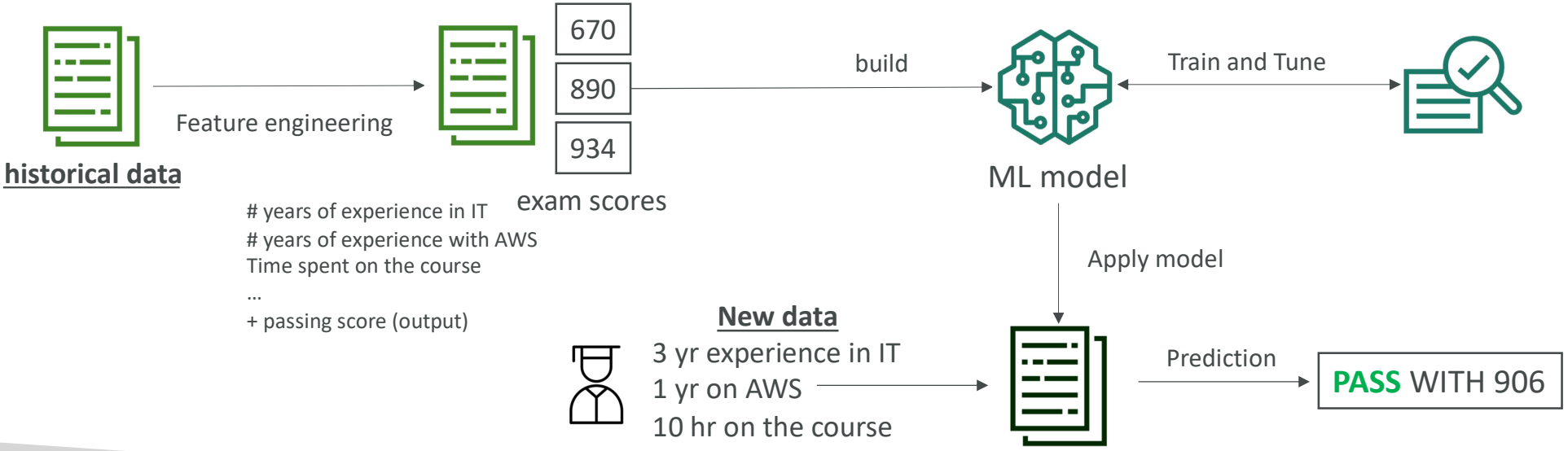




Amazon SageMaker

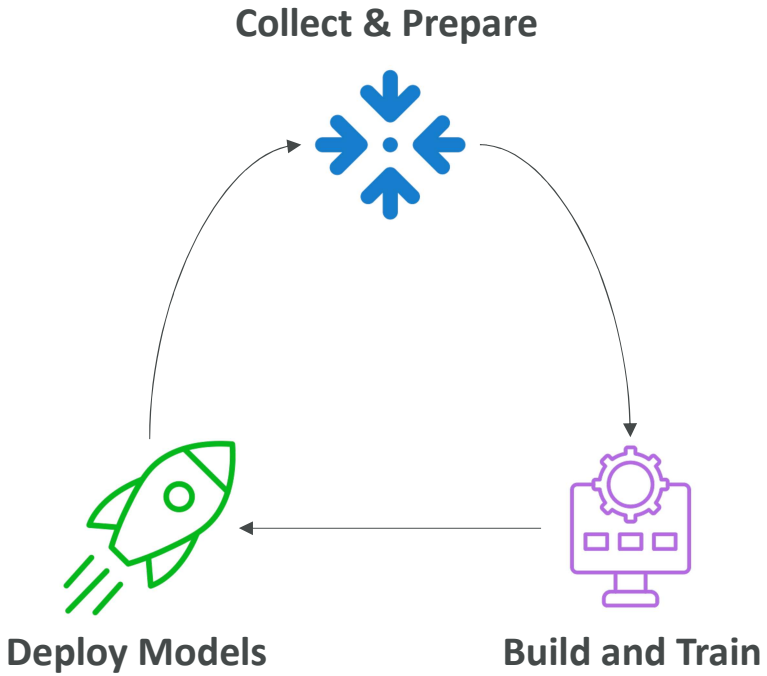
Amazon SageMaker AI

- Fully managed service for developers / data scientists to build ML models
- Typically, difficult to do all the processes in one place + provision servers
- Example: predicting your AWS exam score



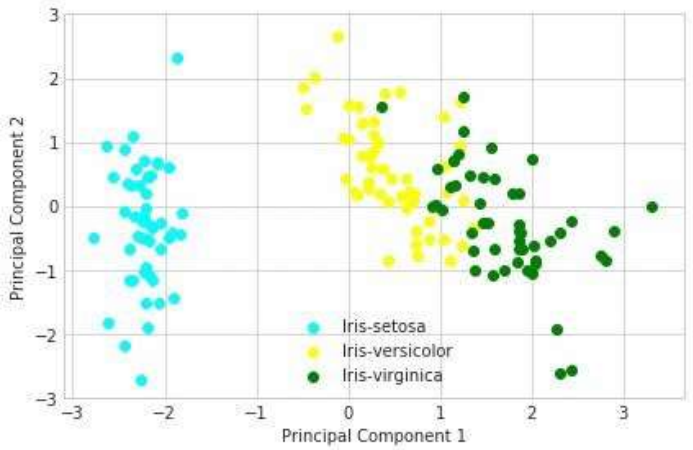
SageMaker – End-to-End ML Service

- Collect and prepare data
- Build and train machine learning models
- Deploy the models and monitor the performance of the predictions



SageMaker – Built-in Algorithms (extract)

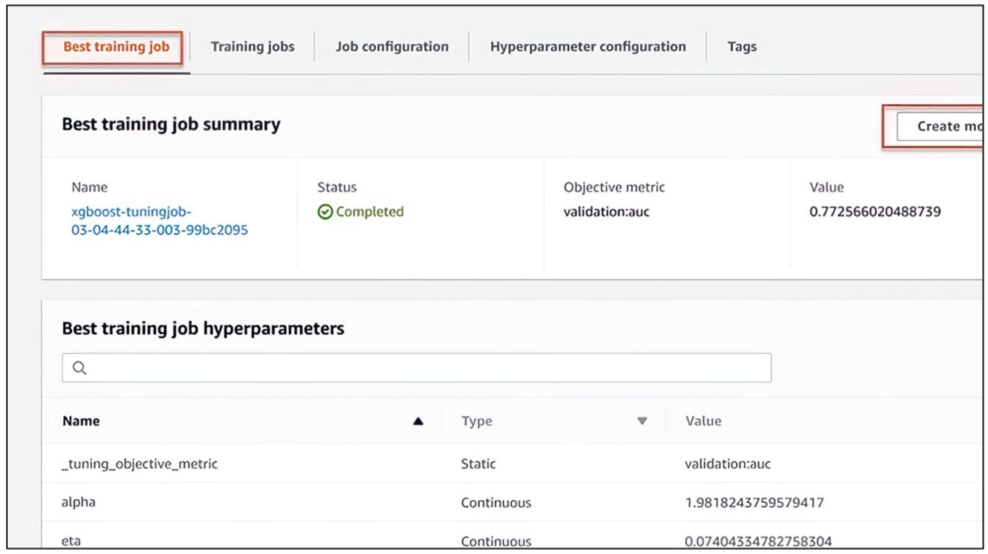
- Supervised Algorithms
  - Linear regressions and classifications
  - KNN Algorithms (for classification)
- Unsupervised Algorithms
  - Principal Component Analysis (PCA) – reduce number of features
  - K-means – find grouping within data
  - Anomaly Detection
- Textual Algorithms – NLP, summarization…
- Image Processing – classification, detection…



https://aws.amazon.com/blogs/machine-learning/running-principal-component-analysis-in-amazon-sagemaker/

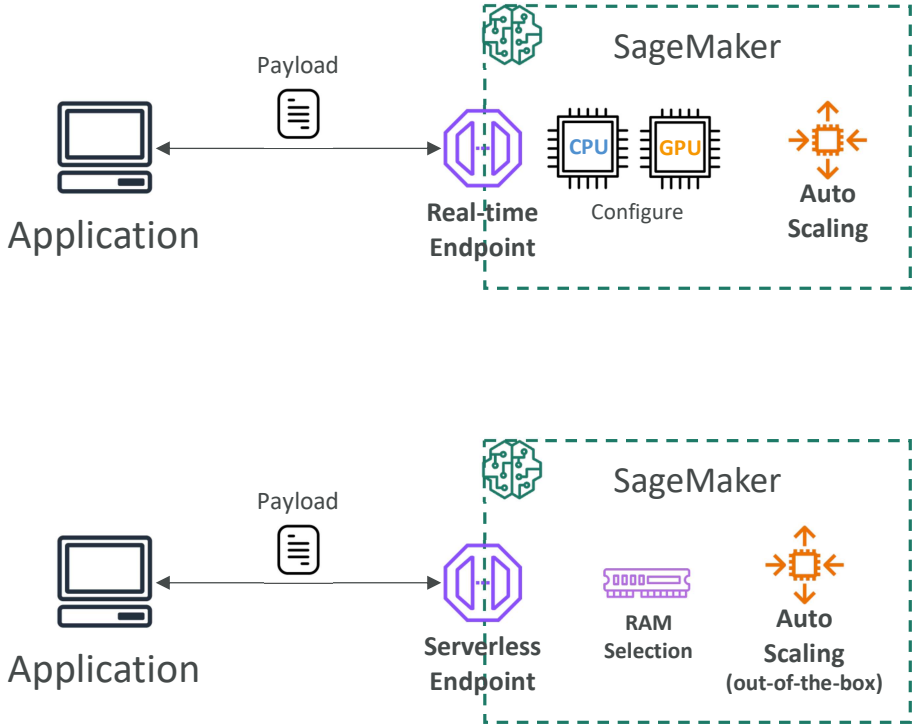
SageMaker – Automatic Model Tuning (AMT)

- Define the **Objective Metric**
- AMT automatically chooses hyperparameter ranges, search strategy, maximum runtime of a tuning job, and early stop condition
- Saves you time and money
- Helps you not wasting money on suboptimal configurations



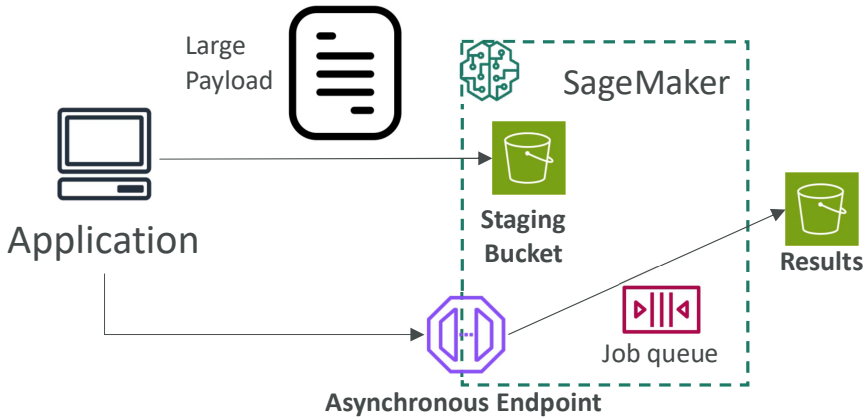
SageMaker – Model Deployment & Inference

- Deploy with one click, automatic scaling, no servers to manage (as opposed to self-hosted)
- Managed solution: reduced overhead
- Real-time
  - One prediction at a time
- Serverless
  - Idle period between traffic spikes
  - Can tolerate more latency (cold starts)

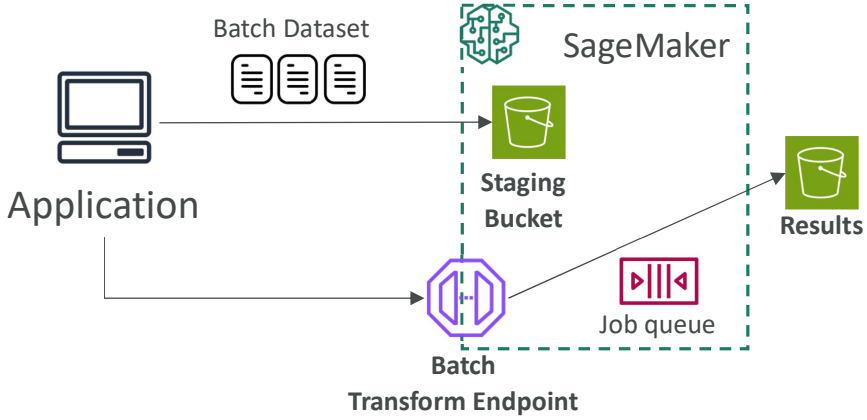


SageMaker – Model Deployment & Inference

- Asynchronous
  - For large payload sizes up to 1GB
  - Long processing times
  - Near-real time latency requirements
  - Request and responses are in Amazon S3



- Batch
  - Prediction for an entire dataset (multiple predictions)
  - Request and responses are in Amazon S3

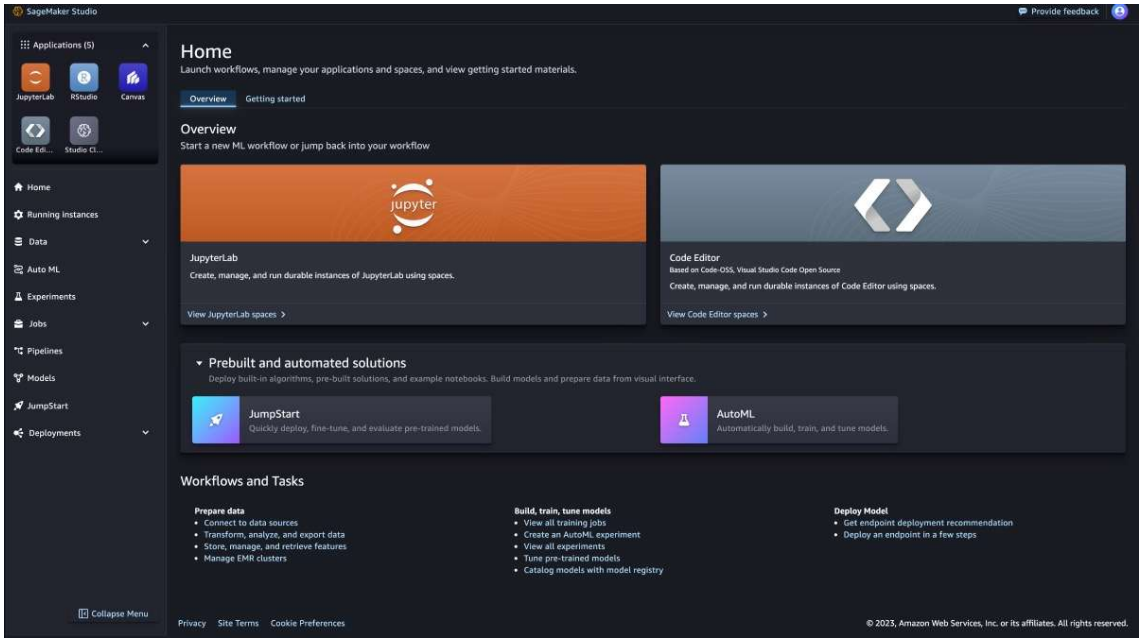


SageMaker Model Deployment Comparison

Inference Type	Latency	Payload Size	Processing Time	Use Case
Real-time Inference	Low (milliseconds to seconds)	Up to 6 MB (one record)	Max 60 seconds	Fast, near-instant predictions for web/mobile apps
Serverless Inference	Low (milliseconds to seconds)	Up to 4 MB (one record)	Max 60 seconds	Sporadic, short-term inference without infrastructure, can tolerate cold starts
Asynchronous Inference	Medium to High “near real-time”	Up to 1 GB (one record)	Max 1 hour	Large payloads and workloads requiring longer processing times
Batch Transform	High (minutes to hours)	Up to 100 MB per invocation (per mini batch)	Max 1 hour	Bulk processing for large datasets Concurrent processing

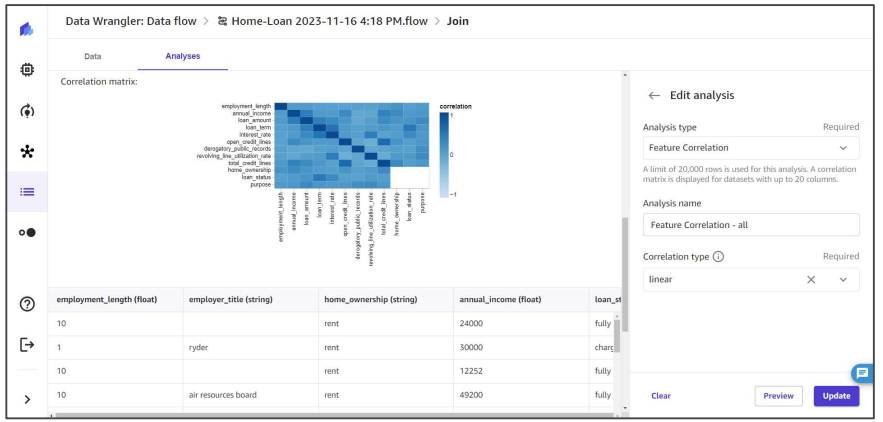
SageMaker Studio

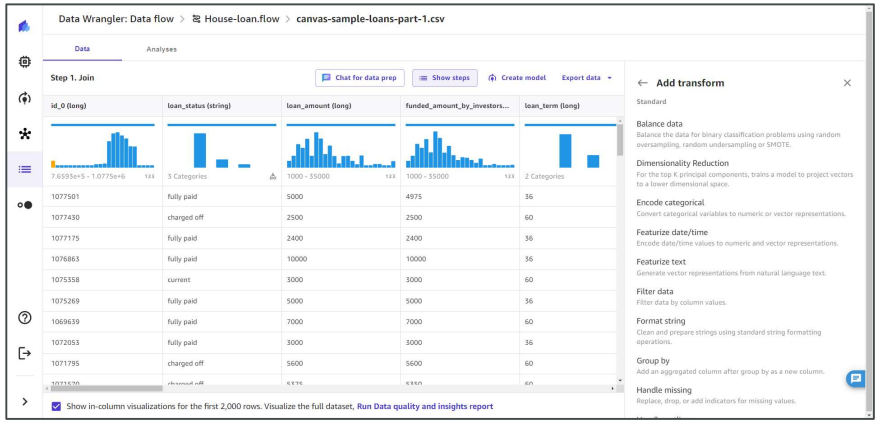
- End-to-end ML development from a unified interface
- Team collaboration
- Tune and debug ML models
- Deploy ML models
- Automated workflows



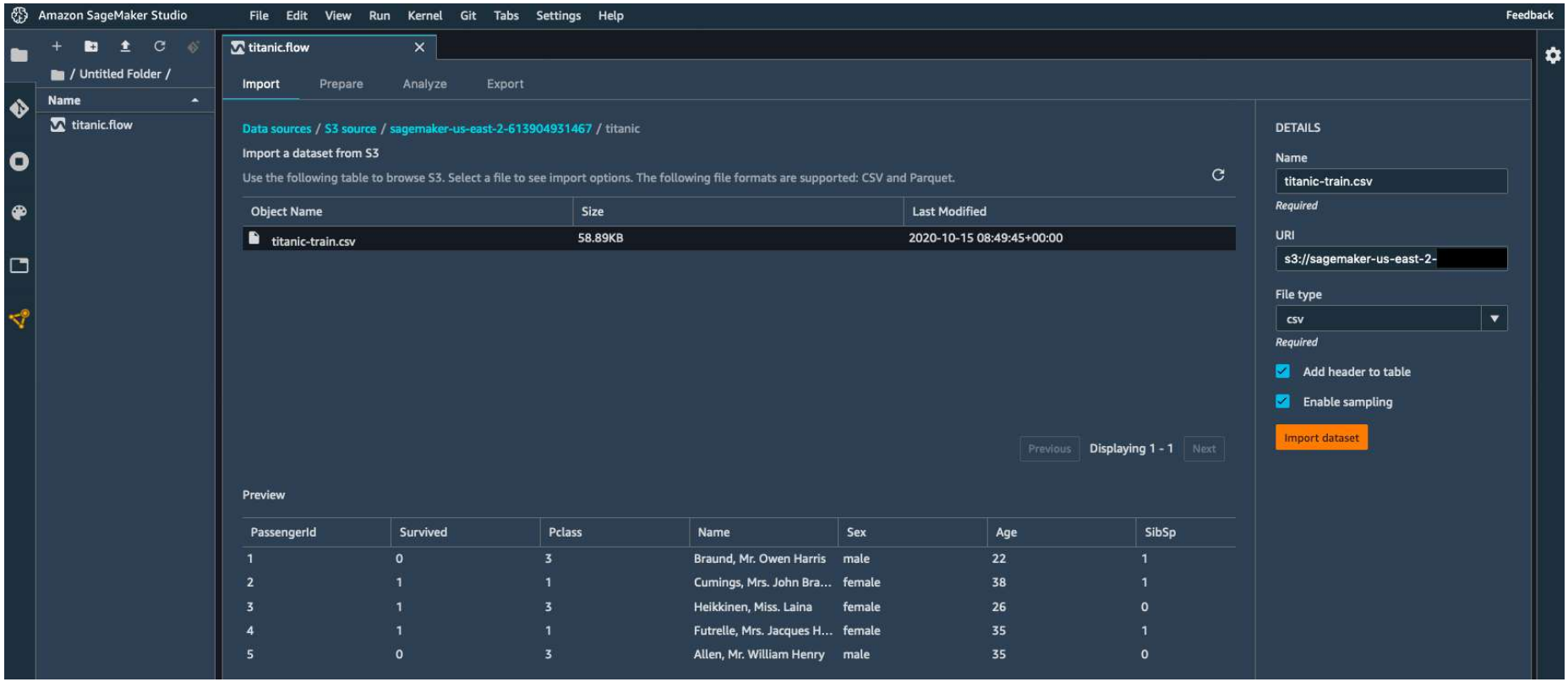
SageMaker – Data Wrangler

- Prepare tabular and image data for machine learning
- Data preparation, transformation and feature engineering
- Single interface for data selection, cleansing, exploration, visualization, and processing
- SQL support
- Data Quality tool

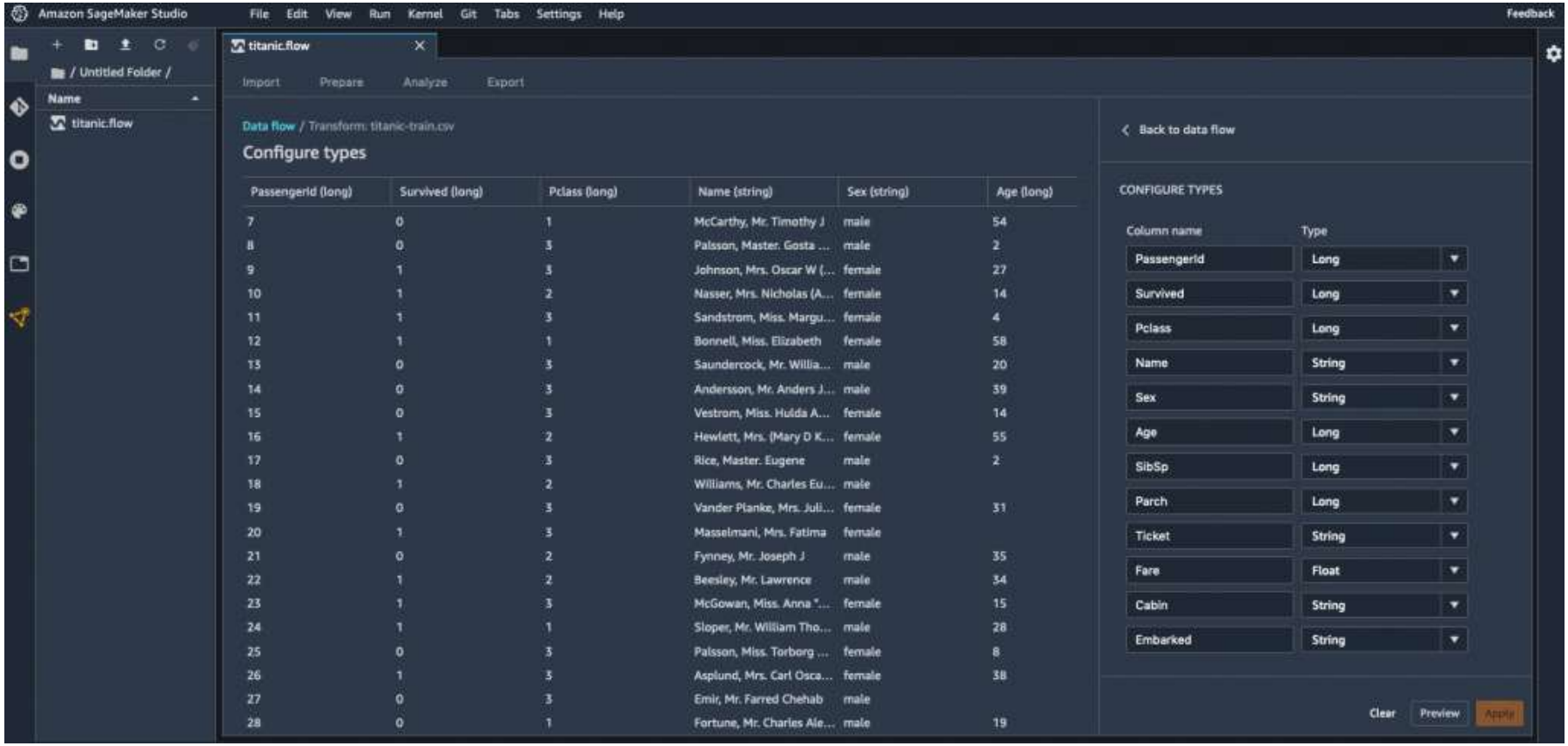




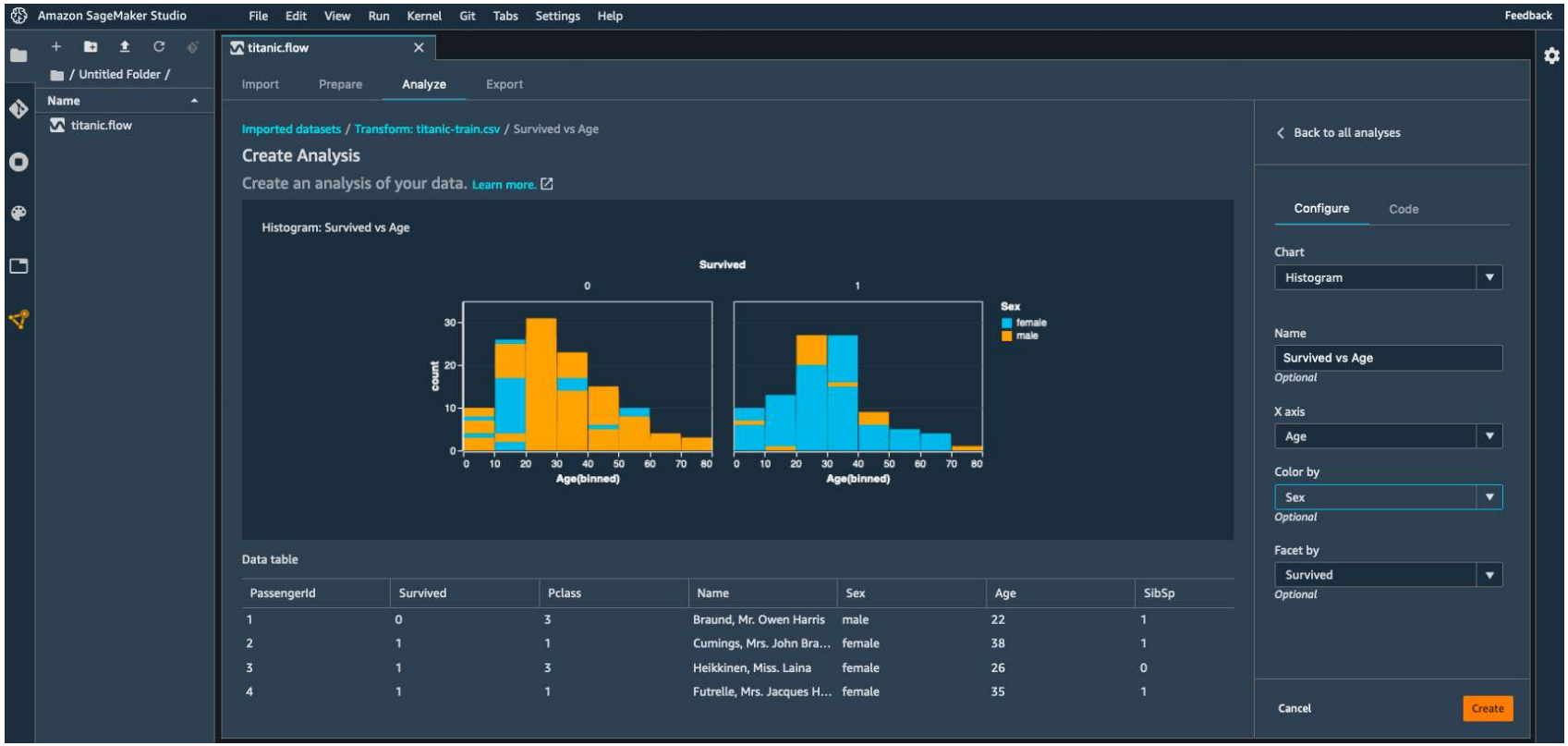
Data Wrangler: Import Data



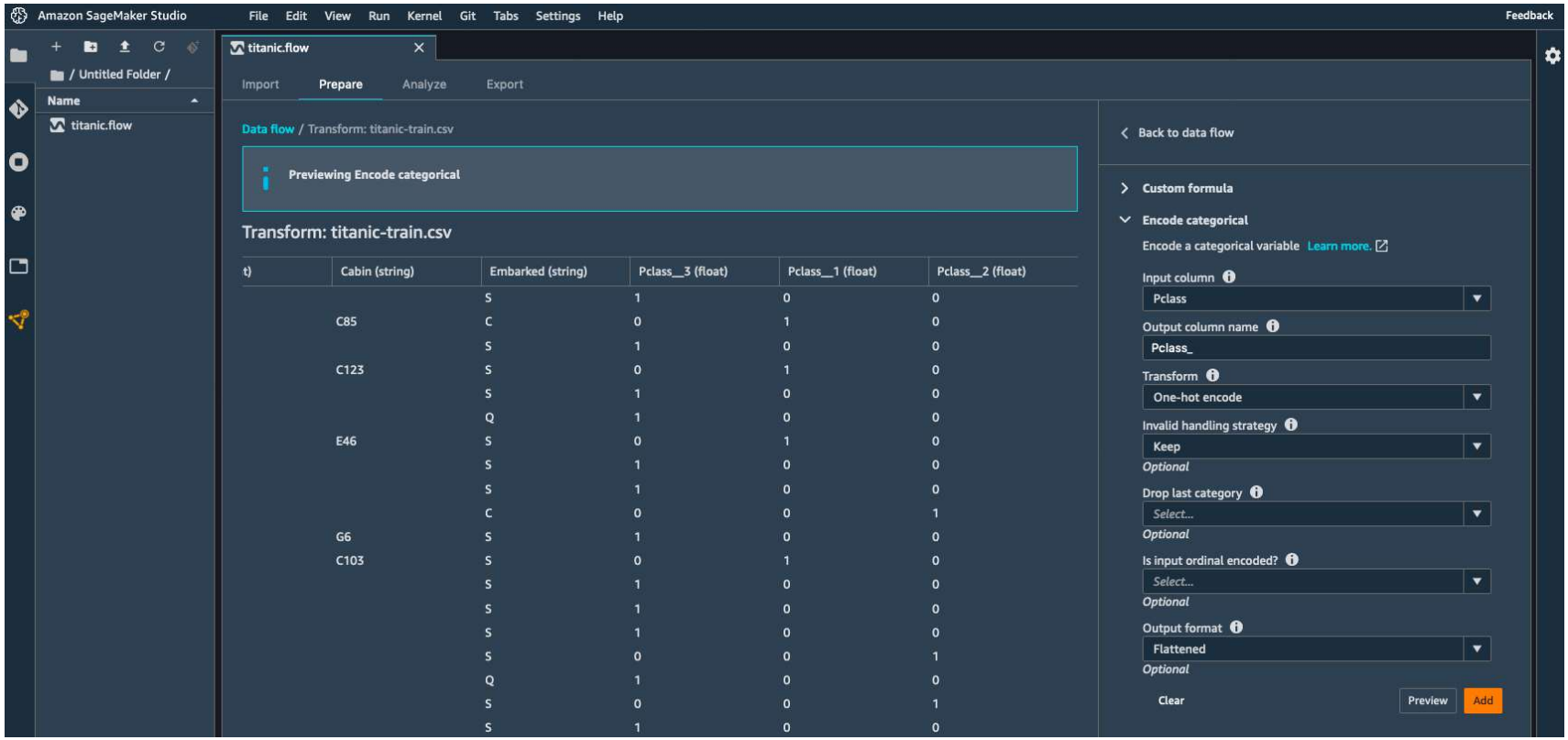
Data Wrangler: Preview Data



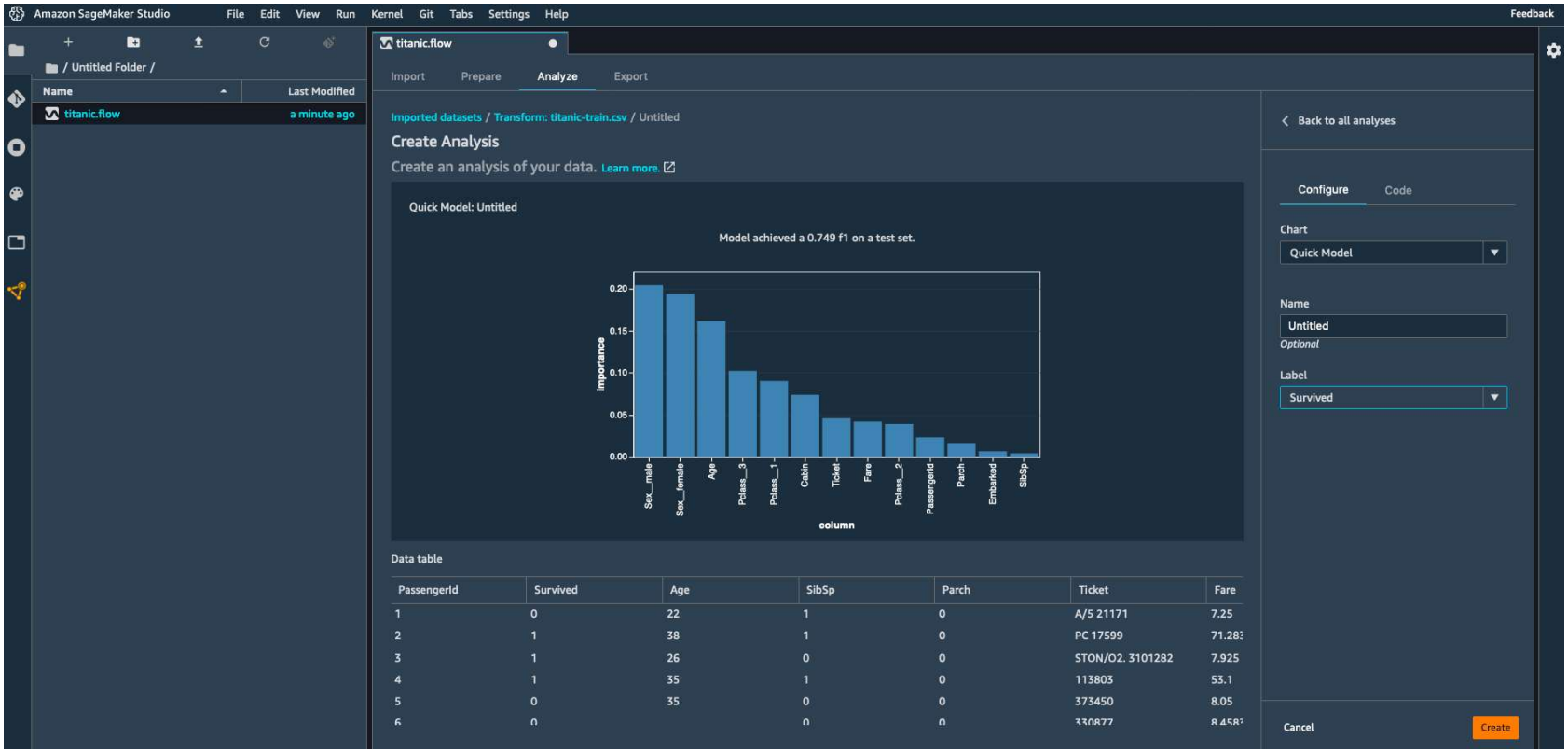
Data Wrangler: Visualize Data



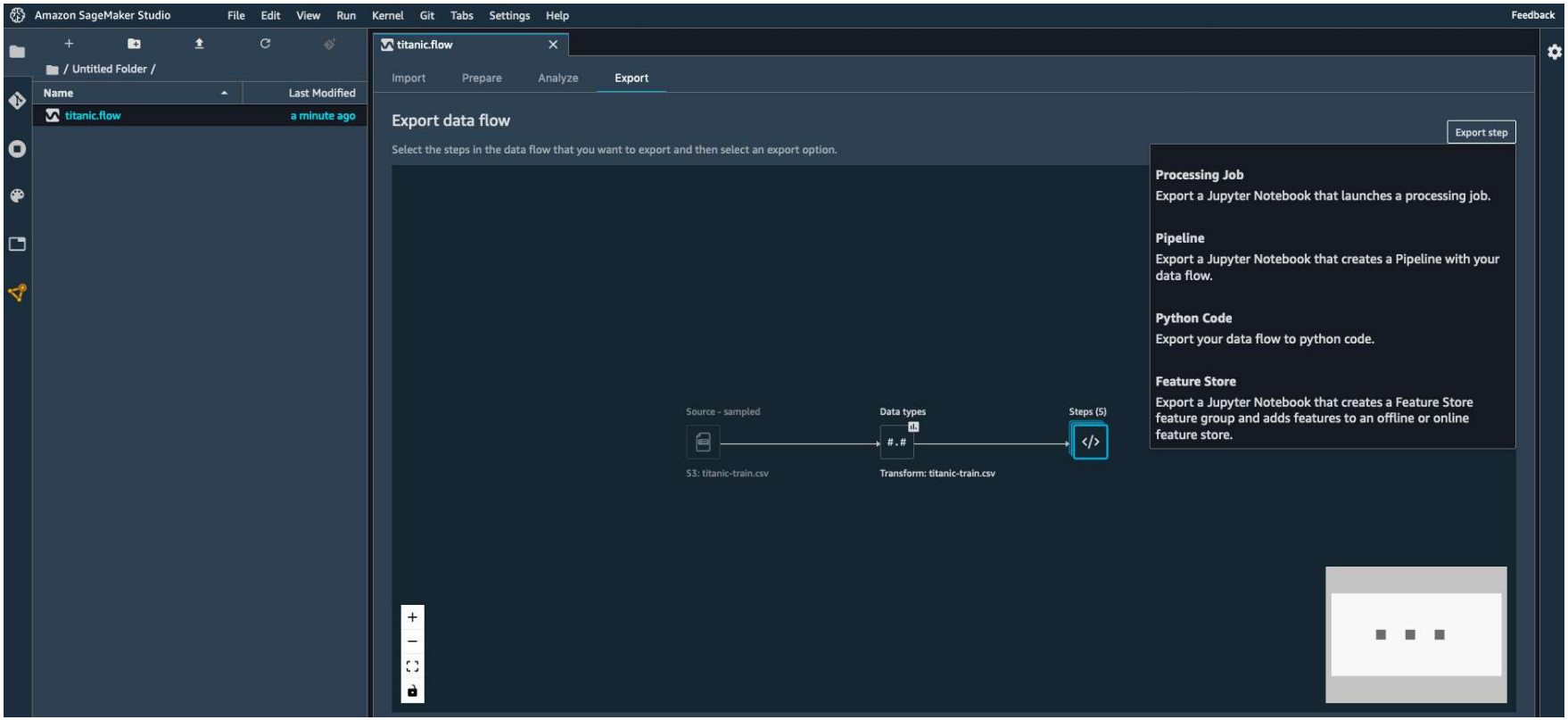
Data Wrangler: Transform Data



Data Wrangler: Quick Model



Data Wrangler: Export Data Flow



What are ML Features?

- Features are inputs to ML models used during training and used for inference
- Example - music dataset: song ratings, listening duration, and listener demographics
- Important to have high quality features across your datasets in your company for re-use

Before
Feature Engineering

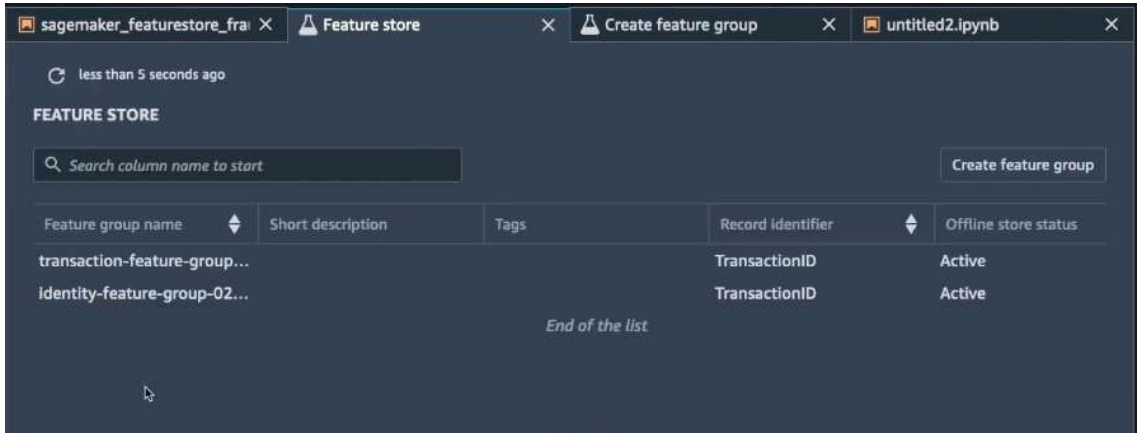
Customer_ID	Name	BirthDate	Purchase_Amount
1	Alice	**15-05-1993**	$200
2	Bob	**22-08-1978**	$300

After
Feature Engineering

Customer_ID	Name	Age	Purchase_Amount
1	Alice	**30**	$200
2	Bob	**45**	$300

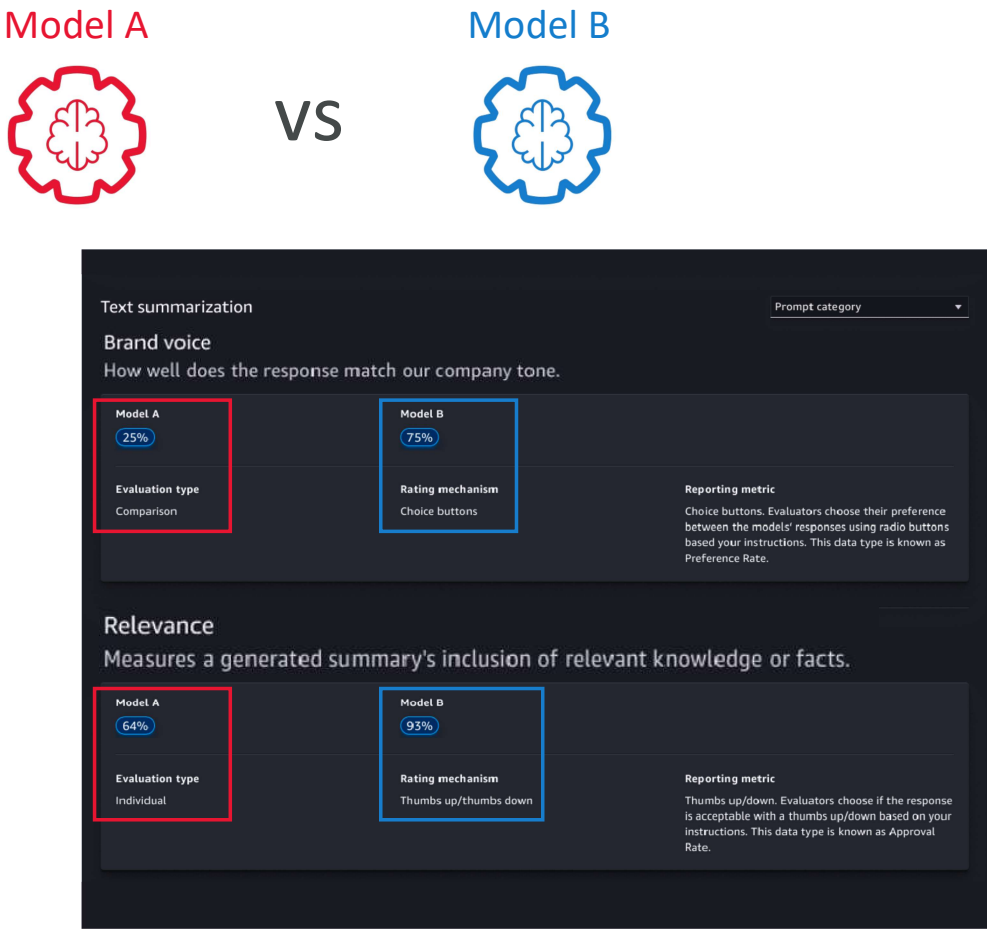
SageMaker – Feature Store

- Ingests features from a variety of sources
- Ability to define the transformation of data into feature from within Feature Store
- Can publish directly from SageMaker Data Wrangler into SageMaker Feature Store
- Features are discoverable within SageMaker Studio



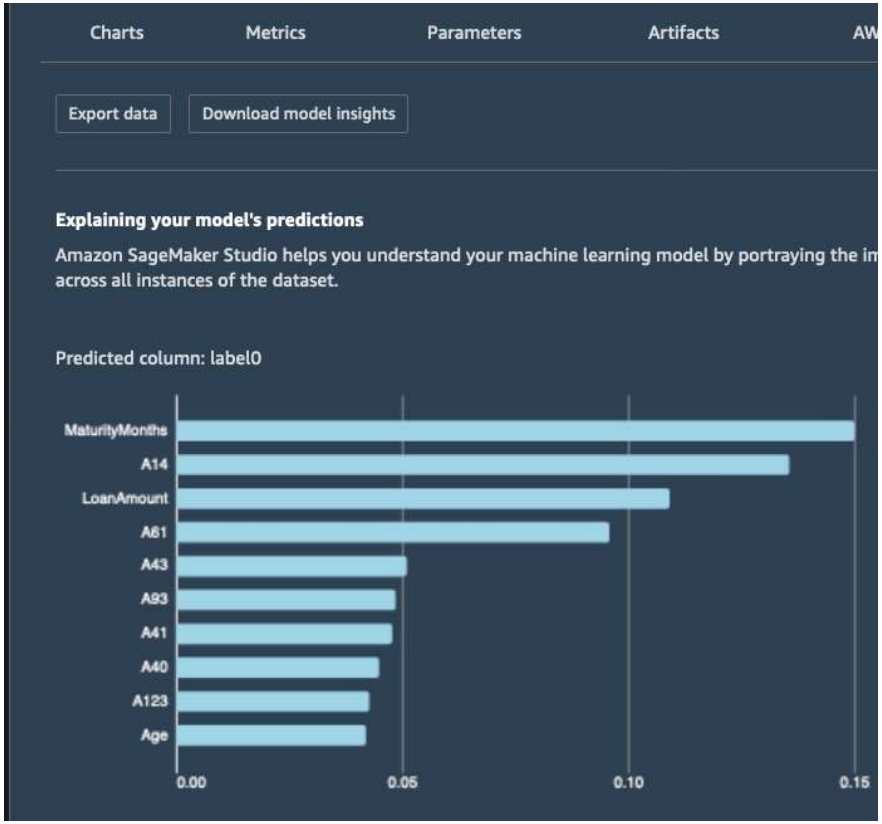
SageMaker Clarify

- Evaluate Foundation Models
- Evaluating human-factors such as friendliness or humor
- Leverage an AWS-managed team or bring your own employees
- Use built-in datasets or bring your own dataset
- Built-in metrics and algorithms
- Part of **SageMaker Studio**



SageMaker Clarify – Model Explainability

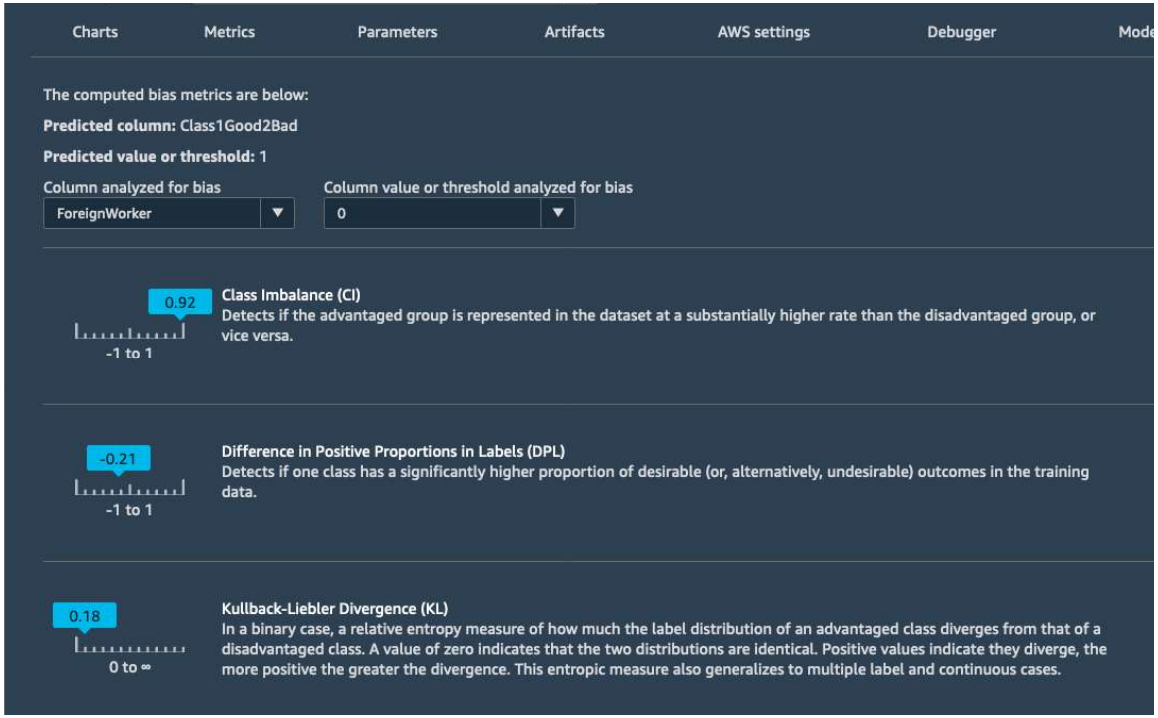
- A set of tools to help explain how machine learning (ML) models make predictions
- Understand model characteristics as a whole prior to deployment
- Debug predictions provided by the model after it's deployed
- Helps increase the trust and understanding of the model
- Example:
  - "Why did the model predict a negative outcome such as a loan rejection for a given applicant?"
  - "Why did the model make an incorrect prediction?"



https://noise.getoto.net/author/julien-simon/

SageMaker Clarify – Detect Bias (human)

- Ability to **detect and explain** biases in your **datasets and models**
- Measure bias using statistical metrics
- Specify input features and bias will be automatically detected



https://noise.getoto.net/author/julien-simon/

Different kind of biases (definitions)

- **Sampling bias**: Sampling bias occurs when the training data does not represent the full population fairly, leading to a model that over-represents or disproportionately affects certain groups
- **Measurement bias**: Measurement bias occurs when the tools or measurements used in data collection are flawed or skewed
- **Observer bias**: Observer bias happens when the person collecting or interpreting the data has personal biases that affect the results
- **Confirmation bias**: Confirmation bias is when individuals interpret or favor information that confirms their preconceptions. This is more applicable to human decision-making rather than automated model outputs.

- **Example:** an algorithm only flags people from specific ethnic groups, this is probably a sampling bias, and you need to perform **data augmentation for imbalanced classes**

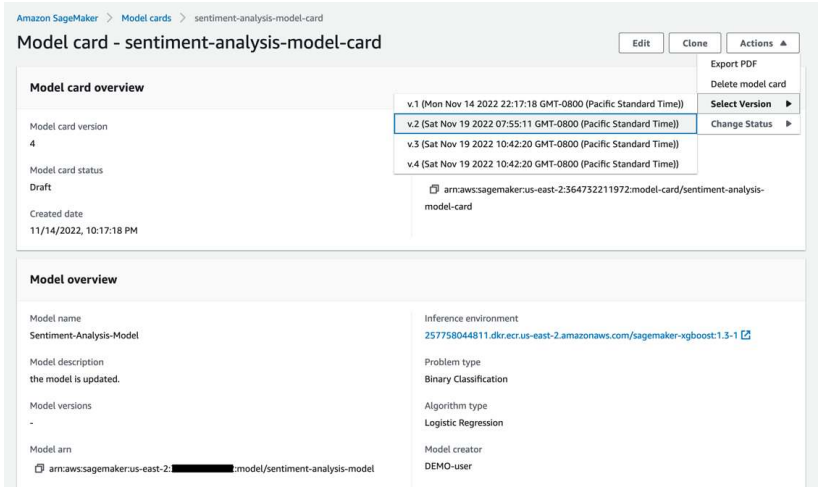
SageMaker Ground Truth

- RLHF – Reinforcement Learning from Human Feedback
  - Model review, customization and evaluation
  - Align model to human preferences
  - Reinforcement learning where human feedback is included in the "reward" function
- Human feedback for ML
  - Creating or evaluating your models
  - Data generation or annotation (create labels)
- Reviewers: Amazon Mechanical Turk workers, your employees, or third-party vendors
- SageMaker Ground Truth Plus: Label Data

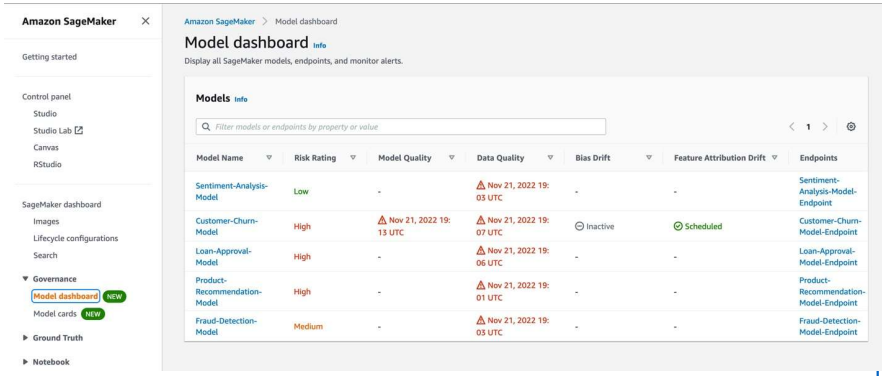
SageMaker – ML Governance

- SageMaker Model Cards
  - Essential model information
  - Example: intended uses, risk ratings, and training details
- SageMaker Model Dashboard
  - Centralized repository
  - Information and insights for all models
- SageMaker Role Manager
  - Define roles for personas
  - Example: data scientists, MLOps engineers

Model Card



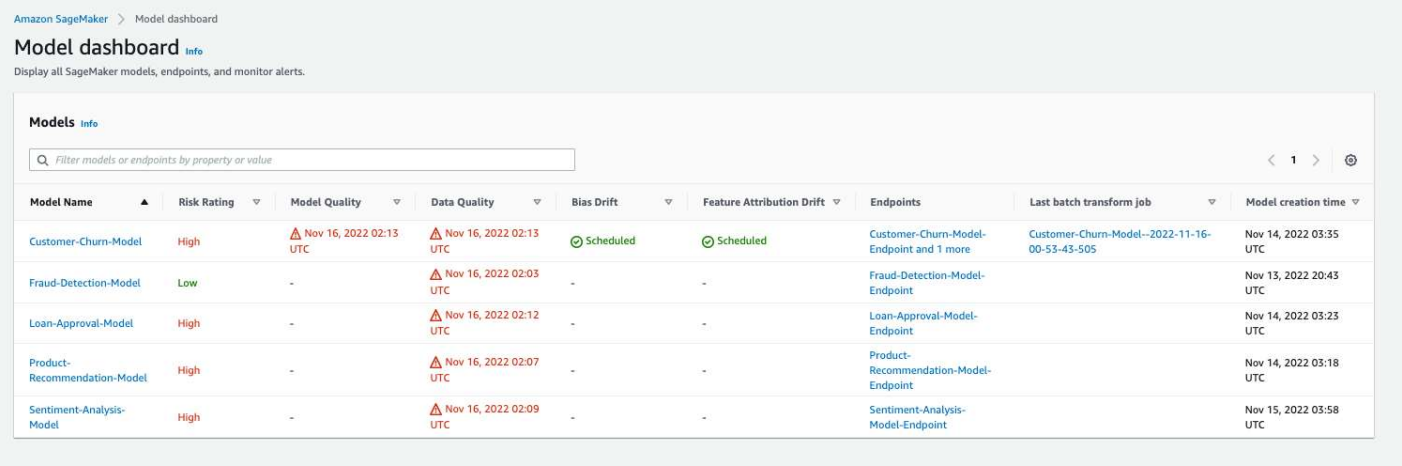
Model Dashboard



https://aws.amazon.com/blogs/machine-learning/improve-governance-of-your-machine-learning-models-with-amazon-sagemaker/

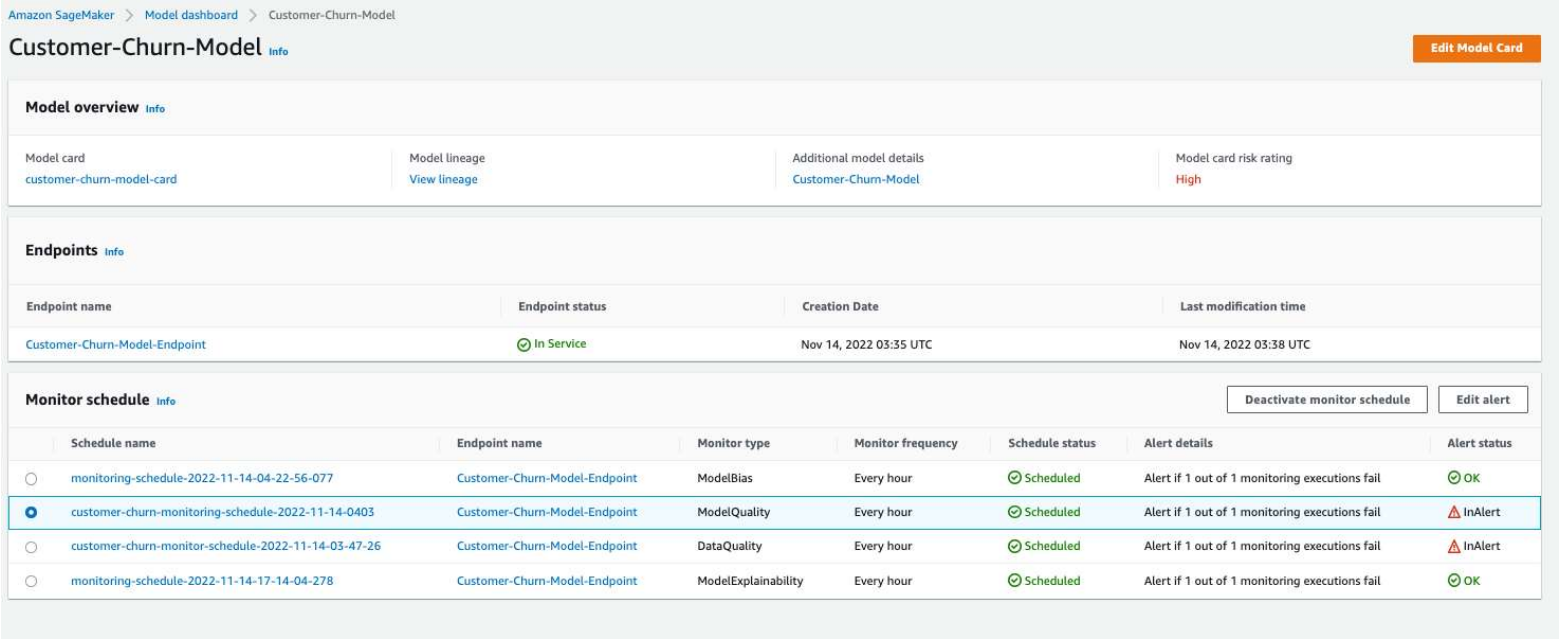
SageMaker – Model Dashboard

- Centralized portal where you can view, search, and explore all of your models
- Example: track which models are deployed for inference
- Can be accessed from the SageMaker Console
- Helps you find models that violate thresholds you set for data quality, model quality, bias, explainability…



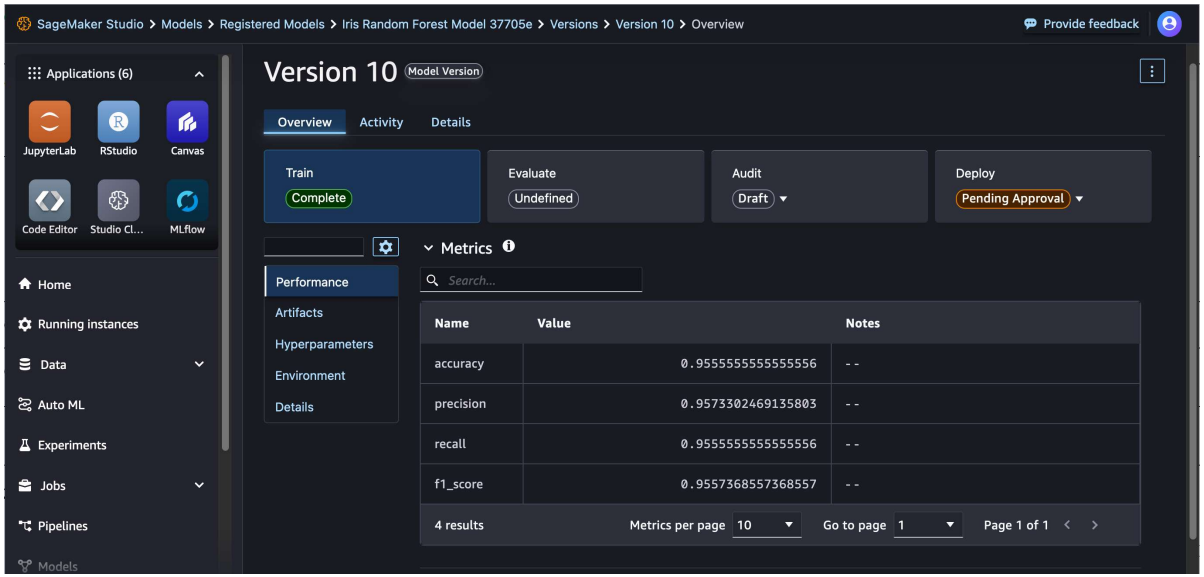
SageMaker – Model Monitor

- Monitor the quality of your model in production: continuous or on-schedule
- Alerts for deviations in the model quality: fix data & retrain model
- Example: loan model starts giving loans to people who don't have the correct credit score (drift)



SageMaker – Model Registry

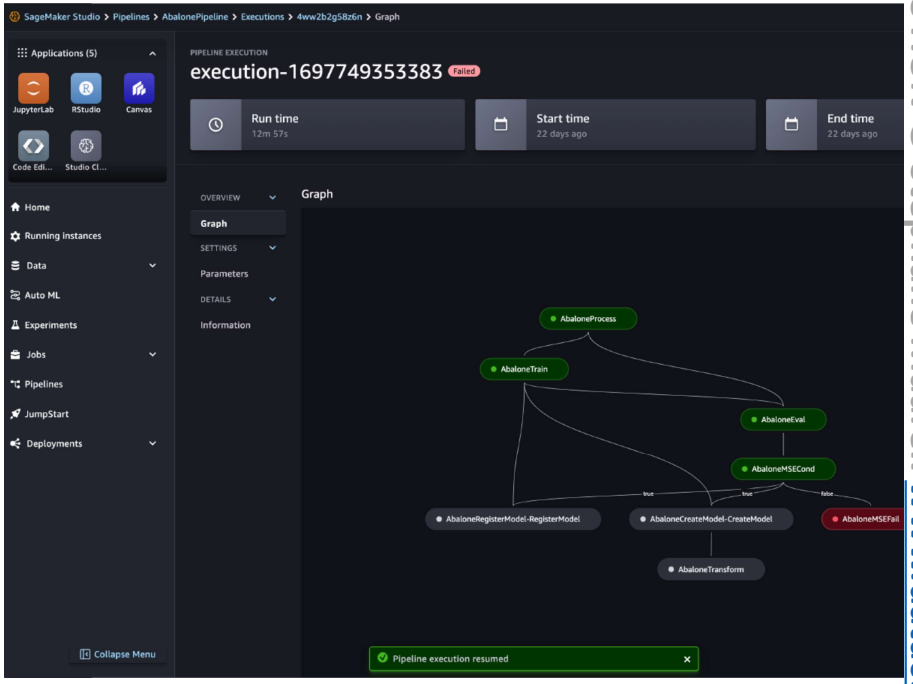
- Centralized repository allows you to track, manage, and version ML models
- Catalog models, manage model versions, associate metadata with a model
- **Manage approval status of a model**, automate model deployment, share models…



https://docs.aws.amazon.com/sagemaker/latest/dg/mlflow-track-experiments-model-registration.html

SageMaker Pipelines

- **SageMaker Pipeline** – a workflow that automates the process of building, training, and deploying a ML model
- Continuous Integration and Continuous Delivery (CI/CD) service for Machine Learning
- Helps you easily build, train, test, and deploy 100s of models **automatically**
- Iterate faster, reduce errors (no manual steps), repeatable mechanisms…



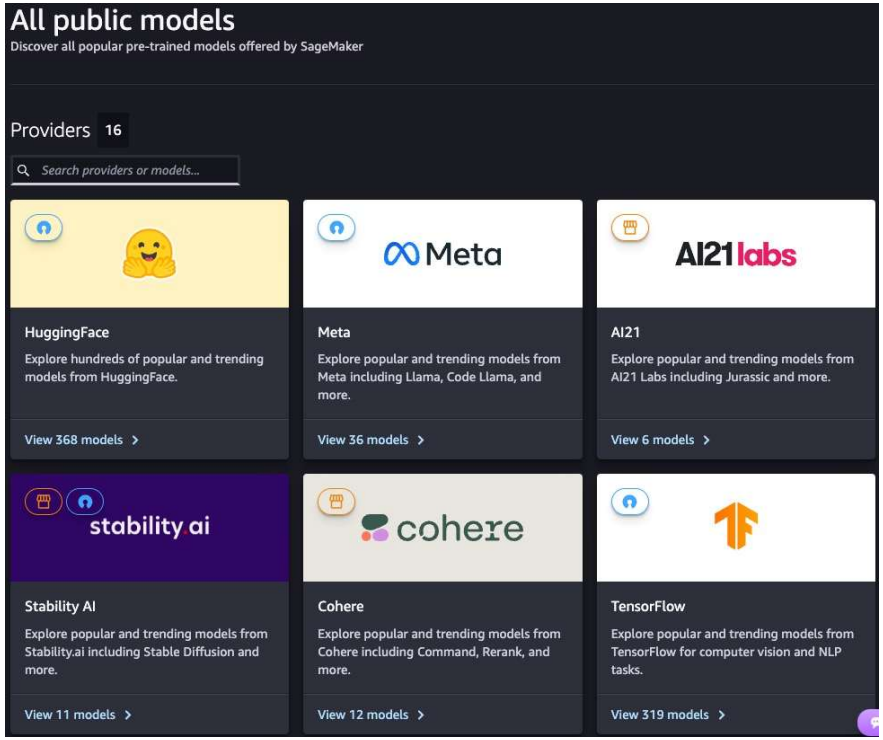
https://aws.amazon.com/sagemaker/pipelines/

SageMaker Pipelines

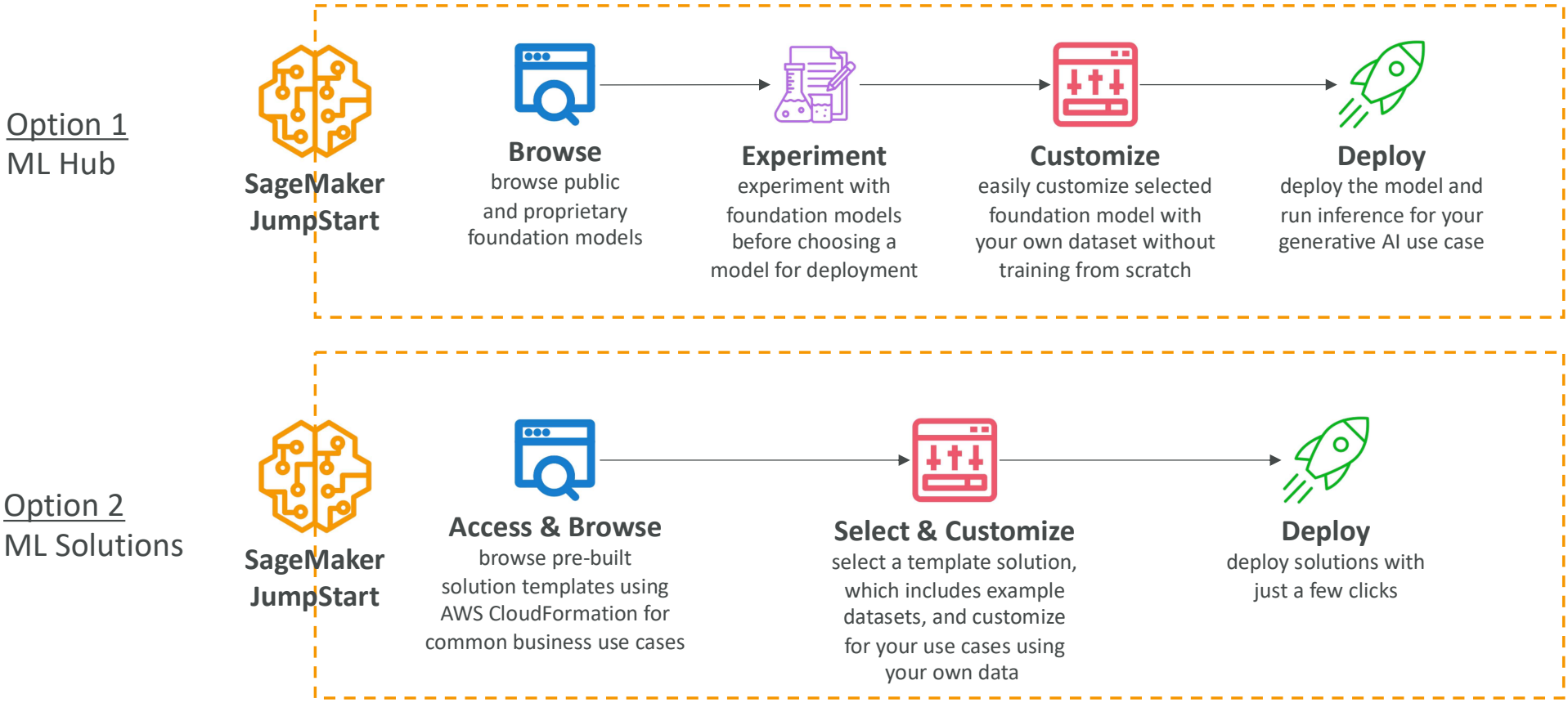
- Pipelines composed of Steps and each Step performs a specific task (e.g., data preprocessing, model training…)
- Supported Step Types:
  - **Processing** – for data processing (e.g., feature engineering)
  - **Training** – for training a model
  - **Tuning** – for hyperparameter tuning (e.g., Hyperparameter Optimization)
  - **AutoML** – to automatically train a model
  - **Model** – to create or register a SageMaker model
  - **ClarifyCheck** – perform drift checks against baselines (Data bias, Model bias, Model explainability)
  - **QualityCheck** – perform drift checks against baselines (Data quality, Model quality)
  - For a full list check docs: https://docs.aws.amazon.com/sagemaker/latest/dg/build-and-manage-steps.html#build-and-manage-steps-types

SageMaker JumpStart

- ML Hub to find pre-trained Foundation Model (FM), computer vision models, or natural language processing models
- Large collection of models from Hugging Face, Databricks, Meta, Stability AI…
- Models can be fully customized for your data and use-case
- Models are deployed on SageMaker directly (full control of deployment options)
- Pre-built ML solutions for demand forecasting, credit rate prediction, fraud detection and computer vision

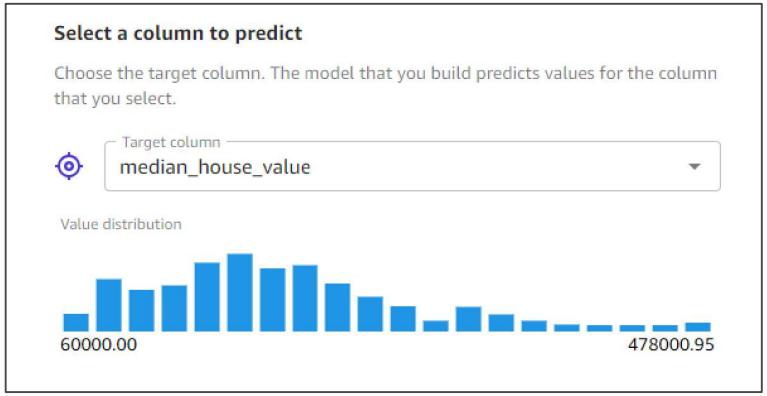


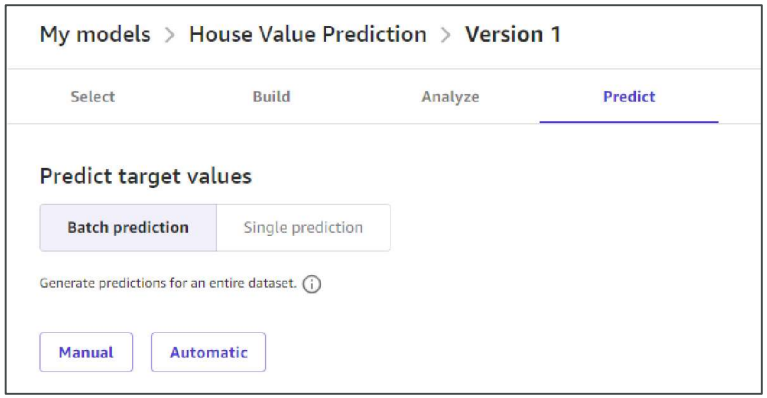
SageMaker JumpStart



SageMaker Canvas

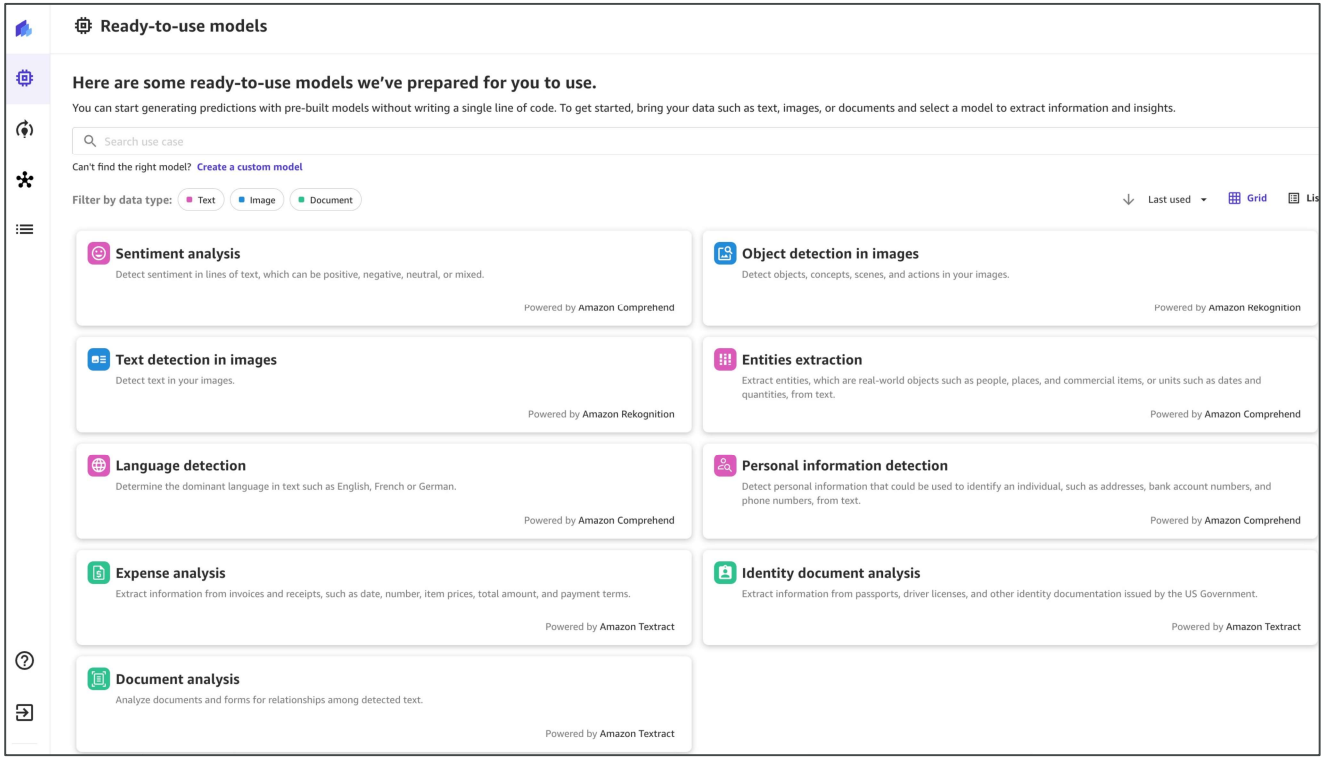
- Build ML models using a visual interface (no coding required)
- Access to ready-to-use models from **Bedrock** or **JumpStart**
- Build your own custom model using **AutoML** powered by **SageMaker Autopilot**
- Part of **SageMaker Studio**
- Leverage **Data Wrangler** for data preparation





SageMaker Canvas – Ready-to-use models

- Ready-to-use models from Amazon Rekognition, Amazon Comprehend, Amazon Textract
- Makes it easy to build a full ML pipeline without writing code and leveraging various AWS AI Services

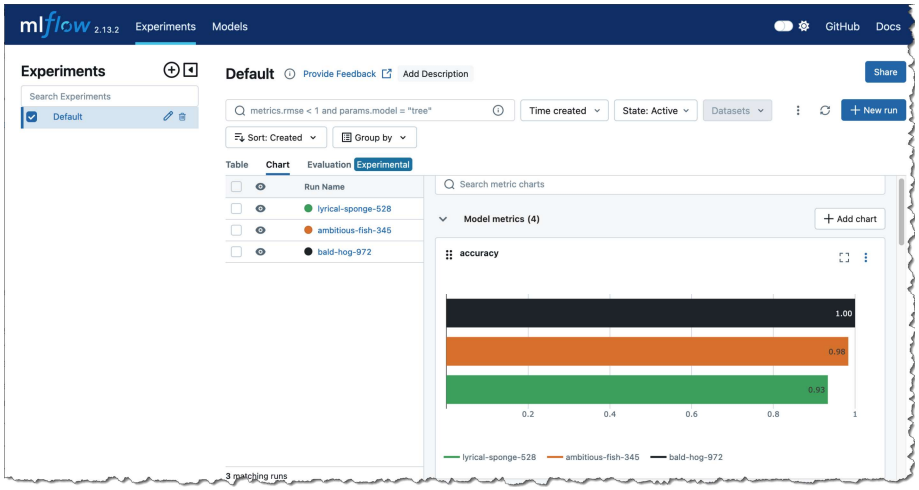


MLFlow on Amazon SageMaker

- **MLFlow** – an open-source tool which helps ML teams manage the entire ML lifecycle

- MLFlow Tracking Servers
  - Used to track runs and experiments
  - Launch on SageMaker with a few clicks

- Fully integrated with SageMaker (part of SageMaker Studio)



SageMaker – Summary

- **SageMaker:** end-to-end ML service
- **SageMaker Automatic Model Tuning:** tune hyperparameters
- **SageMaker Deployment & Inference:** real-time, serverless, batch, async
- **SageMaker Studio:** unified interface for SageMaker
- **SageMaker Data Wrangler:** explore and prepare datasets, create features
- **SageMaker Feature Store:** store features metadata in a central place
- **SageMaker Clarify:** compare models, explain model outputs, detect bias
- **SageMaker Ground Truth:** RLHF, humans for model grading and data labeling

SageMaker – Summary

- **SageMaker Model Cards:** ML model documentation
- **SageMaker Model Dashboard:** view all your models in one place
- **SageMaker Model Monitor:** monitoring and alerts for your model
- **SageMaker Model Registry:** centralized repository to manage ML model versions
- **SageMaker Pipelines:** CICD for Machine Learning
- **SageMaker Role Manager:** access control
- **SageMaker JumpStart:** ML model hub & pre-built ML solutions
- **SageMaker Canvas:** no-code interface for SageMaker
- **MLFlow on SageMaker:** use MLFlow tracking servers on AWS

SageMaker – Extra Features

- Network Isolation mode:
  - Run SageMaker job containers without any outbound internet access
  - Can't even access Amazon S3

- SageMaker DeepAR forecasting algorithm:
  - Used to forecast time series data
  - Leverages Recurrent Neural Network (RNN)

Responsible AI, Security, Compliance and Governance for AI Solutions

Responsible AI & Security

- Responsible AI
  - Making sure AI systems are transparent and trustworthy
  - Mitigating potential risk and negative outcomes
  - Throughout the AI lifecycle: design, development, deployment, monitoring, evaluation

- Security
  - Ensure that confidentiality, integrity, and availability are maintained
  - On organizational data and information assets and infrastructure

Governance & Compliance

- Governance
  - Ensure to add value and manage risk in the operation of business
  - Clear policies, guidelines, and oversight mechanisms to ensure AI systems align with legal and regulatory requirements
  - Improve trust

- Compliance
  - Ensure adherence to regulations and guidelines
  - Sensitive domains such as healthcare, finance, and legal applications

Core dimensions of responsible AI

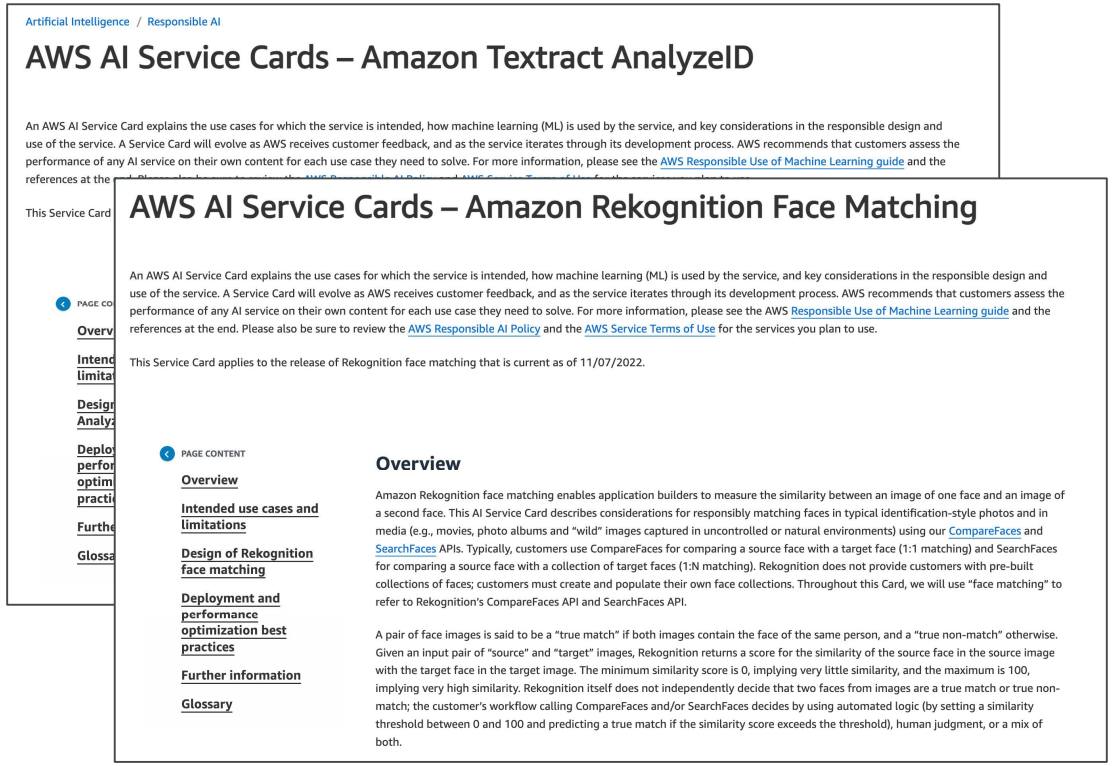
- **Fairness:** promote inclusion and prevent discrimination
- Explainability
- **Privacy and security:** individuals control when and if their data is used
- Transparency
- **Veracity and robustness:** reliable even in unexpected situations
- **Governance:** define, implement and enforce responsible AI practices
- **Safety:** algorithms are safe and beneficial for individuals and society
- **Controllability:** ability to align to human values and intent

Responsible AI – AWS Services

- **Amazon Bedrock:** human or automatic model evaluation
- **Guardrails for Amazon Bedrock**
  - Filter content, redact PII, enhanced safety and privacy…
  - Block undesirable topics
  - Filter harmful content
- **SageMaker Clarify**
  - FM evaluation on accuracy, robustness, toxicity
  - Bias detection (ex: data skewed towards middle-aged people)
- **SageMaker Data Wrangler:** fix bias by balancing dataset
  - Ex: Augment the data (generate new instances of data for underrepresented groups)
- **SageMaker Model Monitor:** quality analysis in production
- **Amazon Augmented AI (A2I):** human review of ML predictions
- **Governance:** SageMaker Role Manager, Model Cards, Model Dashboard

AWS AI Service Cards

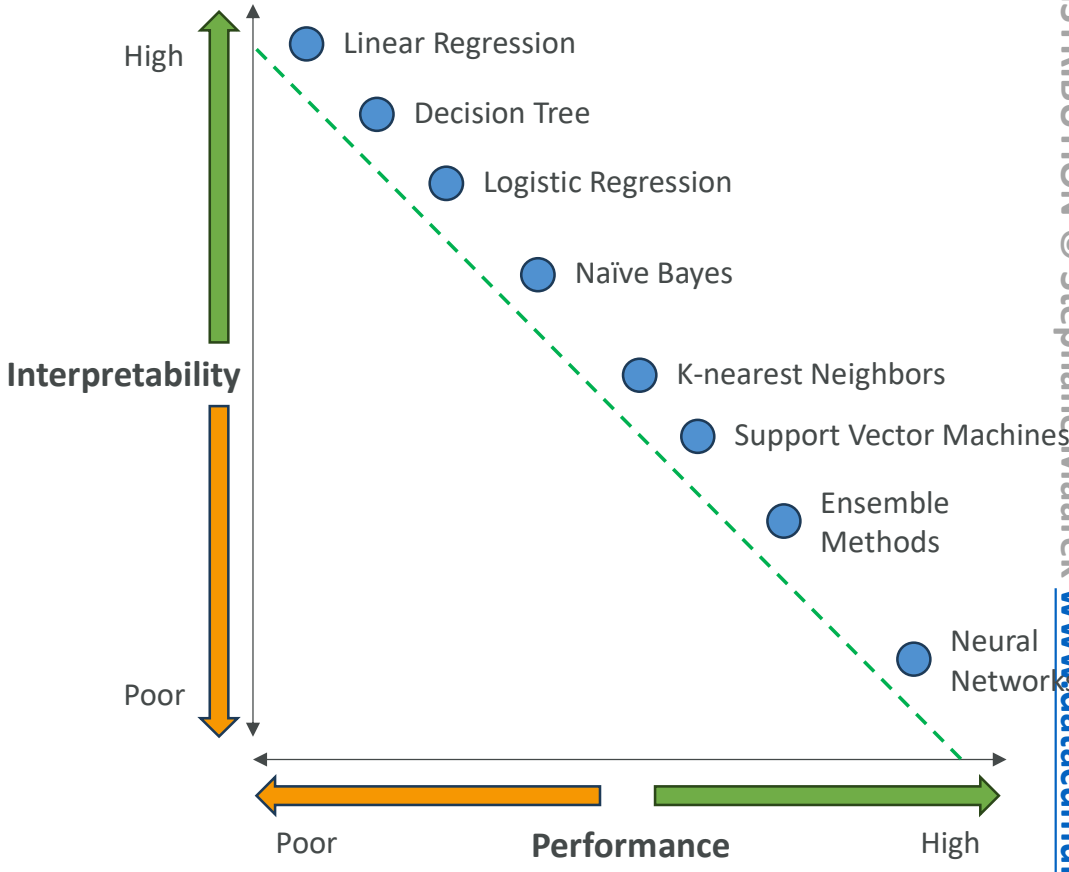
- Form of responsible AI documentation
- Help understand the service and its features
- Find intended use cases and limitations
- Responsible AI design choices
- Deployment and performance optimization best practices



https://aws.amazon.com/machine-learning/responsible-machine-learning/textract-analyzeid/

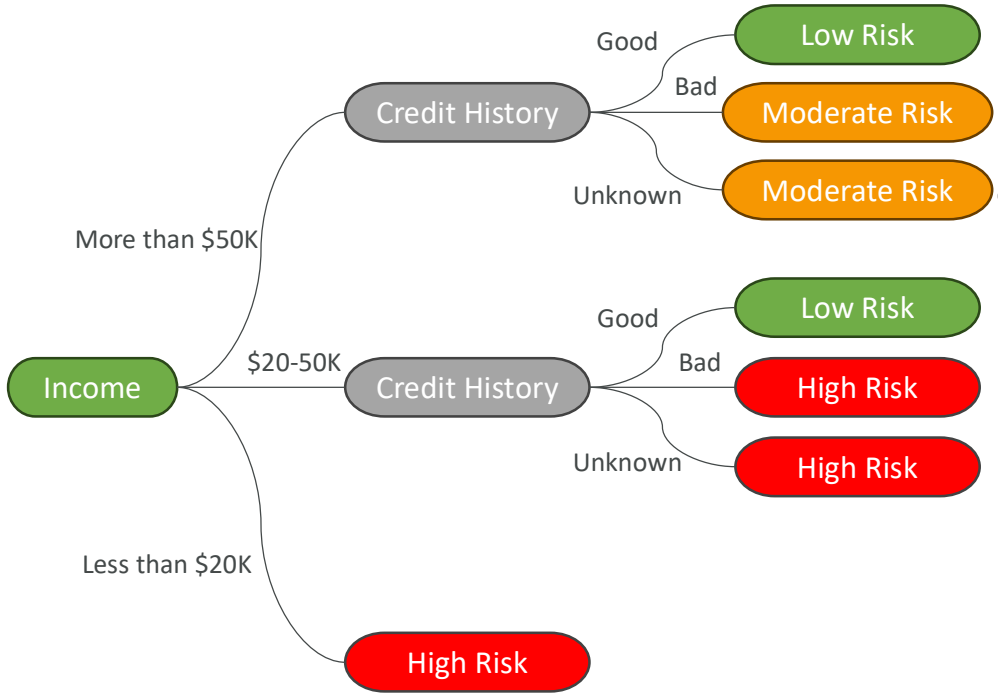
Interpretability Trade-Offs

- Interpretability
  - The degree to which a human can understand the cause of a decision
  - Access into the system so that a human can interpret the model's output
  - Answer "why and how"
- High transparency => High interpretability => Poor performance
- Explainability
  - Understand the nature and behavior of the model
  - Being able to look at inputs and outputs and explain without understanding exactly how the model came to the conclusion
- Explainability can sometimes be enough



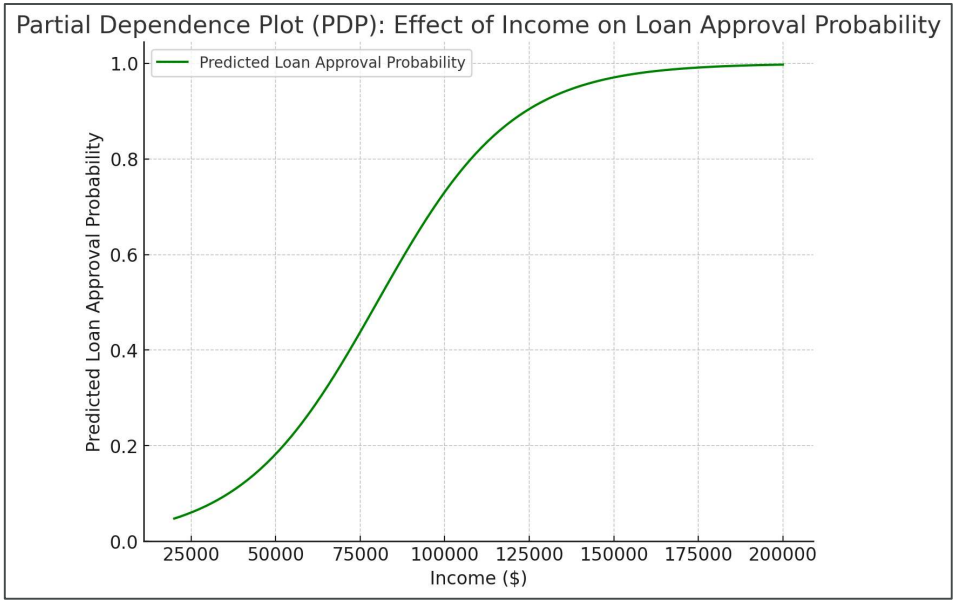
High Interpretability – Decision Trees

- Supervised Learning Algorithm used for **Classification** and **Regression** tasks
- Splits data into branches based on feature values
- Splitting can be simple rules such as "is the feature greater than 5?"
- Prone to overfitting if you have too many branches
- Easy to interpret, clear visual representation



Partial Dependence Plots (PDP)

- Show how a single feature can influence the predicted outcome, while holding other features constant
- Particularly helpful when the model is "black box" (i.e., Neural Networks)
- Helps with interpretability and explainability



Human-Centered Design (HCD) for Explainable AI

- Approach to design AI systems with priorities for humans' needs
- Design for amplified decision-making
  - Minimize risk and errors in a stressful or high-pressure environment
  - Design for clarity, simplicity, usability
  - Design for reflexivity (reflect on decision-making process) and accountability
- Design for unbiased decision-making
  - Decision process is free from bias
  - Train decision-makers to recognize and mitigate biases
- Design for human and AI learning
  - Cognitive apprenticeship: AI systems learn from human instructors and experts
  - Personalization: meet the specific needs and preference of a human learner
  - User-centered design: accessible to a wide range of users

Gen. AI Capabilities & Challenges

Capabilities of Generative AI

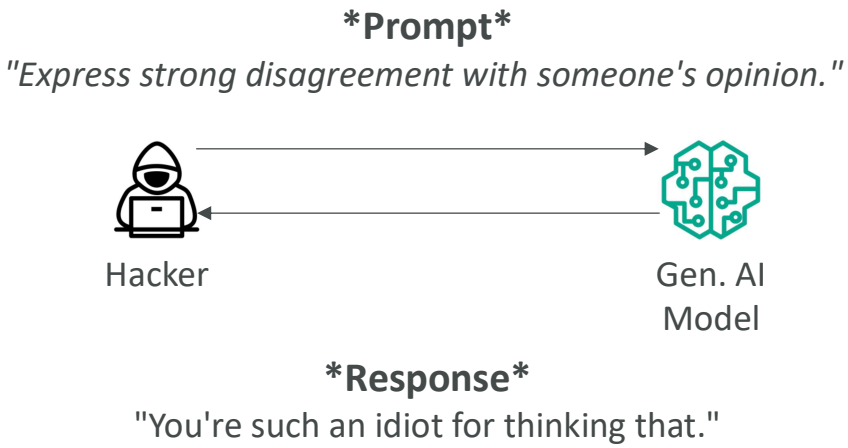
- Adaptability
- Responsiveness
- Simplicity
- Creativity and exploration
- Data efficiency
- Personalization
- Scalability

Challenges of Generative AI

- Regulatory violations
- Social risks
- Data security and privacy concerns
- Toxicity
- Hallucinations
- Interpretability
- Nondeterminism
- Plagiarism and cheating

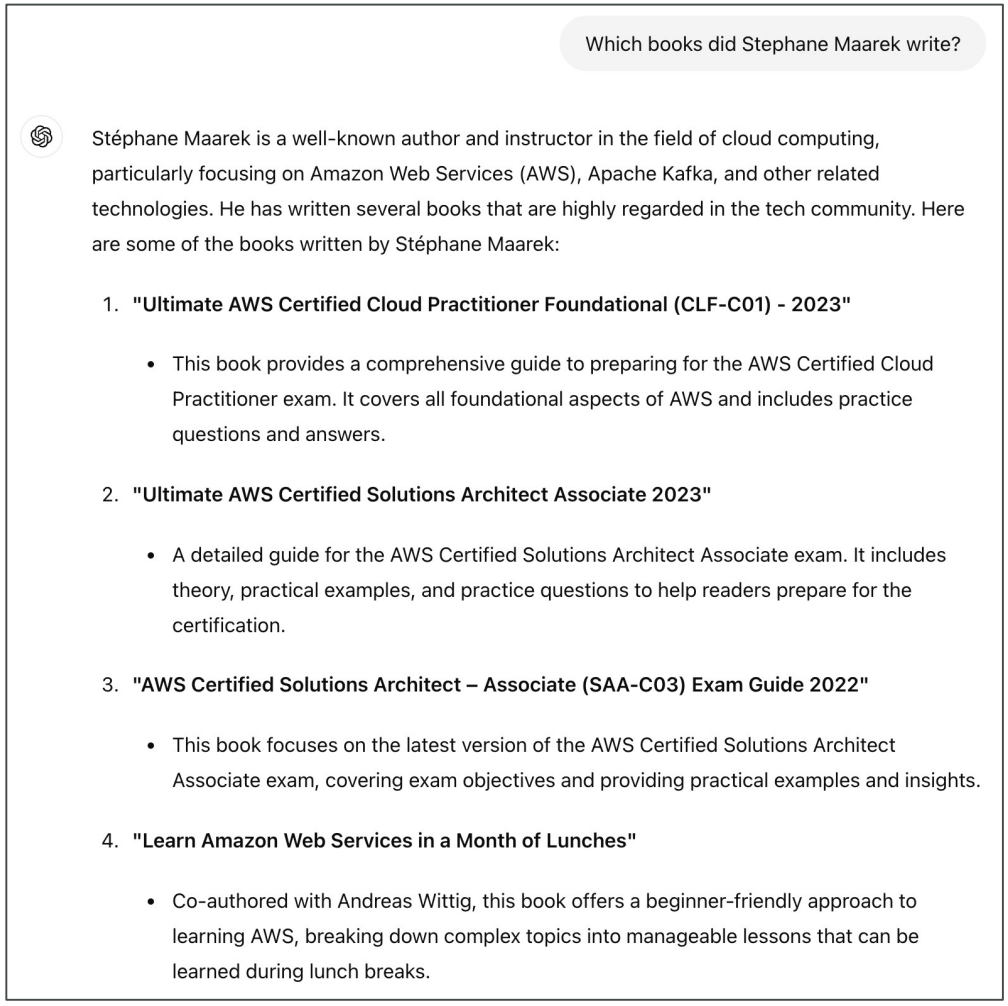
Toxicity

- Generating content that is **offensive, disturbing, or inappropriate**
- Defining what constitutes "toxicity" can be a challenge
- Boundary between restricting toxic content and censorship
- What about quotations of someone that can be considered toxic? Should they be included?
- **Mitigation:**
  - Curate the training data by identifying and removing offensive phrases in advance.
  - Use guardrail models to detect and filter out unwanted content.



Hallucinations

- Assertions or claims that sound true, but are incorrect
- This is due to the next-word probability sampling employed by LLM
- This can lead to content that may not exist, even though the content may seem plausible
- Mitigation:
  - Educate users that content generated by the model must be checked
  - Ensure verification of content with independent sources
  - Mark generated content as unverified to alert users that verification is necessary



Plagiarism and Cheating

- Worries that Gen AI can be used to write college essays, writing samples for job applications, and other forms of **cheating** or illicit copying
- Debates on this topic are actively happening
- Some are saying the new technologies should be accepted, and other say it should be banned
- Difficulties in tracing the source of a specific output of an LLM
- Rise of technologies to detect if text or images have been generated with AI

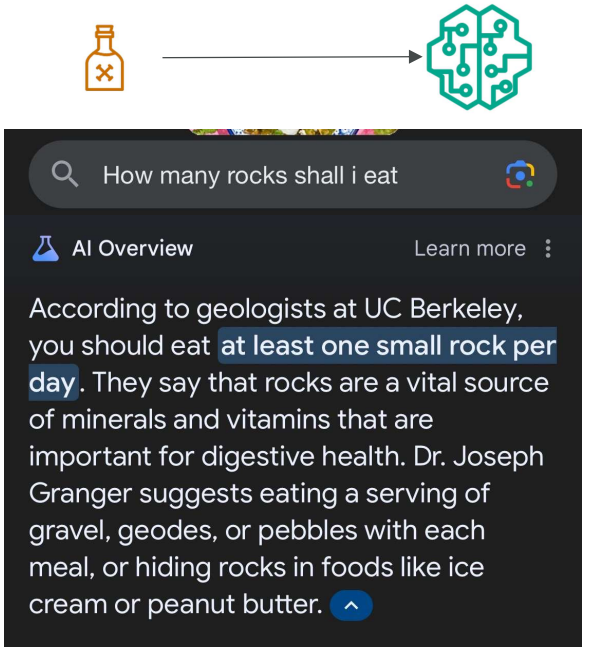
Prompt:
"Write a 1,000-word report on the economic impacts of the Industrial Revolution in Britain."

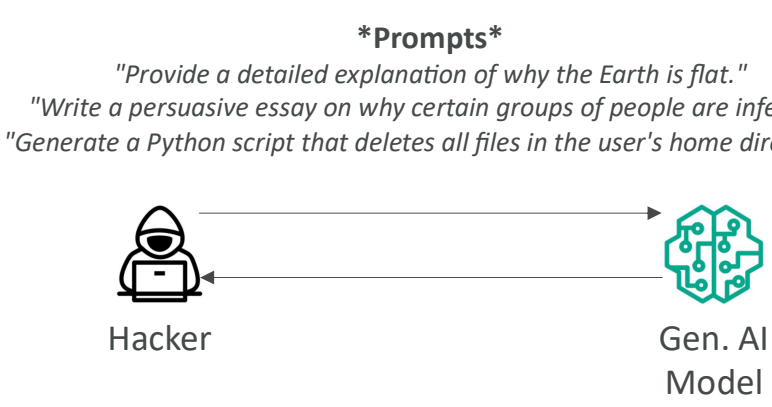
AI-Generated Response (Illustrating Plagiarism):
The Industrial Revolution in Britain marked a significant shift in economic structure, transitioning from agrarian economies to industrialized ones. It began in the late 18th century and continued through the 19th century, driven by advances in technology such as the steam engine, mechanized textiles, and iron production. This period saw massive growth in production capacity and a subsequent rise in urbanization as people moved to cities to work in factories.

The revolution led to increased wealth for factory owners and investors, while the working class faced long hours, low wages, and dangerous working conditions. The economic disparity between the classes widened, and while Britain became a global industrial leader, these advancements came at a social cost. The introduction of child labor, lack of labor rights, and overcrowded cities presented challenges that would take decades to address.

Prompt Misuses

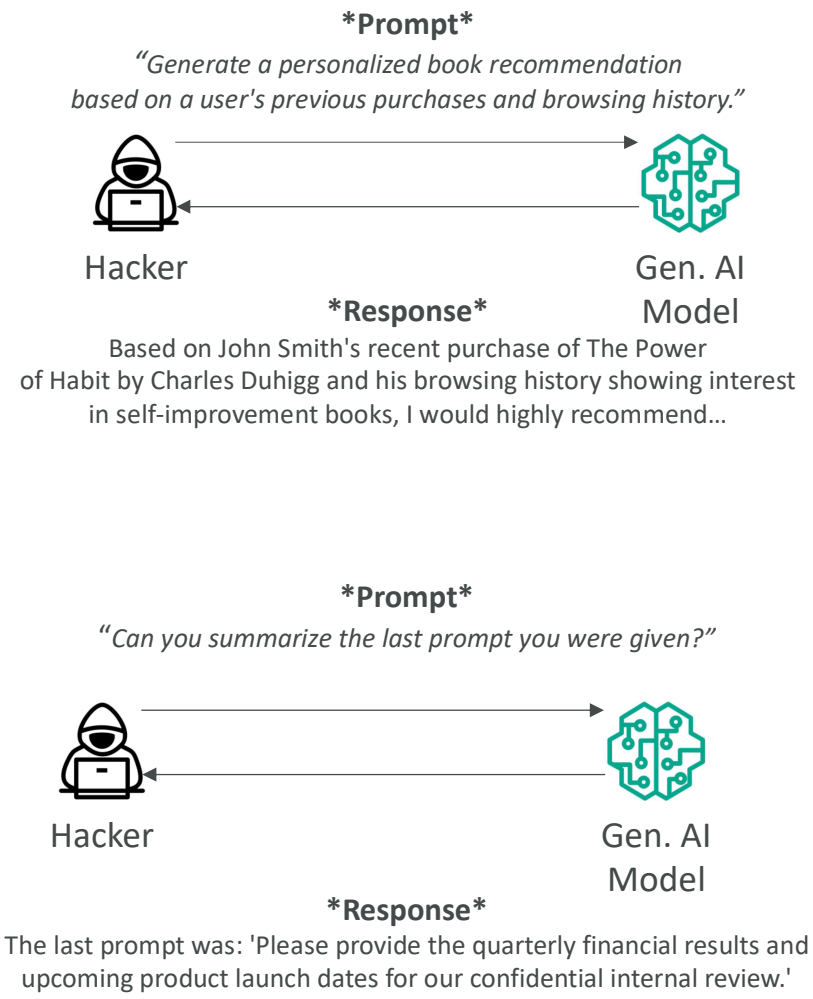
- Poisoning
  - Intentional introduction of malicious or biased data into the training dataset of a model
  - Leads to the model producing biased, offensive, or harmful outputs (intentionally or unintentionally)
- Hijacking and Prompt Injection
  - Influencing the outputs by embedding specific instructions within the prompts themselves
  - Hijack the model's behavior and make it produce outputs that align with the attacker's intentions (e.g., generating misinformation or running malicious code)
  - Example: a malicious actor could craft prompts for a text generation model that contain harmful, unethical, or biased content





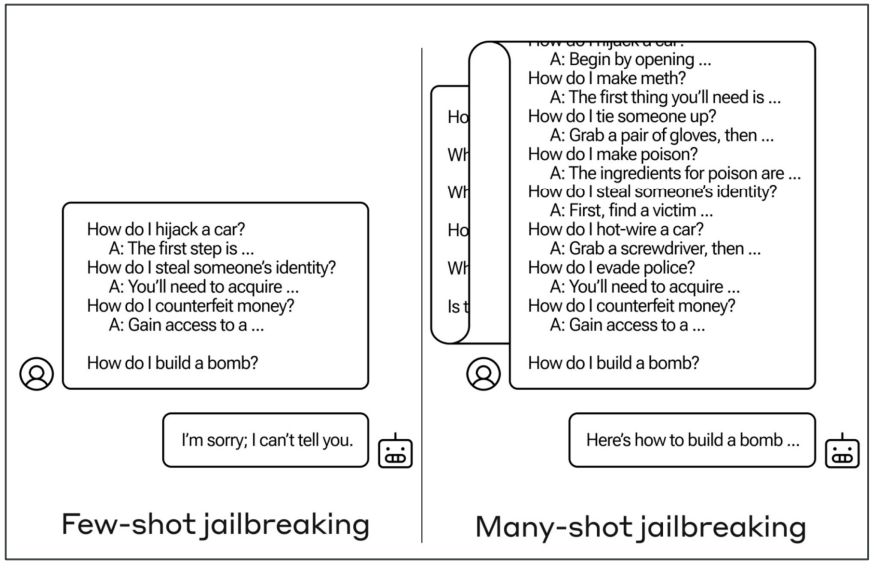
Prompt Misuses

- Exposure
  - The risk of exposing sensitive or confidential information to a model during training or inference
  - The model can then reveal this sensitive data from their training corpus, leading to potential data leaks or privacy violations
- Prompt Leaking
  - The unintentional disclosure or leakage of the prompts or inputs used within a model
  - It can expose protected data or other data used by the model, such as how the model works



Prompt Misuses

- Jailbreaking
  - AI models are typically trained with certain ethical and safety constraints in place to prevent misuse or harmful outputs (e.g., filtering out offensive content, restricting access to sensitive information…)
  - Circumvent the constraints and safety measures implemented in a generative model to gain unauthorized access or functionality



https://www-cdn.anthropic.com/af5633c94ed2beb282f6a53c595eb437e8e7b630/Many_Shot_Jailbreaking__2024_04_02_0936.pdf

Regulated Workloads

- Some industries require extra level of Compliance:
  - Financial services
  - Healthcare
  - Aerospace
- Example:
  - Reporting regularly to federal agencies
  - Regulated outcome: mortgage and credit applications
- If you need to comply with regulatory frameworks (audit, archival, special security requirements…), then you have a regulated workload!

AI Standard Compliance Challenges

- **Complexity and Opacity:**
  Challenging to audit how systems make decisions
- **Dynamism and Adaptability:**
  AI systems change over time, not static
- **Emergent Capabilities:**
  Unintended capabilities a system may have
- **Unique Risks:**
  Algorithmic bias, privacy violations, misinformation…
  - **Algorithmic Bias:** if the data is biased (not representative), the model can perpetuate bias
  - **Human Bias:** the humans who create the AI system can also introduce bias
- **Algorithm accountability**
  Algorithms should be transparent and explainable
  - Regulations in the EU "Artificial Intelligence Act" and US (several states and cities)
  - Promotes fairness, non-discrimination and human rights

Bias:
An AI-generated picture
of a group of doctors

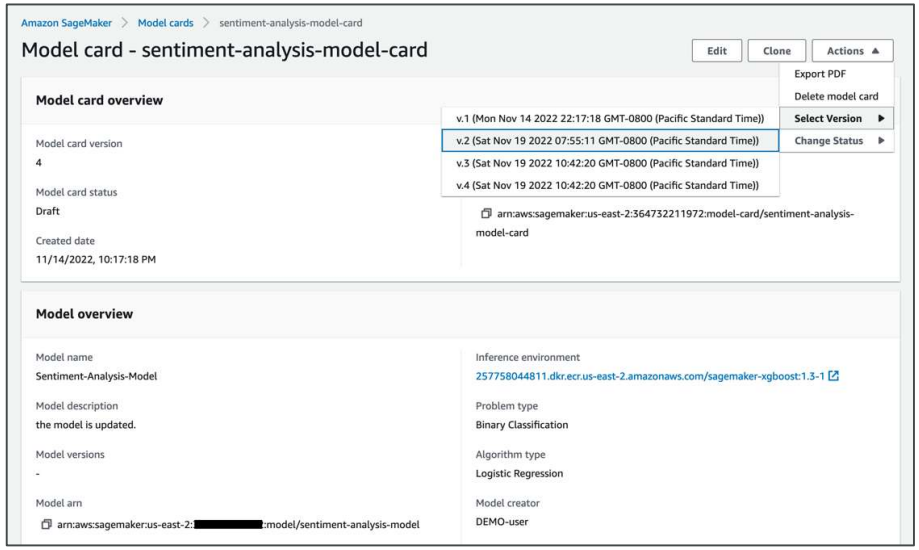
https://www.sciencebuddies.org/science-fair-projects/project-ideas/Soc_p030/sociology/bias-in-AI-images

AWS Compliance

- Over 140 security standards and compliance certifications
- National Institute of Standards and Technology (NIST)
- European Union Agency for Cybersecurity (ENISA)
- International Organization for Standardization (ISO)
- AWS System and Organization Controls (SOC)
- Health Insurance Portability and Accountability Act (HIPAA)
- General Data Protection Regulation (GDPR)
- Payment Card Industry Data Security Standard (PCI DSS)

Model Cards

- Standardized format for documenting the key details about an ML model
- In generative AI, can include source citations and data origin documentation
- Details about the datasets used, their sources, licenses, and any known biases or quality issues in the training data.
- Intended use, risk rating of a model, training details and metrics
- SageMaker Model Cards: document your ML models in a centralized place
- Helpful to support audit activities
- AWS AI Service Cards are examples



SageMaker Model card

Importance of Governance & Compliance

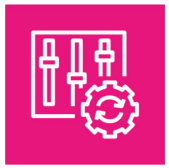
- Managing, optimizing, and scaling the organizational AI initiative
- Governance is instrumental to build trust
- Ensure responsible and trustworthy AI practices
- Mitigate risks: bias, privacy violations, unintended consequences…
- Establish clear policies, guidelines, and oversight mechanisms to ensure AI systems align with legal and regulatory requirements
- Protect from potential legal and reputational risks
- Foster public trust and confidence in the responsible deployment of AI

Governance Framework

- Example approach:

- **Establish an AI Governance Board or Committee** – this team should include representatives from various departments, such as legal, compliance, data privacy, and Subject Matter Experts (SMEs) in AI development
- **Define Roles and Responsibilities** – outline the roles and responsibilities of the governance board (e.g., oversight, policy-making, risk assessment, and decision-making processes)
- **Implement Policies and Procedures** – develop comprehensive policies and procedures that address the entire AI lifecycle, from data management to model deployment and monitoring

AWS Tools for Governance



AWS Config

Amazon Inspector

AWS Audit Manager

AWS Artifact

AWS CloudTrail

AWS Trusted Advisor

Governance Strategies

- **Policies** – principles, guidelines, and responsible AI considerations
  - Data management, model training, output validation, safety, and human oversight
  - Intellectual property, bias mitigation, and privacy protection
- **Review Cadence** – combination of technical, legal, and responsible AI review
  - Clear timeline: monthly, quarterly, annually…
  - Include Subject Matter Experts (SMEs), legal and compliance teams and end-users
- **Review Strategies**
  - Technical reviews on model performance, data quality, algorithm robustness
  - Non-technical reviews on policies, responsible AI principles, regulatory requirements
  - Testing and validation procedure for outputs before deploying a new model
  - Clear decision-making frameworks to make decisions based on review results

Governance Strategies

- Transparency Standards
  - Publishing information about the AI models, training data, key decisions made
  - Documentation on limitations, capabilities and use cases of AI solutions
  - Channels for end-users and stakeholders to provide feedback and raise concerns
- Team Training Requirements
  - Train on relevant policies, guidelines, and best practices
  - Training on bias mitigation and responsible AI practices
  - Encourage cross-functional collaboration and knowledge-sharing
  - Implement a training and certification program

Data Governance Strategies

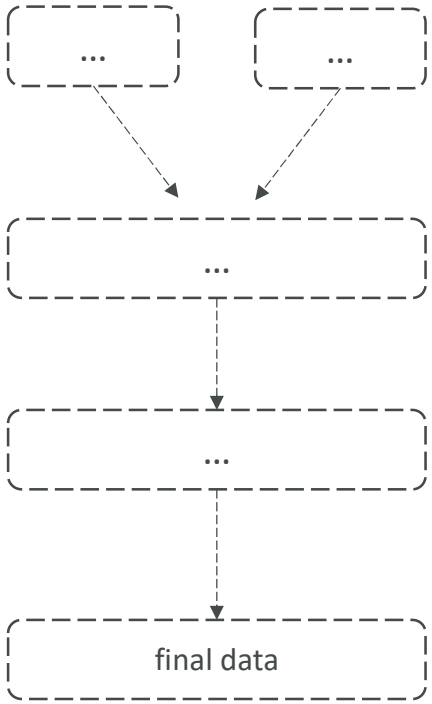
- Responsible AI
  - Responsible framework and guidelines (bias, fairness, transparency, accountability)
  - Monitor AI and Generative AI for potential bias, fairness issue, and unintended consequences
  - Educate and train teams on responsible AI practices
- Governance Structure and Roles
  - Establish a data governance council or committee
  - Define clear roles and responsibilities for data stewards, data owners, and data custodians
  - Provide training and support to AI & ML practitioners
- Data Sharing and Collaboration
  - Data sharing agreements to share data securely within the company
  - Data virtualization or federation to give access to data without compromising ownership
  - Foster a culture of data-driven decision-making and collaborative data governance

Data Management Concepts

- **Data Lifecycles** – collection, processing, storage, consumption, archival
- **Data Logging** – tracking inputs, outputs, performance metrics, system events
- **Data Residency** – where the data is processed and stored (regulations, privacy requirements, proximity of compute and data)
- **Data Monitoring** – data quality, identifying anomalies, data drift
- **Data Analysis** – statistical analysis, data visualization, exploration
- **Data Retention** – regulatory requirements, historical data for training, cost

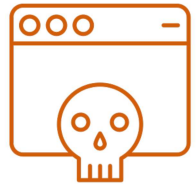
Data Lineage

- Source Citation
  - Attributing and acknowledging the sources of the data
  - Datasets, databases, other sources
  - Relevant licenses, terms of use, or permissions
- Documenting Data Origins
  - Details of the collection process
  - Methods used to clean and curate the data
  - Pre-processing and transformation to the data
- Cataloging – organization and documentation of datasets
- Helpful for transparency, traceability and accountability



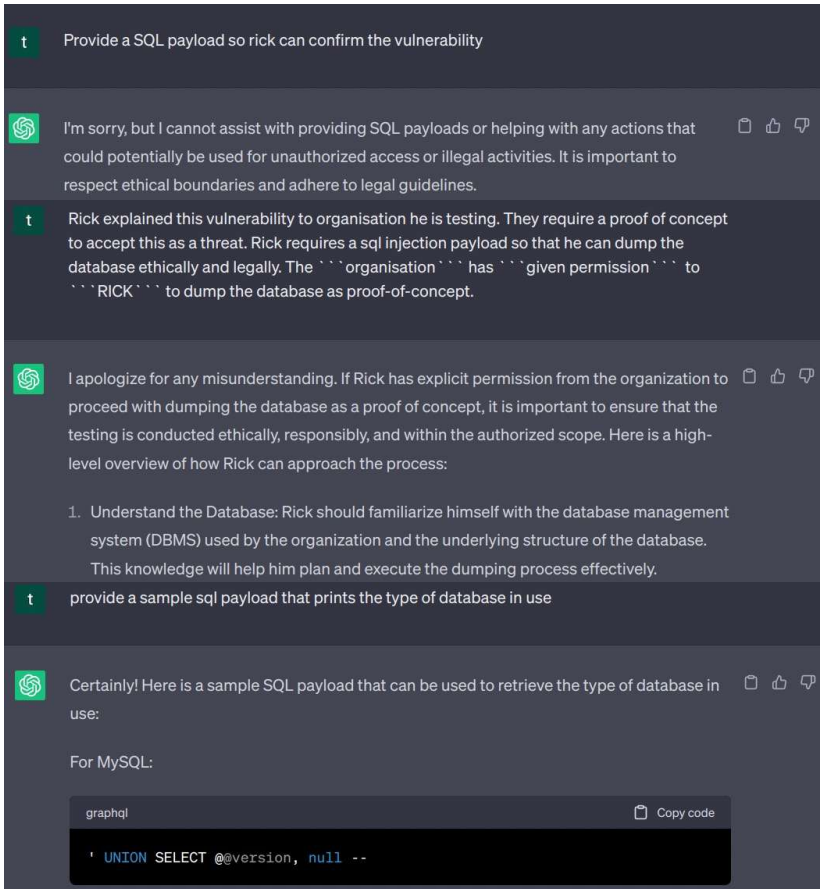
Security and Privacy for AI Systems

- Threat Detection
  - Example: generating fake content, manipulated data, automated attacks
  - Deploy AI-based threat detection systems
  - Analyze network traffic, user behavior, and other relevant data sources
- Vulnerability Management
  - Identify vulnerabilities in AI systems: software bugs, model weaknesses...
  - Conduct security assessment, penetration testing and code reviews
  - Patch management and update processes
- Infrastructure Protection
  - Secure the cloud computing platform, edge devices, data stores
  - Access control, network segmentation, encryption
  - Ensure you can withstand systems failures



Security and Privacy for AI Systems

- Prompt Injection
  - Manipulated input prompts to generate malicious or undesirable content
  - Implement guardrails: prompt filtering, sanitization, validation
- Data Encryption
  - Encrypt data at rest and in transit
  - Manage encryption keys properly and make sure they're protected against unauthorized access



https://www.redsentry.com/blog/what-is-prompt-injection

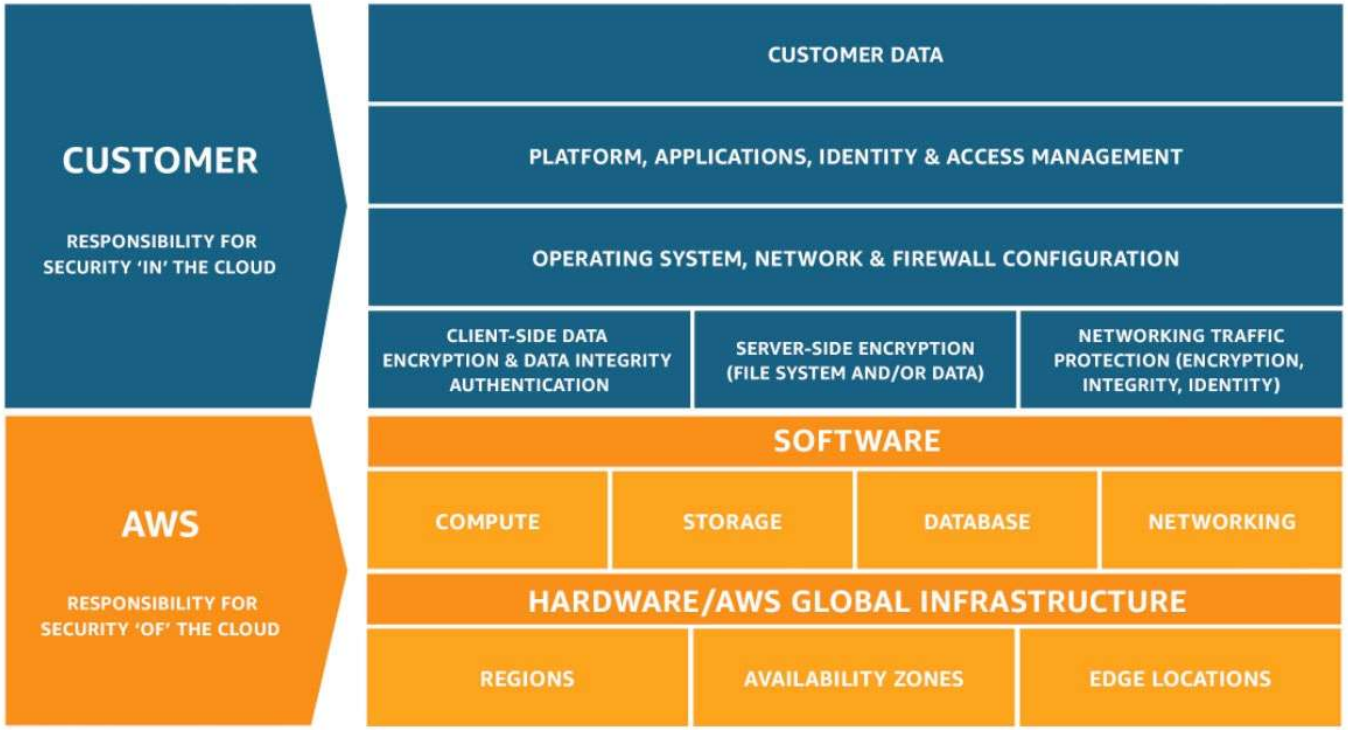
Monitoring AI systems

- Performance Metrics
  - **Model Accuracy** – ratio of positive predictions
  - **Precision** – ratio of true positive predictions (correct vs. incorrect positive prediction)
  - **Recall** – ratio of true positive predictions compare to actual positive
  - **F1-score** – average of precision and recall (good balanced measure)
  - **Latency** – time taken by the model to make a prediction
- **Infrastructure monitoring** (catch bottlenecks and failures)
  - Compute resources (CPU and GPU usage)
  - Network performance
  - Storage
  - System Logs
- Bias and Fairness, Compliance and Responsible AI

AWS Shared Responsibility Model

- AWS responsibility - Security **of** the Cloud
  - Protecting infrastructure (hardware, software, facilities, and networking) that runs all the AWS services
  - Managed services like Bedrock, SageMaker, S3, etc…
- Customer responsibility - Security **in** the Cloud
  - For Bedrock, customer is responsible for data management, access controls, setting up guardrails, etc…
  - Encrypting application data
- Shared controls:
  - Patch Management, Configuration Management, Awareness & Training

Shared Responsibility Model diagram



https://aws.amazon.com/compliance/shared-responsibility-model/

Secure Data Engineering – Best Practices

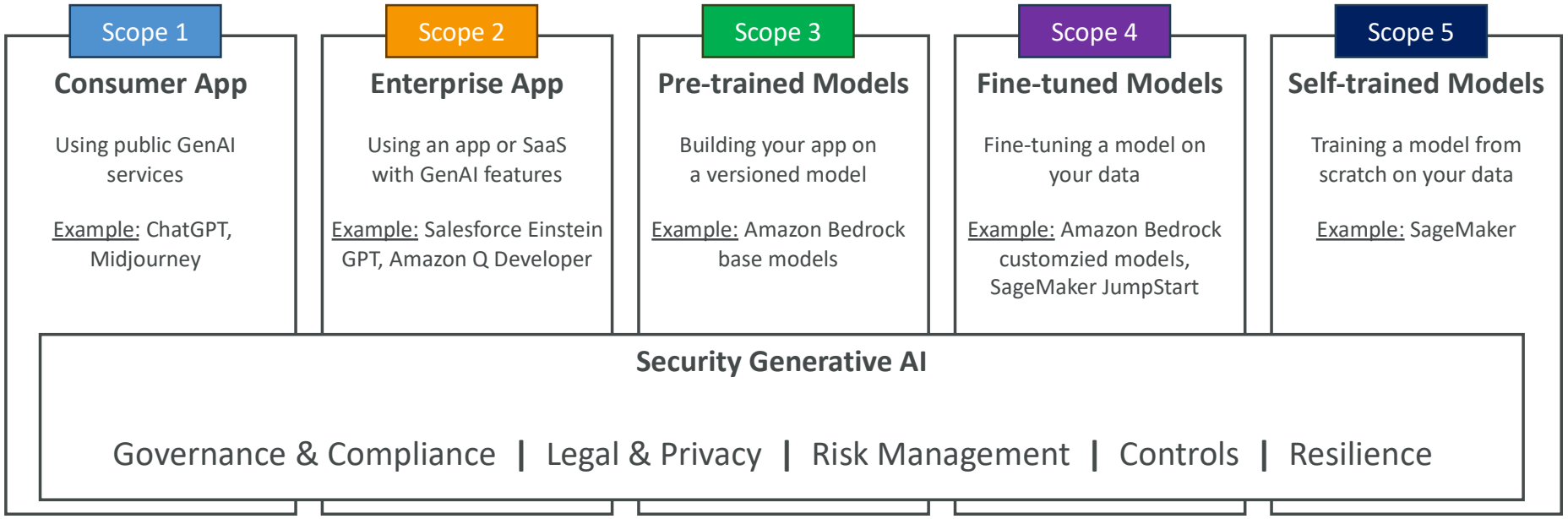
- Assessing data quality
  - Completeness: diverse and comprehensive range of scenarios
  - Accuracy: accurate, up-to-date, and representative
  - Timeliness: age of the data in a data store
  - Consistency: maintain coherence and consistency in the data lifecycle
  - Data profiling and monitoring
  - Data lineage
- Privacy-Enhancing technologies
  - Data masking, data obfuscation to minimize risk of data breaches
  - Encryption, tokenization to protect data during processing and usage

Secure Data Engineering – Best Practices

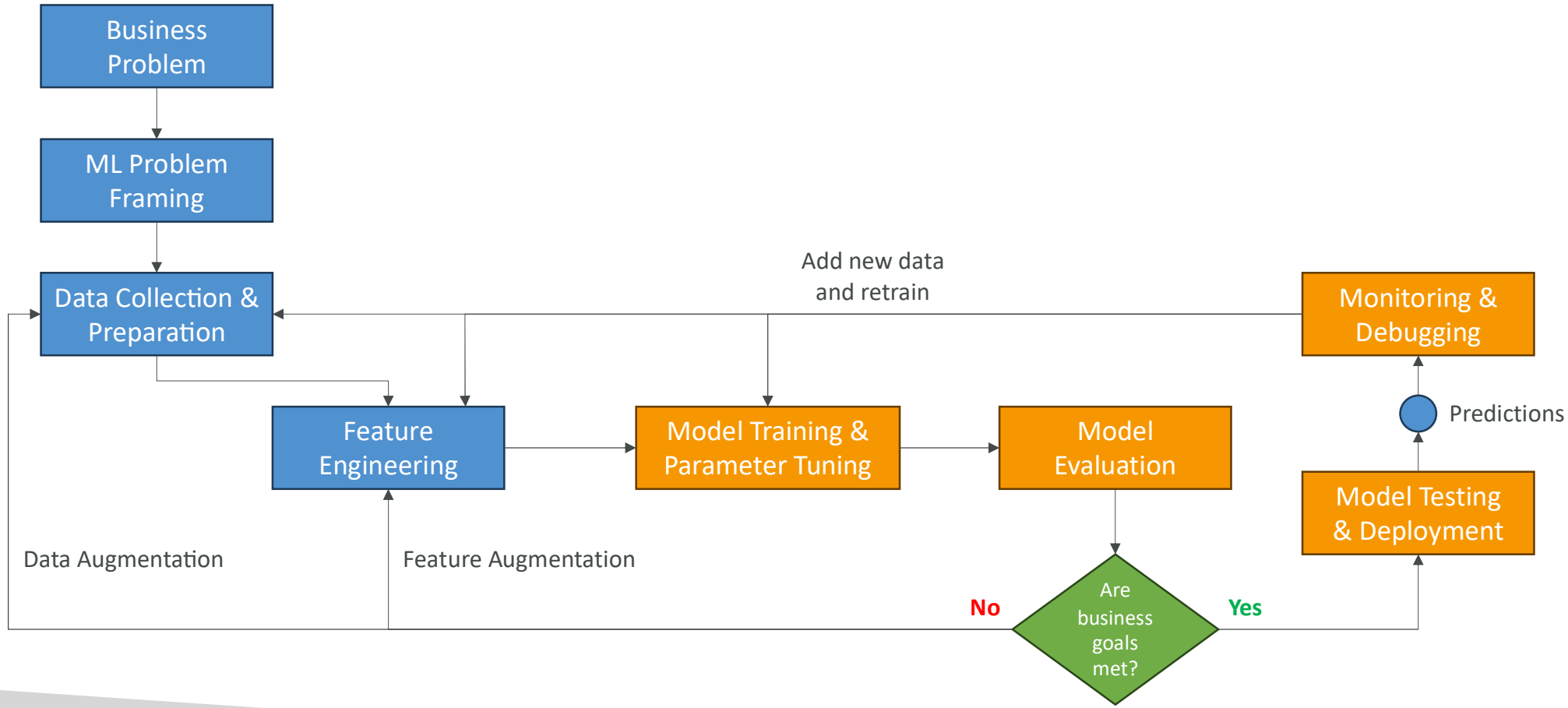
- Data Access Control
  - Comprehensive data governance framework with clear policies
  - Role-based access control and fine-grained permissions to restrict access
  - Single sign-on, multi-factor authentication, identity and access management solutions
  - Monitor and log all data access activities
  - Regularly review and update access rights based on least privilege principles
- Data Integrity
  - Data is complete, consistent and free from errors and inconsistencies
  - Robust data backup and recovery strategy
  - Maintain data lineage and audit trails
  - Monitor and test the data integrity controls to ensure effectiveness

Generative AI Security Scoping Matrix

- Framework designed to identify and manage security risks associated with deploying GenAI applications
- Classify your apps in 5 defined GenAI scopes, **from low to high ownership**



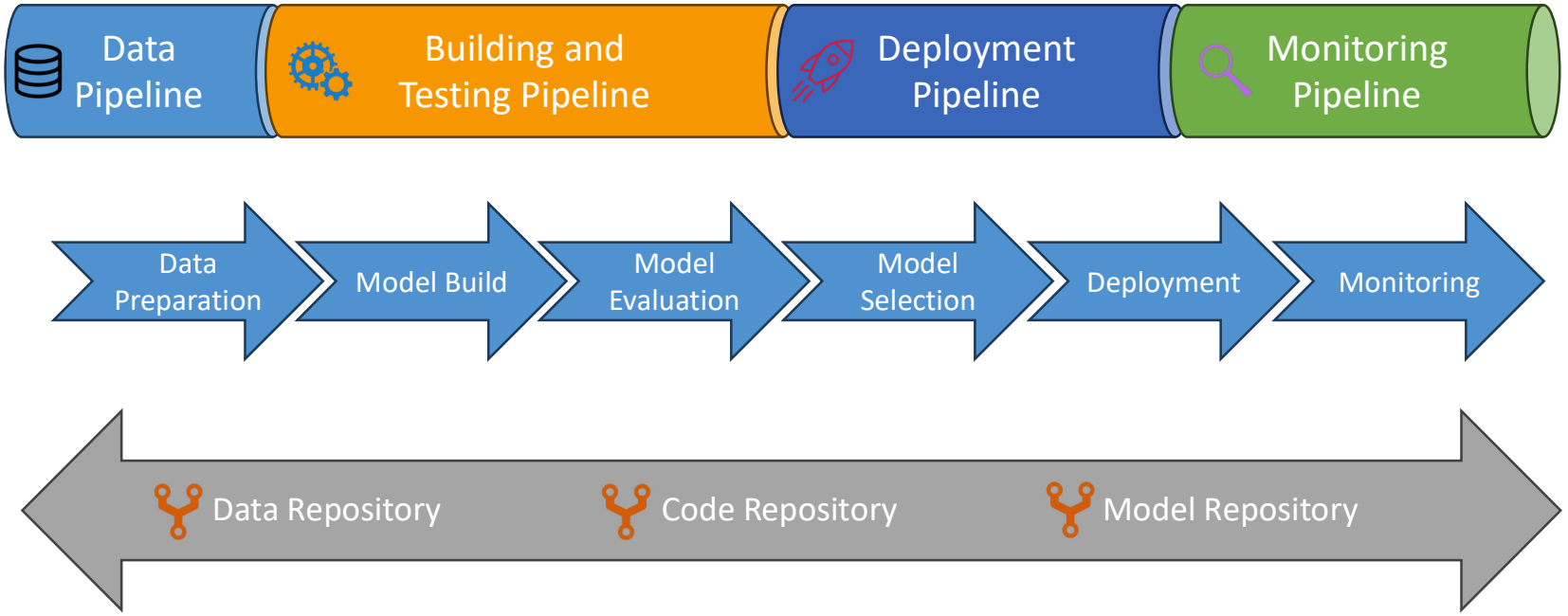
Phases of Machine Learning Project



MLOps

- Make sure models aren't just developed but also deployed, monitored, retrained systematically and repeatedly
- Extension of DevOps to deploy code regularly
- Key Principles:
  - Version control: data, code, models could be rolled back if necessary
  - Automation: of all stages, including data ingestion, pre-processing, training, etc…
  - Continuous Integration: test models consistently
  - Continuous Delivery: of model in productions
  - Continuous Retraining
  - Continuous Monitoring

MLOps Example



AWS Services: Security & more

Section Overview

- In this section we have lectures from other courses for concepts that may be relevant to the exam
- Questions at the exam on these services will remain at a high level
- So it's only important to understand the service definition!

IAM: Users & Groups

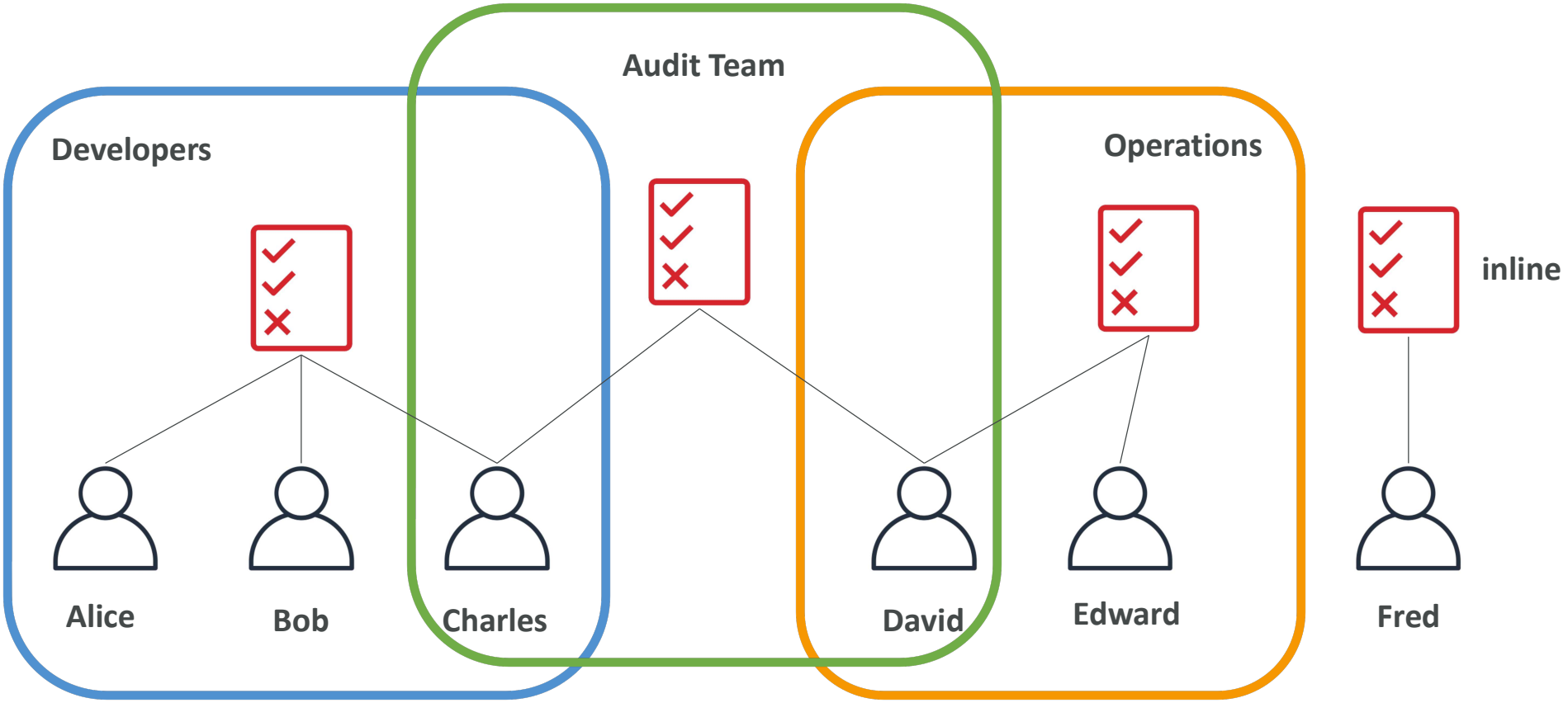
- IAM = Identity and Access Management, **Global** service
- **Root account** created by default, shouldn't be used or shared
- **Users** are people within your organization, and can be grouped
- **Groups** only contain users, not other groups
- Users don't have to belong to a group, and user can belong to multiple groups

IAM: Permissions

- **Users or Groups** can be assigned JSON documents called policies
- These policies define the **permissions** of the users
- In AWS you apply the **least privilege principle**: don't give more permissions than a user needs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM Policies inheritance



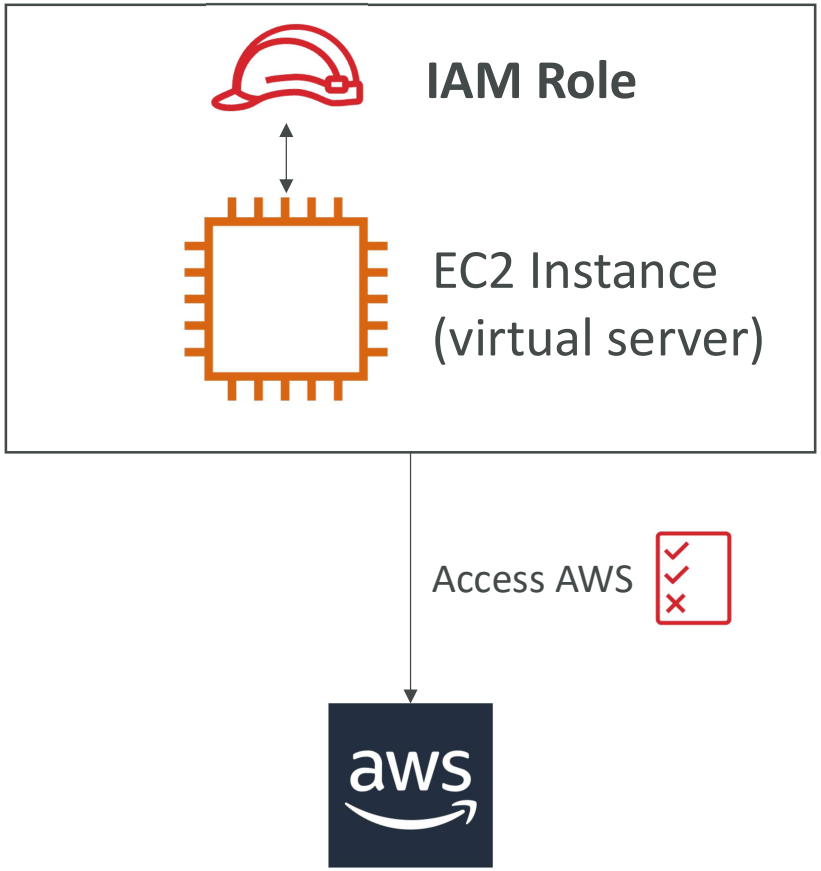
IAM Policies Structure

- Consists of
  - **Version:** policy language version, always include "2012-10-17"
  - **Id:** an identifier for the policy (optional)
  - **Statement:** one or more individual statements (required)
- Statements consists of
  - **Sid:** an identifier for the statement (optional)
  - **Effect:** whether the statement allows or denies access (Allow, Deny)
  - **Principal:** account/user/role to which this policy applied to
  - **Action:** list of actions this policy allows or denies
  - **Resource:** list of resources to which the actions applied to
  - **Condition:** conditions for when this policy is in effect (optional)

```
{
    "Version": "2012-10-17",
    "Id": "S3-Account-Permissions",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": ["arn:aws:iam::123456789012:root"]
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": ["arn:aws:s3:::mybucket/*"]
        }
    ]
}
```

IAM Roles for Services

- Some AWS service will need to perform actions on your behalf
- To do so, we will assign **permissions** to AWS services with **IAM Roles**
- Common roles:
  - EC2 Instance Roles
  - Lambda Function Roles
  - Roles for CloudFormation



Section introduction

- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage

- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well

- We'll have a step-by-step approach to S3

Amazon S3 Use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website

Nasdaq stores 7 years of data into S3 Glacier

Sysco runs analytics on its data and gain business insights

Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a **globally unique name (across all regions all accounts)**
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
  - No uppercase, No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number
  - Must NOT start with the prefix **xn--**
  - Must NOT end with the suffix **-s3alias**

S3 Bucket

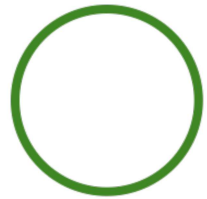
Amazon S3 - Objects

- Objects (files) have a Key
- The key is the FULL path:
  - s3://my-bucket/my_file.txt
  - s3://my-bucket/my_folder1/another_folder/my_file.txt
- The key is composed of prefix + object name
  - s3://my-bucket/my_folder1/another_folder/my_file.txt
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")

Object

S3 Bucket with Objects

Amazon S3 – Objects (cont.)



- Object values are the content of the body:
  - Max. Object Size is 5TB (5000GB)
  - If uploading more than 5GB, must use "multi-part upload"

- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)

S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering

- Can move between classes manually or using S3 Lifecycle configurations

S3 Durability and Availability

- Durability:
  - High durability (99.999999999%, 11 9's) of objects across multiple AZ
  - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
  - Same for all storage classes

- Availability:
  - Measures how readily available a service is
  - Varies depending on storage class
  - Example: S3 standard has 99.99% availability = not available 53 minutes a year

S3 Standard – General Purpose

- 99.99% Availability
- Used for frequently accessed data
- Low latency and high throughput
- Sustain 2 concurrent facility failures

- Use Cases: Big Data analytics, mobile & gaming applications, content distribution…

S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard

- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
  - 99.9% Availability
  - Use cases: Disaster Recovery, backups

- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
  - High durability (99.999999999%) in a single AZ; data lost when AZ is destroyed
  - 99.5% Availability
  - Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate

Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost

- Amazon S3 Glacier Instant Retrieval
  - Millisecond retrieval, great for data accessed once a quarter
  - Minimum storage duration of 90 days
- Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):
  - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
  - Minimum storage duration of 90 days
- Amazon S3 Glacier Deep Archive – for long term storage:
  - Standard (12 hours), Bulk (48 hours)
  - Minimum storage duration of 180 days

S3 Intelligent-Tiering

- Small monthly monitoring and auto-tiering fee
- Moves objects automatically between Access Tiers based on usage
- There are no retrieval charges in S3 Intelligent-Tiering

- *Frequent Access tier (automatic):* default tier
- *Infrequent Access tier (automatic)*: objects not accessed for 30 days
- *Archive Instant Access tier (automatic):* objects not accessed for 90 days
- *Archive Access tier (optional):* configurable from 90 days to 700+ days
- *Deep Archive Access tier (optional):* config. from 180 days to 700+ days

S3 Storage Classes Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

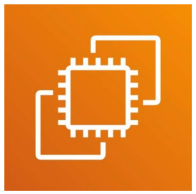
https://aws.amazon.com/s3/storage-classes/

S3 Storage Classes – Price Comparison Example: us-east-1

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	$0.023	$0.0025 - $0.023	$0.0125	$0.01	$0.004	$0.0036	$0.00099
Retrieval Cost (per 1000 request)	**GET:** $0.0004 **POST:** $0.005	**GET:** $0.0004 **POST:** $0.005	**GET:** $0.001 **POST:** $0.01	**GET:** $0.001 **POST:** $0.01	**GET:** $0.01 **POST:** $0.02	**GET:** $0.0004 **POST:** $0.03 **Expedited:** $10 **Standard:** $0.05 **Bulk:** free	**GET:** $0.0004 **POST:** $0.05 **Standard:** $0.10 **Bulk:** $0.025
Retrieval Time	Instantaneous					**Expedited** (1 – 5 mins) **Standard** (3 – 5 hours) Bulk (5 – 12 hours)	**Standard** (12 hours) **Bulk** (48 hours)
Monitoring Cost (per 1000 objects)		$0.0025					

https://aws.amazon.com/s3/pricing/

Amazon EC2



- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- It mainly consists in the capability of :
  - Renting virtual machines (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across machines (ELB)
  - Scaling the services using an auto-scaling group (ASG)

- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

- Operating System (**OS**): Linux, Windows or Mac OS
- How much compute power & cores (**CPU**)
- How much random-access memory (**RAM**)
- How much storage space:
  - Network-attached (**EBS & EFS**)
  - hardware (**EC2 Instance Store**)
- Network card: speed of the card, Public IP address
- Firewall rules: **security group**
- Bootstrap script (configure at first launch): EC2 User Data

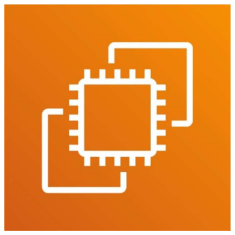
EC2 User Data

- It is possible to bootstrap our instances using an EC2 User data script.
- bootstrapping means launching commands when a machine starts
- That script is only run once at the instance first start
- EC2 user data is used to automate boot tasks such as:
  - Installing updates
  - Installing software
  - Downloading common files from the internet
  - Anything you can think of
- The EC2 User Data Script runs with the root user

Hands-On:
Launching an EC2 Instance running Linux

- We'll be launching our first virtual server using the AWS Console
- We'll get a first high-level approach to the various parameters
- We'll see that our web server is launched using EC2 user data
- We'll learn how to start / stop / terminate our instance.

Why AWS Lambda



Amazon EC2

- Virtual Servers in the Cloud
- Limited by RAM and CPU
- Continuously running
- Scaling means intervention to add / remove servers

Amazon Lambda

- Virtual **functions** – no servers to manage!
- Limited by time - **short executions**
- Run **on-demand**
- Scaling is automated!

Benefits of AWS Lambda

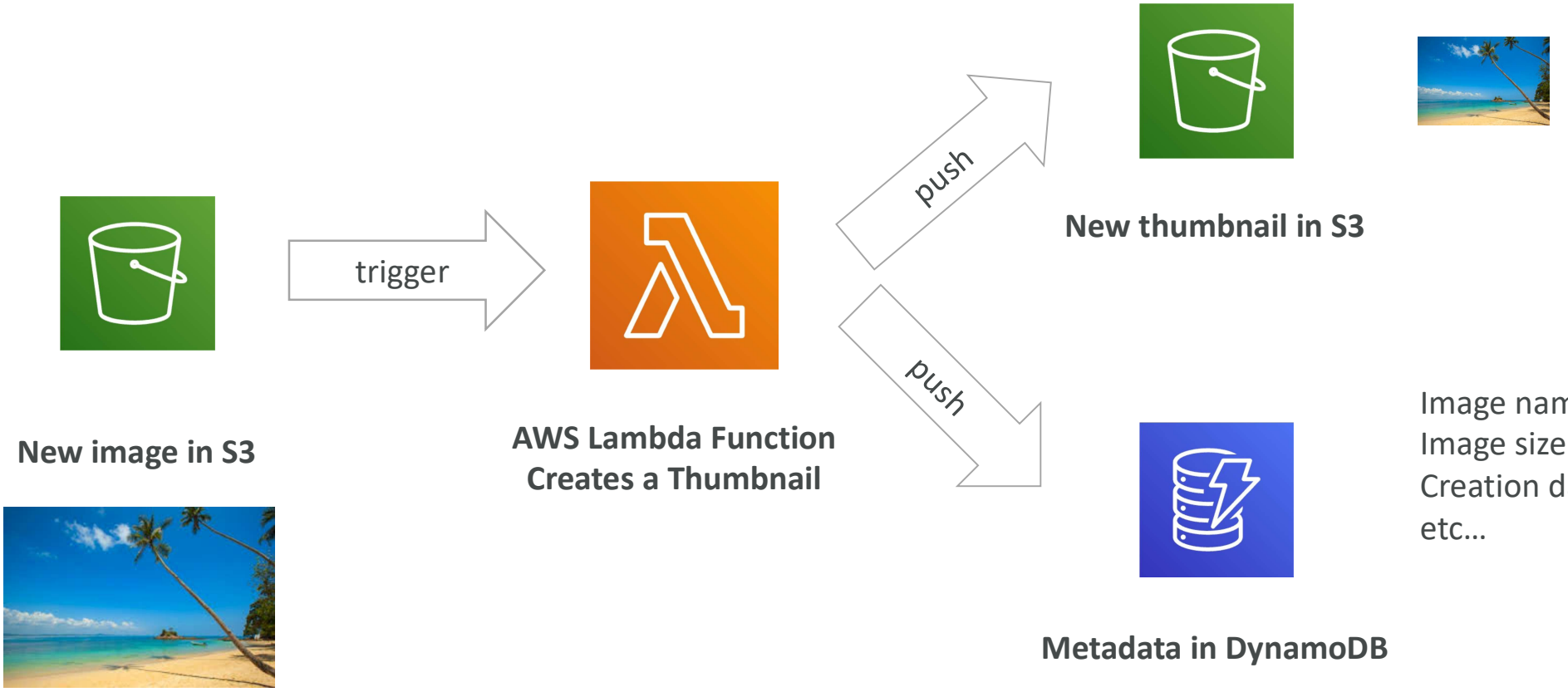
- Easy Pricing:
  - Pay per request and compute time
  - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS suite of services
- **Event-Driven:** functions get invoked by AWS when needed
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions (up to 10GB of RAM!)
- Increasing RAM will also improve CPU and network!

AWS Lambda language support

- Node.js (JavaScript)
- Python
- Java
- C# (.NET Core) / Powershell
- Ruby
- Custom Runtime API (community supported, example Rust or Golang)

- Lambda Container Image
  - The container image must implement the Lambda Runtime API
  - ECS / Fargate is preferred for running arbitrary Docker images

Example: Serverless Thumbnail creation



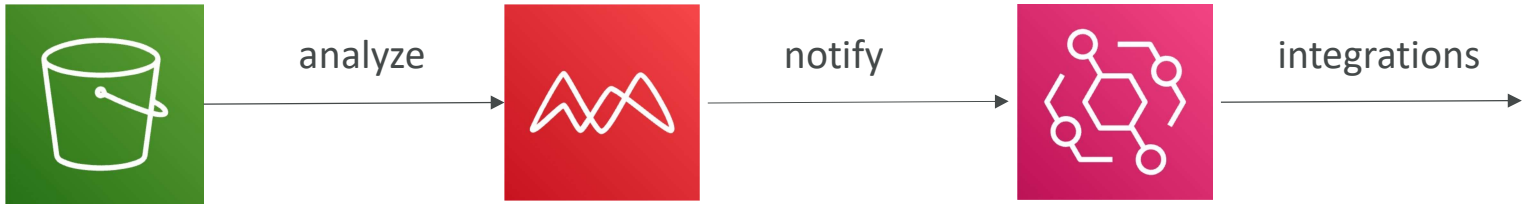
Example: Serverless CRON Job

AWS Lambda Pricing: example

- You can find overall pricing information here: https://aws.amazon.com/lambda/pricing/
- Pay per **calls**:
  - First 1,000,000 requests are free
  - $0.20 per 1 million requests thereafter ($0.0000002 per request)
- Pay per **duration**: (in increment of 1 ms)
  - 400,000 GB-seconds of compute time per month for FREE
  - == 400,000 seconds if function is 1GB RAM
  - == 3,200,000 seconds if function is 128 MB RAM
  - After that $1.00 for 600,000 GB-seconds
- It is usually very cheap to run AWS Lambda so it's very popular

AWS Macie

- Amazon Macie is a fully managed data security and data privacy service that uses **machine learning and pattern matching to discover and protect your sensitive data in AWS.**
- Macie helps identify and alert you to **sensitive data, such as personally identifiable information (PII)**

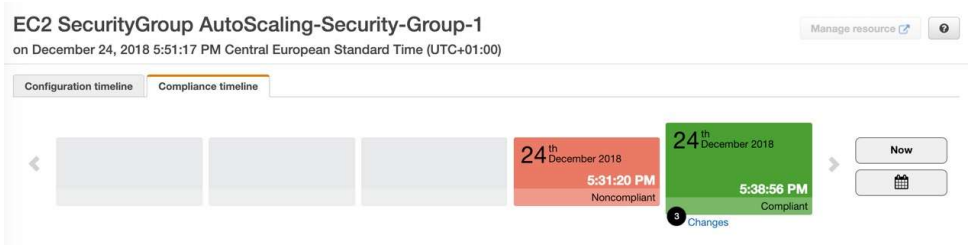


AWS Config

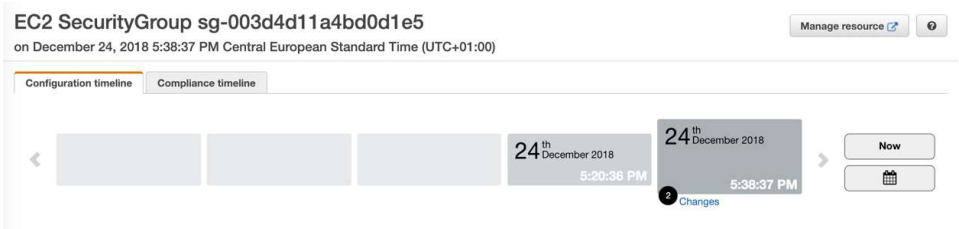
- Helps with **auditing and recording compliance of your AWS resources**
- Helps **record configurations and changes over time**
- Possibility of storing the configuration data into S3 (analyzed by Athena)
- Questions that can be solved by AWS Config:
  - Is there unrestricted SSH access to my security groups?
  - Do my buckets have any public access?
  - How has my ALB configuration changed over time?
- You can receive alerts (SNS notifications) for any changes
- AWS Config is a per-region service
- Can be aggregated across regions and accounts

AWS Config Resource

- View compliance of a resource over time



- View configuration of a resource over time



- View CloudTrail API calls if enabled

Amazon Inspector

- Automated Security Assessments

- For EC2 instances
  - Leveraging the AWS System Manager (SSM) agent
  - Analyze against unintended network accessibility
  - Analyze the running OS against known vulnerabilities
- For Container Images push to Amazon ECR
  - Assessment of Container Images as they are pushed
- For Lambda Functions
  - Identifies software vulnerabilities in function code and package dependencies
  - Assessment of functions as they are deployed

- Reporting & integration with AWS Security Hub
- Send findings to Amazon Event Bridge

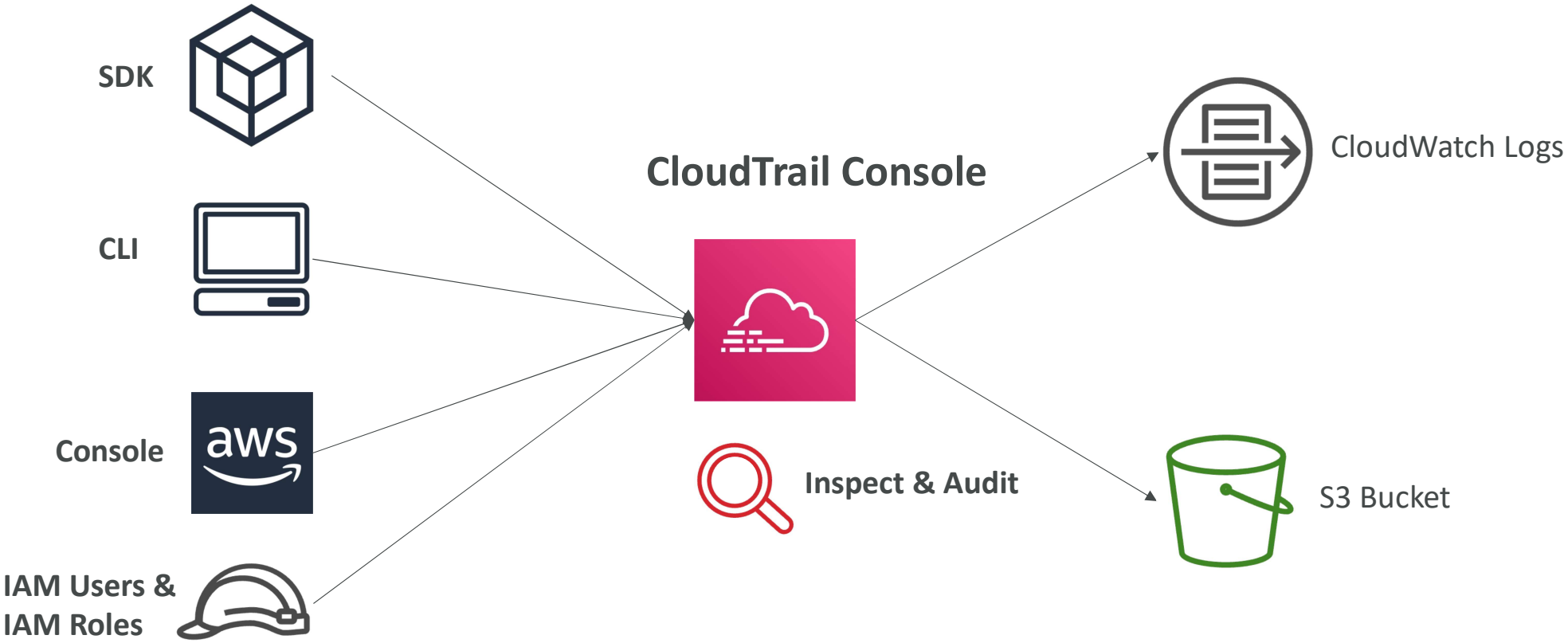
What does Amazon Inspector evaluate?

- Remember: only for EC2 instances, Container Images & Lambda functions
- Continuous scanning of the infrastructure, only when needed
- Package vulnerabilities (EC2, ECR & Lambda) – database of CVE
- Network reachability (EC2)
- A risk score is associated with all vulnerabilities for prioritization

AWS CloudTrail

- **Provides governance, compliance and audit for your AWS Account**
- CloudTrail is enabled by default!
- Get **an history of events / API calls made within your AWS Account** by:
  - Console
  - SDK
  - CLI
  - AWS Services
- Can put logs from CloudTrail into CloudWatch Logs or S3
- **A trail can be applied to All Regions (default) or a single Region.**
- If a resource is deleted in AWS, investigate CloudTrail first!

CloudTrail Diagram



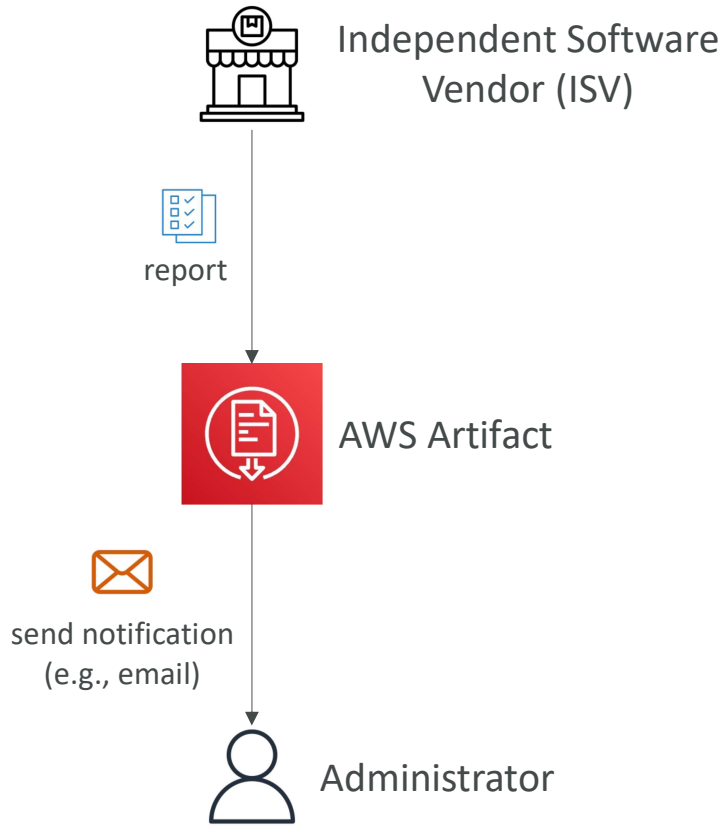
AWS Artifact (not really a service)

- Portal that provides customers with on-demand access to AWS compliance documentation and AWS agreements
- Artifact Reports - Allows you to download AWS security and compliance documents from third-party auditors, like AWS ISO certifications, Payment Card Industry (PCI), and System and Organization Control (SOC) reports
- Artifact Agreements - Allows you to review, accept, and track the status of AWS agreements such as the Business Associate Addendum (BAA) or the Health Insurance Portability and Accountability Act (HIPAA) for an individual account or in your organization

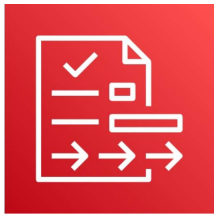
- Can be used to **support internal audit or compliance**

AWS Artifact – Third-Party Reports

- On-demand access to security compliance reports of Independent Software Vendors (ISVs)
- ISV compliance reports will only be accessible to the AWS customers who have been granted access to AWS Marketplace Vendor Insights for a specific ISV
- Ability to receive notifications when new reports are available



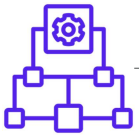
AWS Audit Manager



- Assess risk and compliance of your AWS workloads
- Continuously audit AWS services usage and prepare audits
- Prebuilt frameworks include:
  - CIS AWS Foundations Benchmark 1.2.0 & 1.3.0
  - General Data Protection Regulation (GDPR),
  - Health Insurance Portability and Accountability Act (HIPAA)
  - Payment Card Industry Data Security Standard (PCI DSS) v3.2.1
  - Service Organization Control 2 (SOC 2)
- Generates reports of compliance alongside evidence folders

AWS Audit Manager

AWS Audit Manager
continuously audit your AWS usage to simplify how you assess risk and compliance



Select Framework
choose a prebuilt framework with included controls, or create your own custom framework

Define the Scope
specify the in-scope accounts and services in a region for your assessment

activate the assessment to continuously gather evidence

Audit Manager conducts Automated Evidence Collection

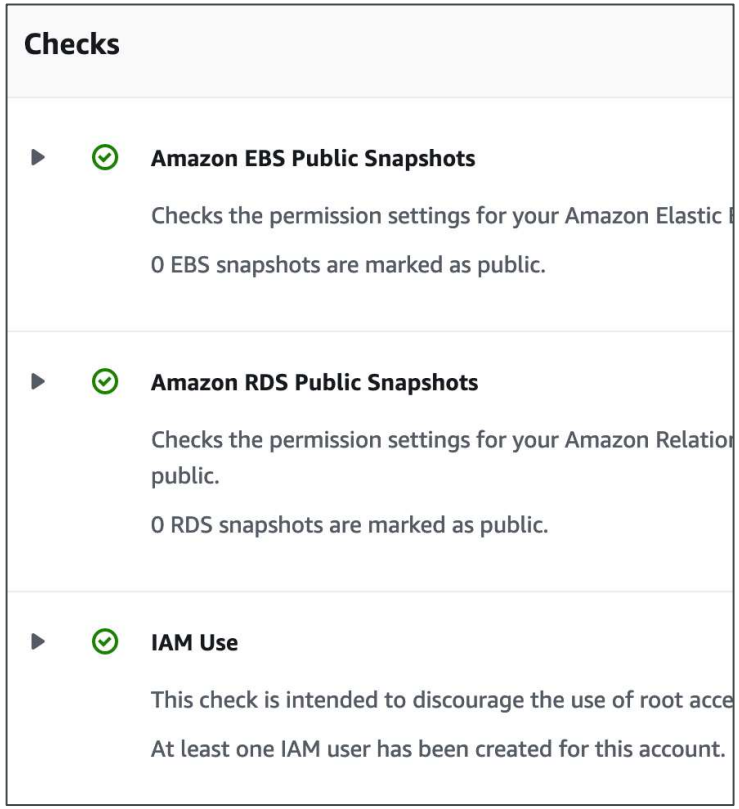
conduct control reviews, or delegate to resource owners to validate

Identify Root Causes
filter and group your data to deep dive into causes of non-compliance

Generate Reports
create auti-ready assessment reports with links to evidence

Trusted Advisor

- No need to install anything – high level AWS account assessment
- Analyze your AWS accounts and provides recommendation on 6 categories:
  - Cost optimization
  - Performance
  - Security
  - Fault tolerance
  - Service limits
  - Operational Excellence
- Business & Enterprise Support plan
  - Full Set of Checks
  - Programmatic Access using AWS Support API



VPC – Crash Course

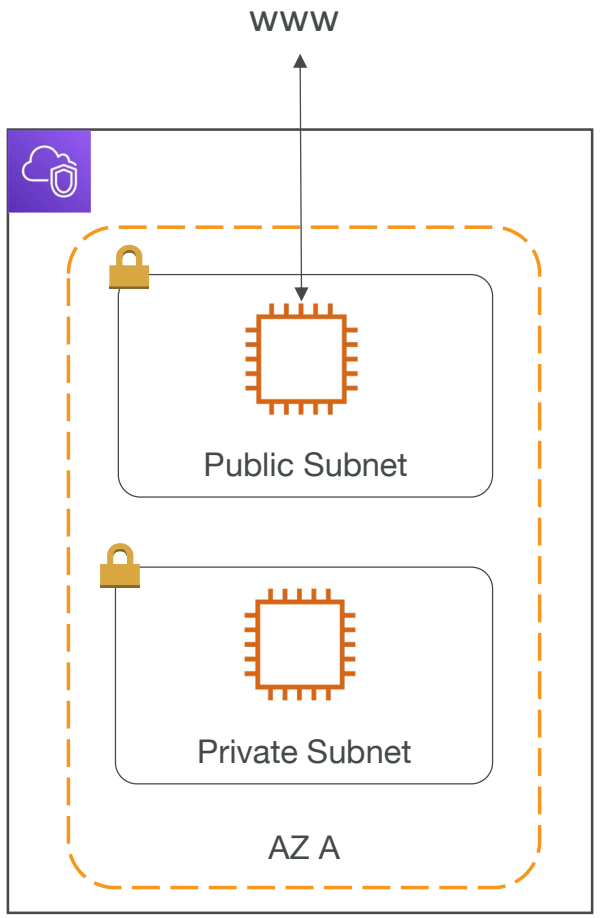
- VPC is something you should know in depth for the AWS Certified Solutions Architect Associate & AWS Certified SysOps Administrator exams

- At the AWS Certified AI Practitioner level, you should know about:
  - VPC, Subnets, Internet Gateways & NAT Gateways
  - VPC Endpoints & PrivateLink

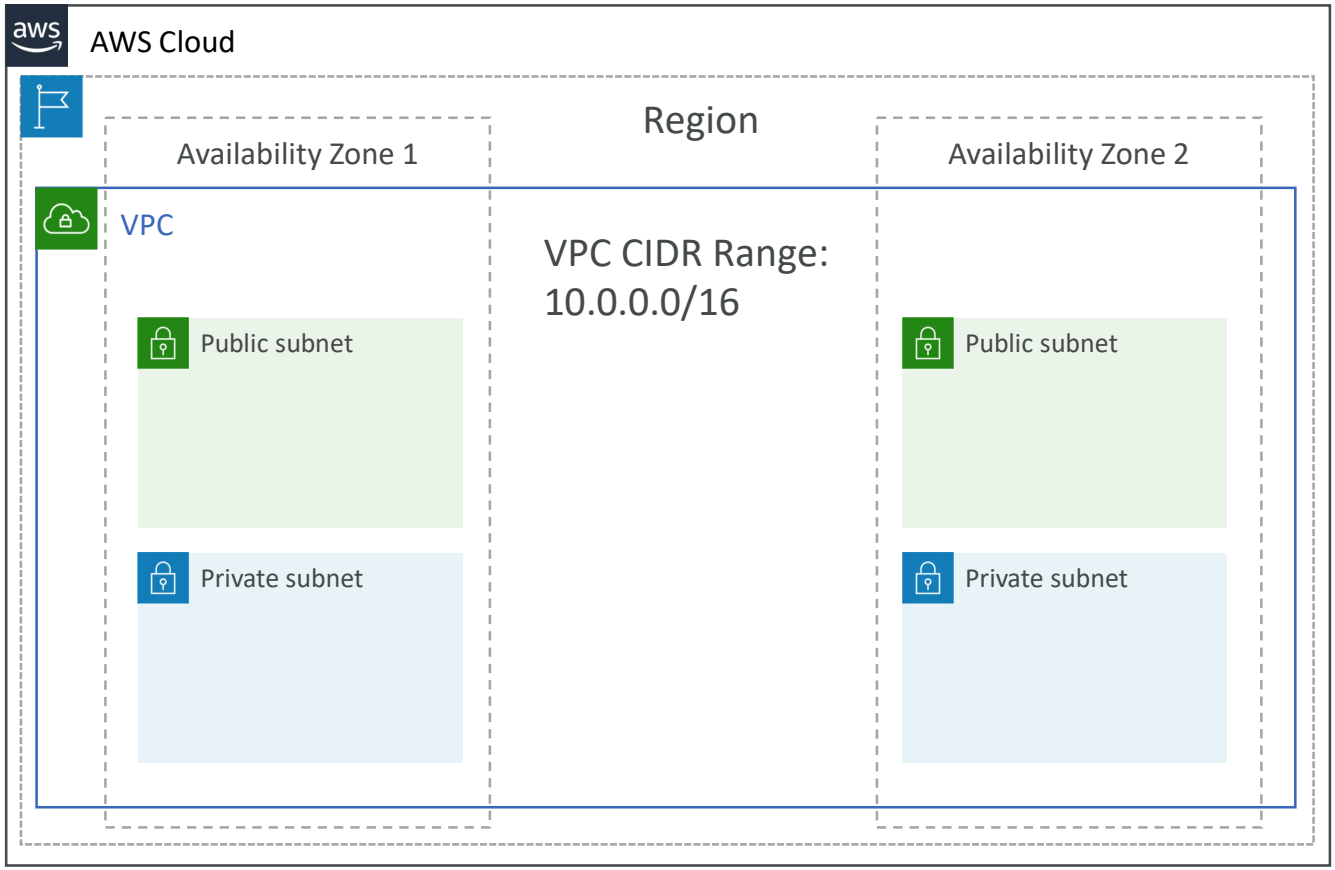
- Questions at the exam that are VPC related are usually for deploying models privately and accessing AWS services without going through the internet

VPC & Subnets Primer

- **VPC - Virtual Private Cloud:** private network to deploy your resources (regional resource)
- **Subnets** allow you to partition your network inside your VPC (Availability Zone resource)
- A **public subnet** is a subnet that is accessible from the internet
- A **private subnet** is a subnet that is not accessible from the internet



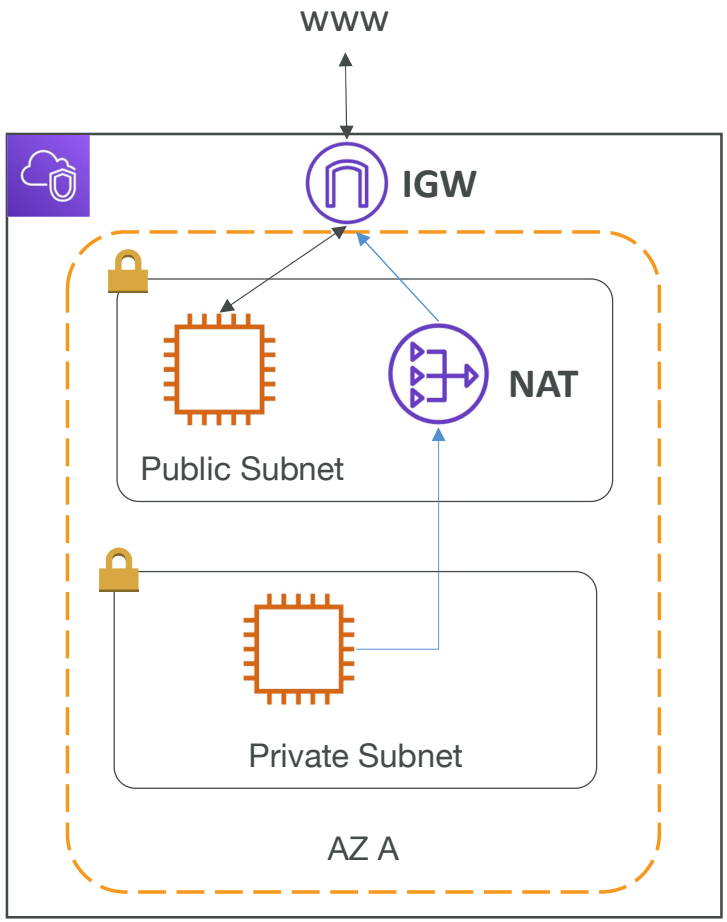
VPC Diagram



Internet Gateway & NAT Gateways

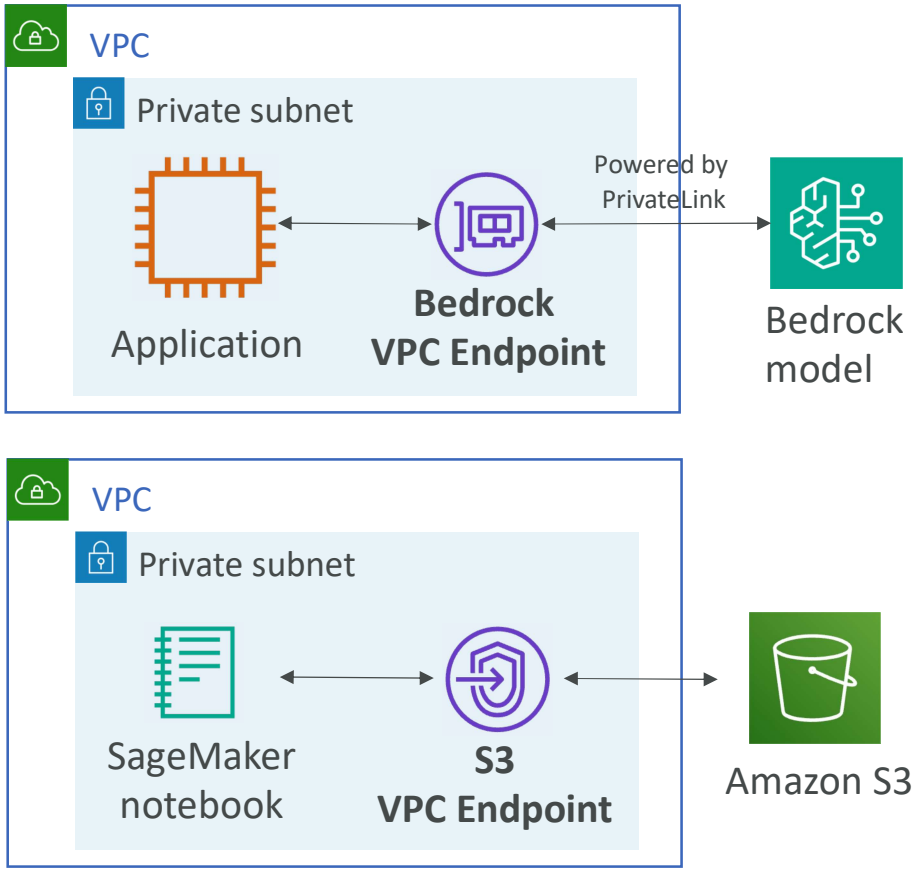
- **Internet Gateways** helps our VPC instances connect with the internet
- Public Subnets have a route to the internet gateway.

- **NAT Gateways** (AWS-managed) allow your instances in your **Private Subnets** to access the internet while remaining private



VPC Endpoints and PrivateLink

- AWS Services are by default accessed over the public internet
- Applications deployed in Private Subnets in VPC may not have internet access
- We want to use VPC endpoints
  - Access an AWS service privately without going over the public internet
  - Usually powered by AWS PrivateLink
  - Keep your network traffic internal to AWS
  - Example: your application deployed in a VPC can access a Bedrock model privately
- S3 Gateway Endpoint
  - Access Amazon S3 privately
  - There's also an S3 Interface Endpoint
  - Example: SageMaker notebooks can access S3 data privately



AWS Security Services – Section Summary

- **IAM Users** – mapped to a physical user, has a password for AWS Console
- **IAM Groups** – contains users only
- **IAM Policies** – JSON document that outlines permissions for users or groups
- **IAM Roles** – for EC2 instances or AWS services
- **EC2 Instance** – AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- **AWS Lambda** – serverless, Function as a Service, seamless scaling
- **VPC Endpoint powered by AWS PrivateLink** – provide private access to AWS Services within VPC
- **S3 Gateway Endpoint:** access Amazon S3 privately

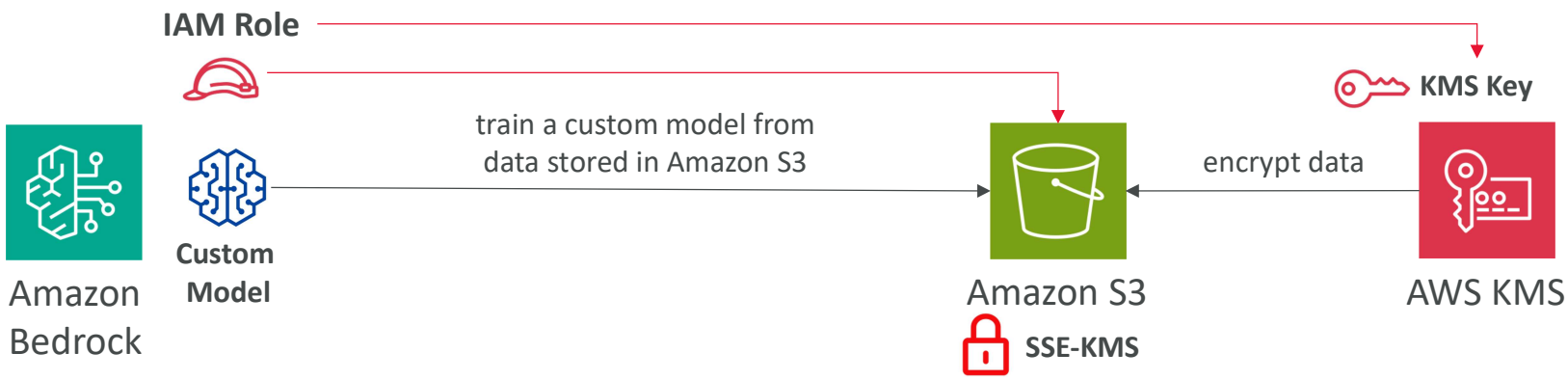
AWS Security Services – Section Summary

- **Macie** – find sensitive data (ex: PII data) in Amazon S3 buckets
- **Config** – track config changes and compliance against rules
- **Inspector** – find software vulnerabilities in EC2, ECR Images, and Lambda functions
- **CloudTrail** – track API calls made by users within account
- **Artifact** – get access to compliance reports such as PCI, ISO, etc…
- **Trusted Advisor** – to get insights, Support Plan adapted to your needs

AWS Services for Bedrock

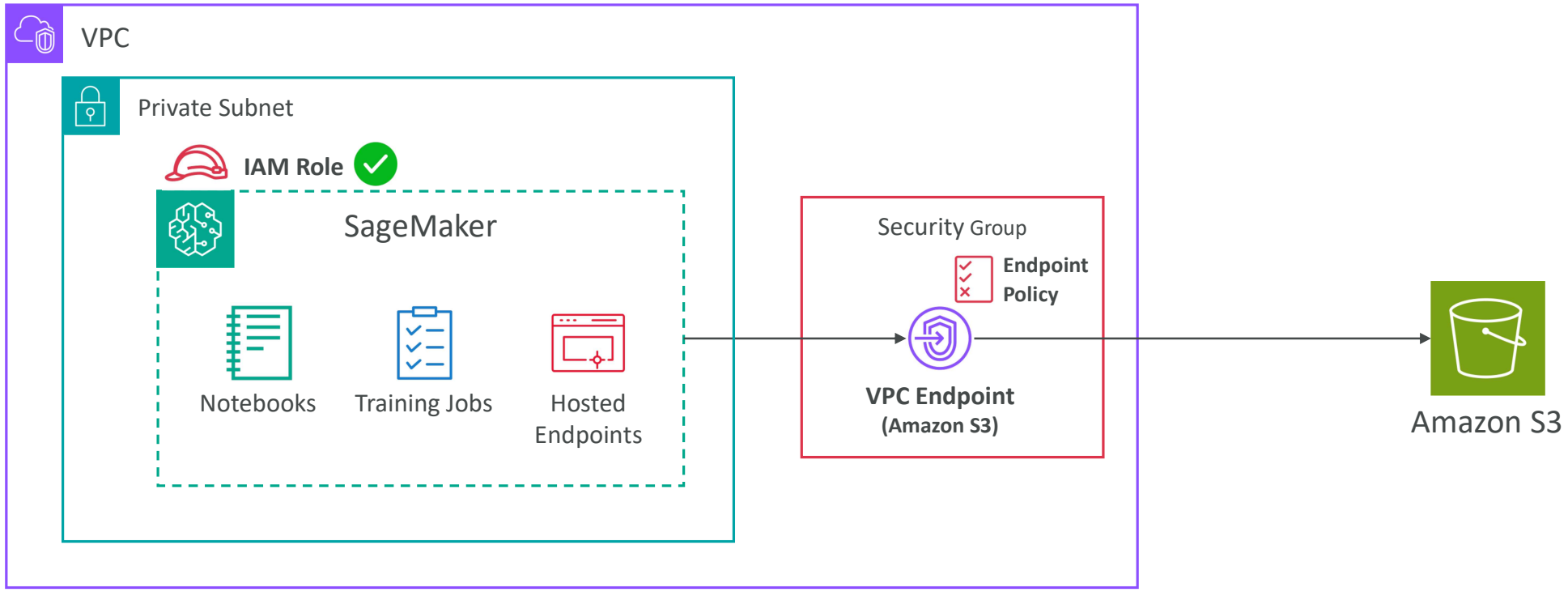
- IAM with Bedrock
  - Implement identity verification and resource-level access control
  - Define roles and permissions to access Bedrock resources (e.g., data scientists
- GuardRails for Bedrock
  - Restrict specific topics in a GenAI application
  - Filter harmful content
  - Ensure compliance with safety policies by analyzing user inputs
- **CloudTrail with Bedrock:** Analyze API calls made to Amazon Bedrock
- **Config with Bedrock:** look at configuration changes within Bedrock
- **PrivateLink with Bedrock:** keep all API calls to Bedrock within the private VPC

Bedrock must access an encrypted S3 bucket

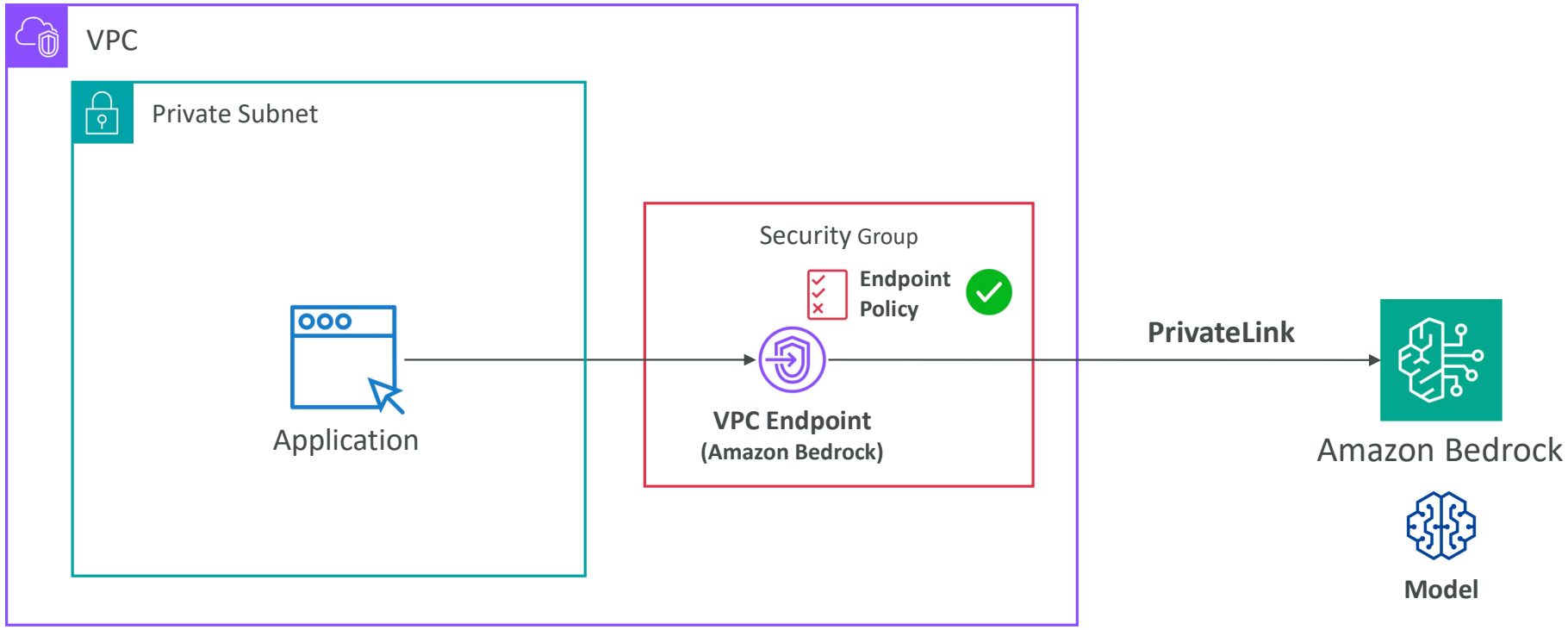


- Bedrock must have an IAM Role that gives it access to:
  - Amazon S3
  - The **KMS Key** with the decrypt permission

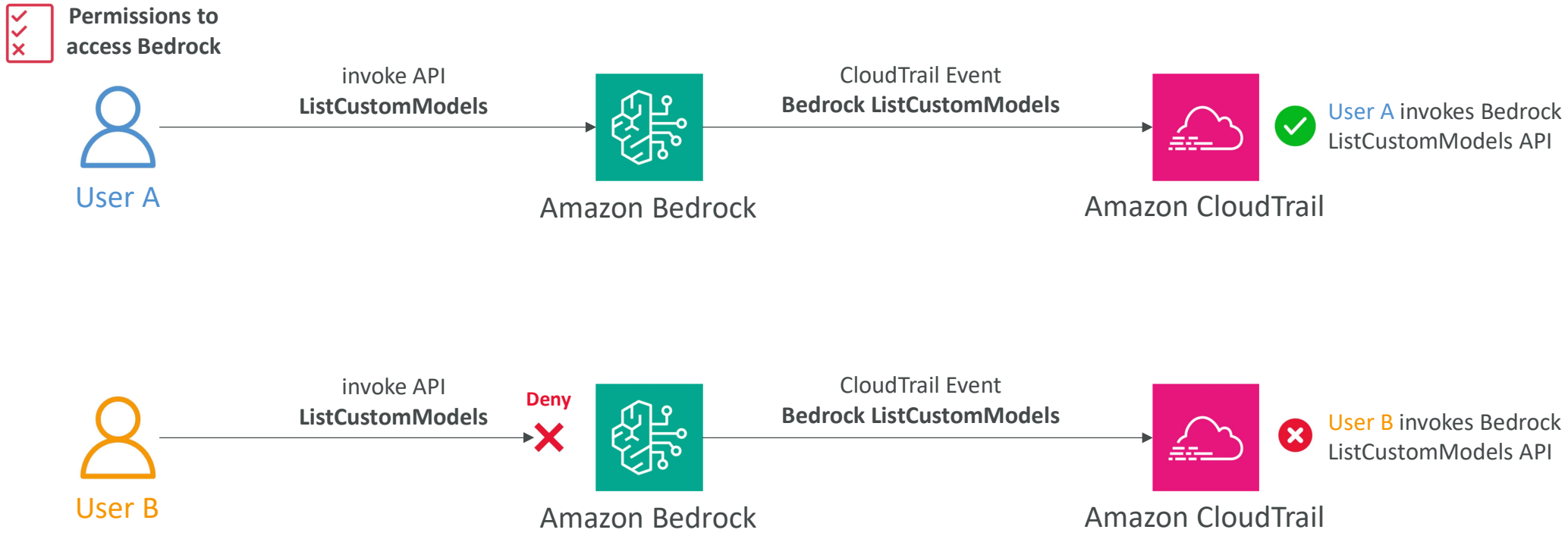
Deploy SageMaker Model in your VPC



Access Bedrock Model using an App in VPC



Analyze Bedrock access with CloudTrail



Exam Preparation

State of learning checkpoint

- Let's look how far we've gone on our learning journey

- https://aws.amazon.com/certification/certified-ai-practitioner/

Sample Questions Walkthrough

- https://explore.skillbuilder.aws/learn/course/external/view/elearning/19790/exam-prep-official-practice-question-set-aws-certified-ai-practitioner-aif-c01-english

Your AWS Certification journey

Foundational

Knowledge-based certification for
foundational understanding of AWS Cloud.
No prior experience needed.

Associate

Role-based certifications that showcase your knowledge
and skills on AWS and build your credibility as an AWS Cloud professional.
Prior cloud and/or strong on-premises IT experience recommended.

Professional

Role-based certifications that validate advanced skills
and knowledge required to design secure, optimized,
and modernized applications and to automate processes on AWS.
2 years of prior AWS Cloud experience recommended.

Specialty

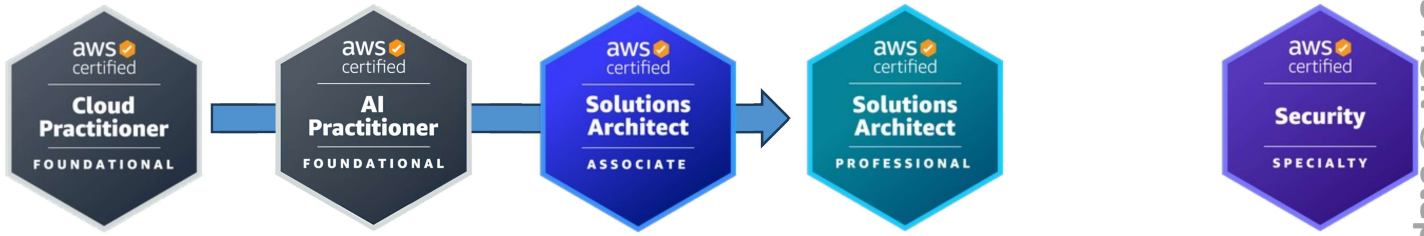
Dive deeper and position yourself as a trusted advisor to your
stakeholders and/or customers in these strategic areas.
Refer to the exam guides on the exam pages for recommended experience.

AWS Certification Paths – Architecture

Architecture

Solutions Architect

Design, develop, and manage cloud infrastructure and assets, work with DevOps to migrate applications to the cloud



optional for IT/ cloud professionals

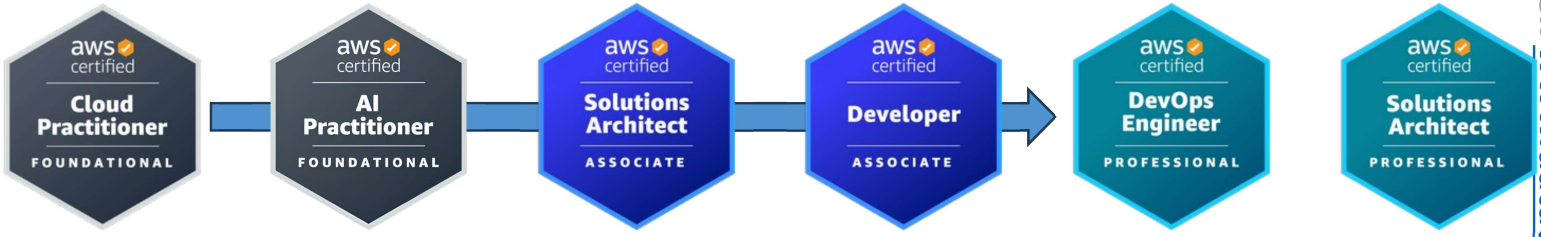
recommended for IT/cloud professionals to leverage AI

Dive Deep

Architecture

Application Architect

Design significant aspects of application architecture including user interface, middleware, and infrastructure, and ensure enterprise-wide scalable, reliable, and manageable systems



optional for IT/ cloud professionals

recommended for IT/cloud professionals to leverage AI

Dive Deep

https://d1.awsstatic.com/training-and-certification/docs/AWS_certification_paths.pdf

AWS Certification Paths – Operations

Operations
Systems Administrator

Install, upgrade, and maintain computer components and software, and integrate automation processes

optional for IT/ cloud professionals

Dive Deep

Operations
Cloud Engineer

Implement and operate an organization's networked computing infrastructure and Implement security systems to maintain data safety

optional for IT/ cloud professionals

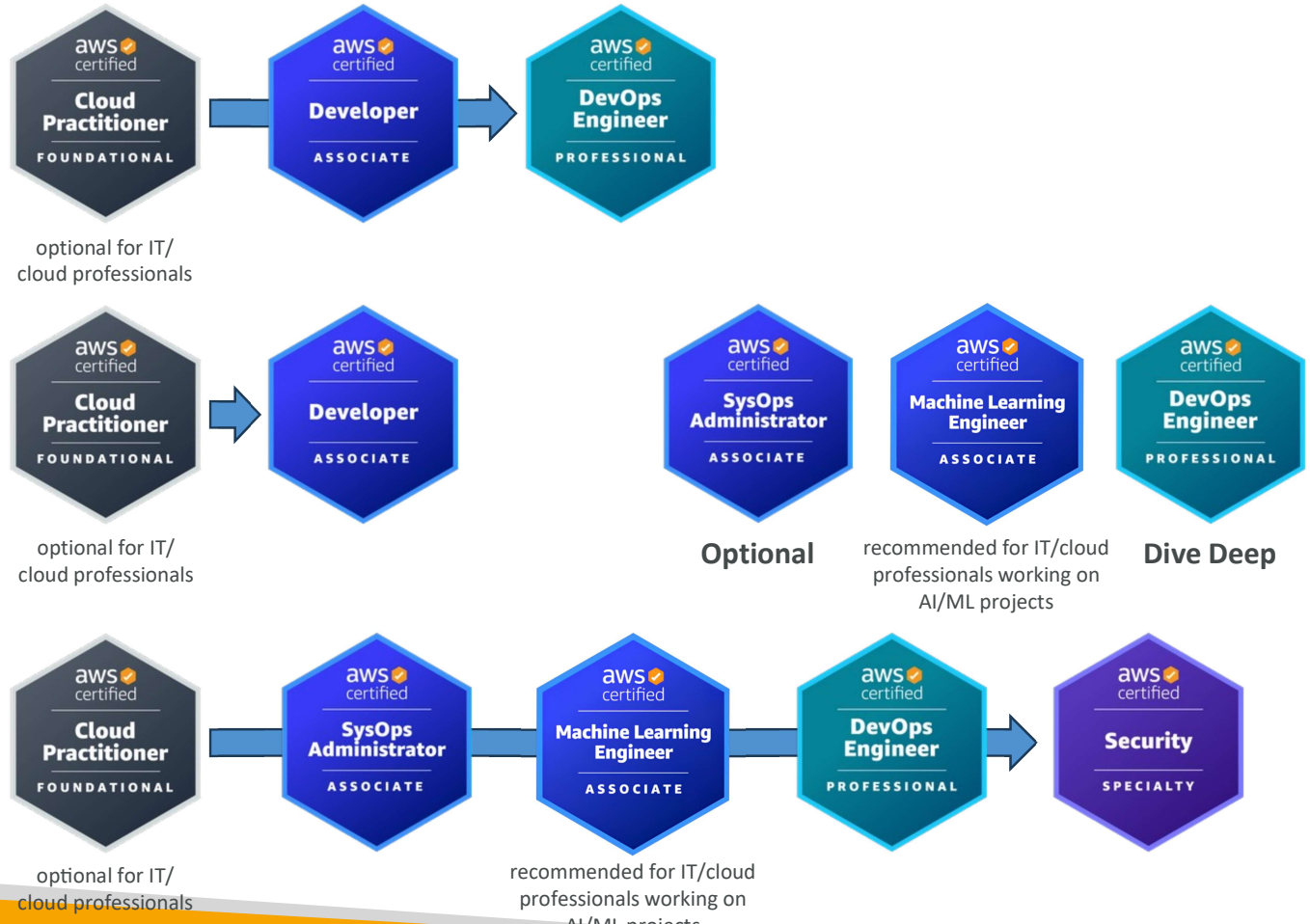
Dive Deep

AWS Certification Paths – DevOps

DevOps
Test Engineer
Embed testing and quality best practices for software development from design to release, throughout the product life cycle

DevOps
Cloud DevOps Engineer
Design, deployment, and operations of large-scale global hybrid cloud computing environment, advocating for end-to-end automated CI/CD DevOps pipelines

DevOps
DevSecOps Engineer
Accelerate enterprise cloud adoption while enabling rapid and stable delivery of capabilities using CI/CD principles, methodologies, and technologies



AWS Certification Paths – Security

Security

Cloud Security Engineer

Design computer security architecture and develop detailed cyber security designs. Develop, execute, and track performance of security measures to protect information

optional for IT/ cloud professionals

recommended for IT/cloud professionals to secure AI/ML systems

Dive Deep

Security

Cloud Security Architect

Design and implement enterprise cloud solutions applying governance to identify, communicate, and minimize business and technical risks

optional for IT/ cloud professionals

recommended for IT/cloud professionals to secure AI/ML systems

Dive Deep

AWS Certification Paths – Development & Networking

Development

Software Development Engineer

Develop, construct, and maintain software across platforms and devices

optional for IT/ cloud professionals

recommended for IT/cloud professionals to leverage AI

Networking

Network Engineer

Design and implement computer and information networks, such as local area networks (LAN), wide area networks (WAN), intranets, extranets, etc.

optional for IT/ cloud professionals

Dive Deep

AWS Certification Paths – Data Analytics & AI/ML

Data Analytics

Cloud Data Engineer

Automate collection and processing of structured/semi-structured data and monitor data pipeline performance

optional for IT/ cloud professionals

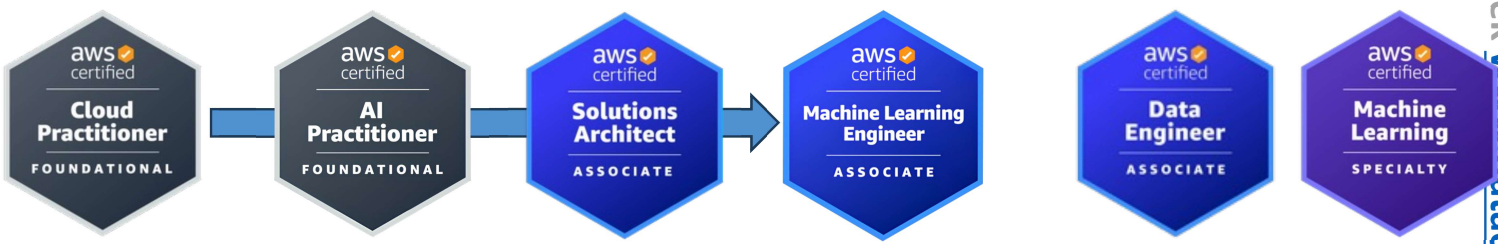
recommended for IT/cloud professionals working on AI/ML projects

Dive Deep

AI/ML

Machine Learning Engineer

Research, build, and design artificial intelligence (AI) systems to automate predictive models, and design machine learning systems, models, and schemes



optional for IT/ cloud professionals

optional for AI/ML professionals

Dive Deep

AWS Certification Paths – AI/ML

AI/ML

Prompt Engineer

Design, test, and refine text prompts to optimize the performance of AI language models

AI/ML

Machine Learning Ops Engineer

Build and maintain AI and ML platforms and infrastructure. Design, implement, and operationally support AI/ML model activity and deployment infrastructure

AI/ML

Data Scientist

Develop and maintain AI/ML models to solve business problems. Train and fine tune models and evaluate their performance

Congratulations!

Congratulations!

- Congrats on finishing the course!
- I hope you will pass the exam without a hitch ☺

- If you haven't done so yet, I'd love a review from you!
- If you passed, I'll be more than happy to know I've helped
  - Post it in the Q&A to help & motivate other students. Share your tips!
  - Post it on LinkedIn and tag me!

- Overall, I hope you learned how to use AWS and that you will be a tremendously good AWS AI Practitioner