ML OPERATIONS (MLOPS)

Objectives

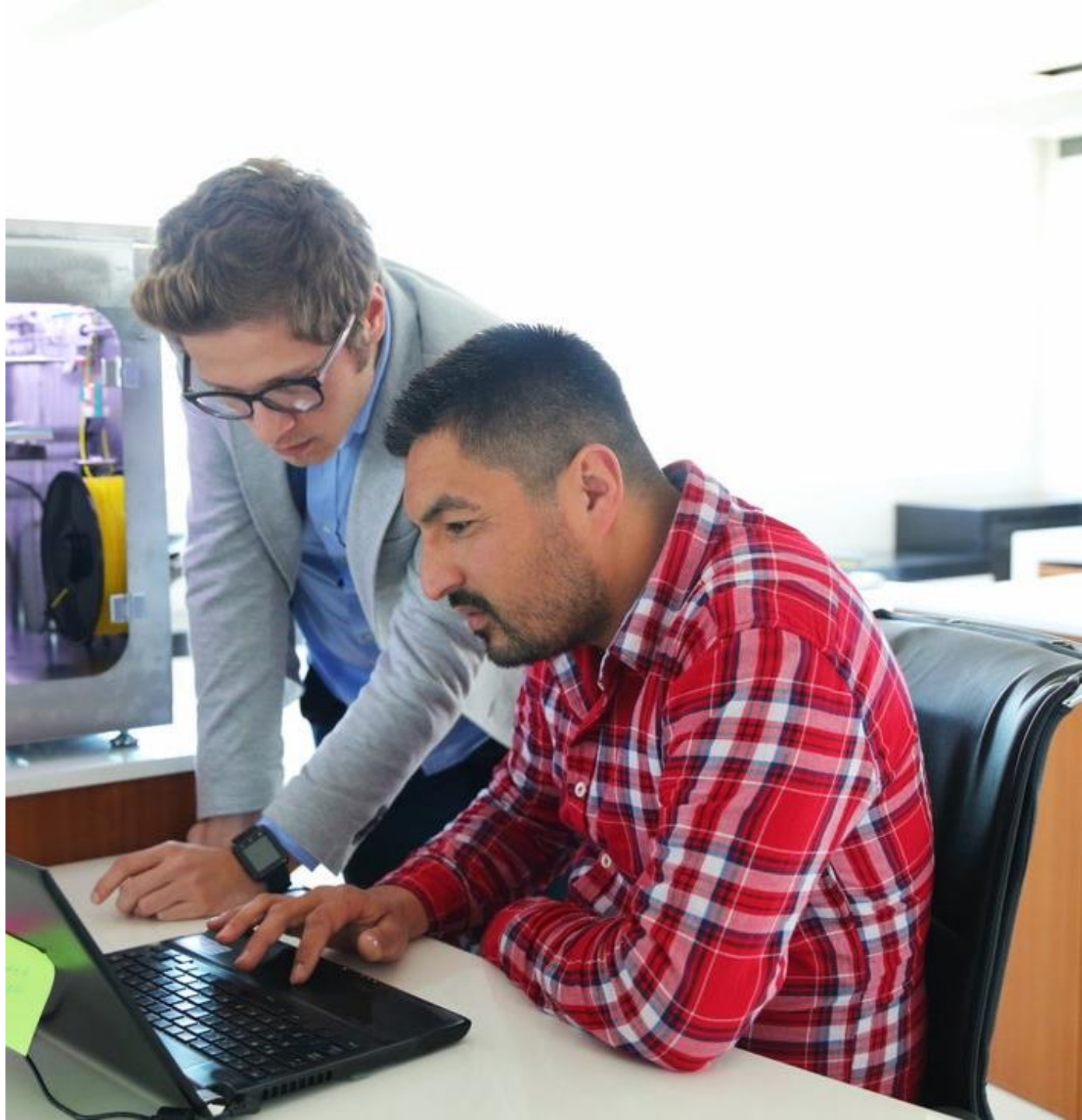
- Examine MLOps experimentation, repeatable processes, and scalable systems
- Explore managing technical debt and achieving production readiness
- Examine model monitoring and retraining
- Compare metrics for model performance including accuracy, F1 score, and area under the ROC curve (AUC)
- Compare business metrics like Cost per user, development costs, customer feedback, and return on investment (ROI)

MLOPS EXPERIMENTATION

- MLOps experimentation entails a structured approach to managing and enhancing the ML lifecycle, encompassing the development, training, and deployment of ML models
- This approach to MLOps experimentation provides a comprehensive framework for managing the ML lifecycle, ensuring that models are developed, deployed, and maintained efficiently and effectively

KEY ASPECTS OF MLOPS EXPERIMENTATION

- **Experiment tracking:** AWS emphasizes the importance of tracking experiments to ensure reproducibility and collaboration
- **Version control:** maintaining version control of datasets, code, and models is crucial
- **Automated workflows:** automating the ML workflow is a key component of MLOps



KEY ASPECTS OF MLOPS EXPERIMENTATION

- **Collaboration:** AWS supports collaboration among data scientists, ML engineers, and other stakeholders
- **Scalability:** AWS services are designed to scale with the needs of the ML project, including scalable compute resources for training models and scalable storage for large datasets

KEY ASPECTS OF MLOPS EXPERIMENTATION

- **Continuous integration and continuous deployment (CI/CD)**: integrating ML workflows with CI/CD pipelines ensures that models are continuously tested and deployed
- **Monitoring and feedback**: continuous monitoring of deployed models is essential for maintaining performance
  - AWS provides tools like Amazon SageMaker Model Monitor to track model performance, detect anomalies, and provide feedback for retraining models

MLOPS REPEATABLE PROCESSES

- MLOps repeatable processes are essential for ensuring consistency, reliability, and efficiency in the ML lifecycle
- These processes help automate and standardize various stages of ML development, from data preparation to model deployment and monitoring

KEY ASPECTS OF MLOPS REPEATABLE PROCESSES

- **Automated workflows:** creating automated workflows for data preprocessing, model training, evaluation, and deployment
- **Version control:** implementing version control for datasets, code, and models
- **Experiment tracking:** tracking experiments to ensure reproducibility and collaboration

KEY ASPECTS OF MLOPS REPEATABLE PROCESSES

- **CI/CD:** integrating ML workflows with CI/CD pipelines to automate the testing and deployment of models
- **Scalability:** designing processes that can scale with the needs of the ML project
- **Monitoring and feedback:** continuously monitoring deployed models to ensure they perform as expected

SCALABLE SYSTEMS

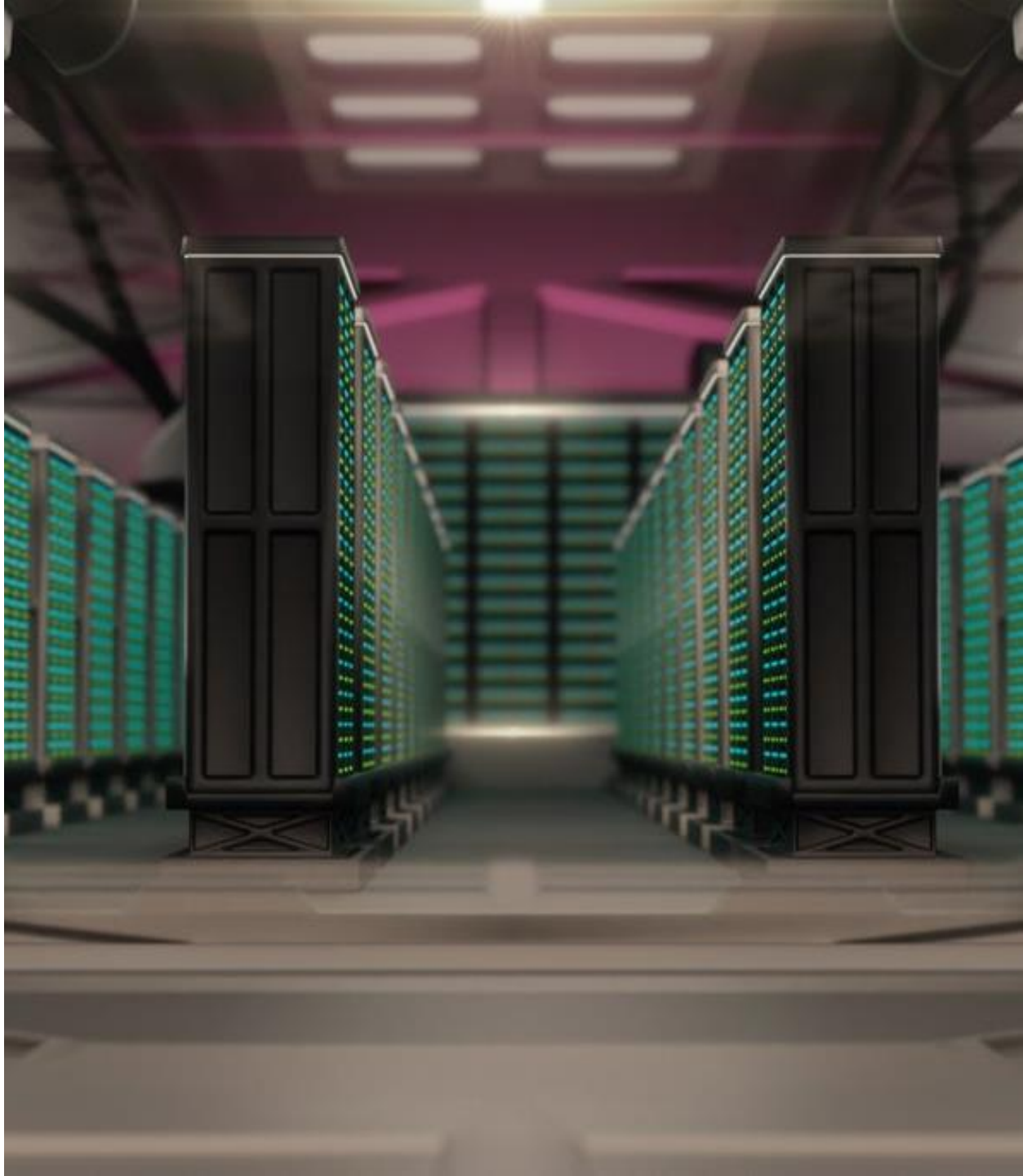
- MLOps scalable systems are constructed to efficiently and effectively manage the expanding demands of ML workloads
- These systems facilitate the development, training, deployment, and monitoring of ML models at scale, accommodating growing data volumes, model complexity, and user requirements

KEY ASPECTS OF MLOPS SCALABLE SYSTEMS

- **Infrastructure scalability:** AWS provides scalable infrastructure services such as Amazon EC2, Amazon S3, and SageMaker
- **Automated workflows:** automating the ML pipeline is crucial for scalability
- **Distributed training:** for large datasets and complex models, distributed training is essential

KEY ASPECTS OF MLOPS SCALABLE SYSTEMS

- **Model deployment:** scalable model deployment ensures they can handle varying levels of traffic
- **Monitoring and management:** continuous monitoring of deployed models is vital for maintaining performance
- **Data management:** efficient data management is key to scalability
- **Security and compliance:** scalable systems must also ensure security and compliance



MLOPS TECHNICAL DEBT

- **Technical debt** refers to the accumulated shortcuts, suboptimal practices, and temporary solutions in the MLOps lifecycle that can hinder the efficiency, scalability, and maintainability of ML systems over time
- This debt is a result of various factors, such as:
  - Rushed deployments
  - Lack of automation
  - Inadequate documentation
  - Insufficient testing

MANAGING TECHNICAL DEBT

- Managing technical debt involves adopting best practices from software engineering and DevOps to ensure that ML systems are maintainable, scalable, and efficient
- The AWS approach to managing technical debt in MLOps provides a robust framework for maintaining the quality and performance of ML systems, ensuring that they continue to deliver value over time

ACHIEVING PRODUCTION READINESS

- Achieving production readiness in MLOps involves implementing best practices that ensure ML models are reliable, scalable, and maintainable in a production environment
- The AWS approach to achieving production readiness in MLOps provides a robust framework for managing the ML lifecycle, ensuring that models are developed, deployed, and maintained efficiently and effectively

MLOPS MODEL MONITORING

- MLOps model monitoring is a critical practice for maintaining the quality and performance of ML models in production
- The AWS approach to MLOps model monitoring provides a comprehensive solution for maintaining the quality and performance of ML models in production, ensuring that they continue to deliver value over time

BENEFITS OF MLOPS MODEL MONITORING

Maintained performance

Ensures that models continue to perform well and provide accurate predictions over time

Early detection of issues

Helps in identifying and addressing issues such as data drift, model drift, and performance degradation before they impact the business

Compliance and accountability

Ensures that models adhere to regulatory requirements and ethical standards by monitoring for bias and providing explanations for predictions

MLOPS RETRAINING

- MLOps retraining is a crucial process to ensure that ML models remain accurate and relevant over time
- AWS's approach to MLOps retraining provides a comprehensive framework for maintaining the quality and performance of ML models in production, ensuring that they continue to deliver value over time

BENEFITS OF MLOPS MODEL RETRAINING

Maintained performance

Ensures that models continue to perform well and provide accurate predictions over time

Adaptability

Allows models to adapt to changing data distributions and new patterns in the data

Fairness and compliance

Ensures that models remain fair and compliant with regulatory requirements by continuously monitoring and mitigating bias

KEY ASPECTS OF MLOPS RETRAINING

- **Model drift detection:** over time, the performance of ML models can degrade due to changes in the underlying data distribution, known as model drift
- **Automated retraining pipelines:** AWS recommends automating the retraining process to ensure timely model updates
- **Data quality monitoring:** effective retraining requires high-quality data
  - AWS suggests monitoring the quality of the data used for retraining to ensure that it is representative of the current environment

KEY ASPECTS OF MLOPS RETRAINING

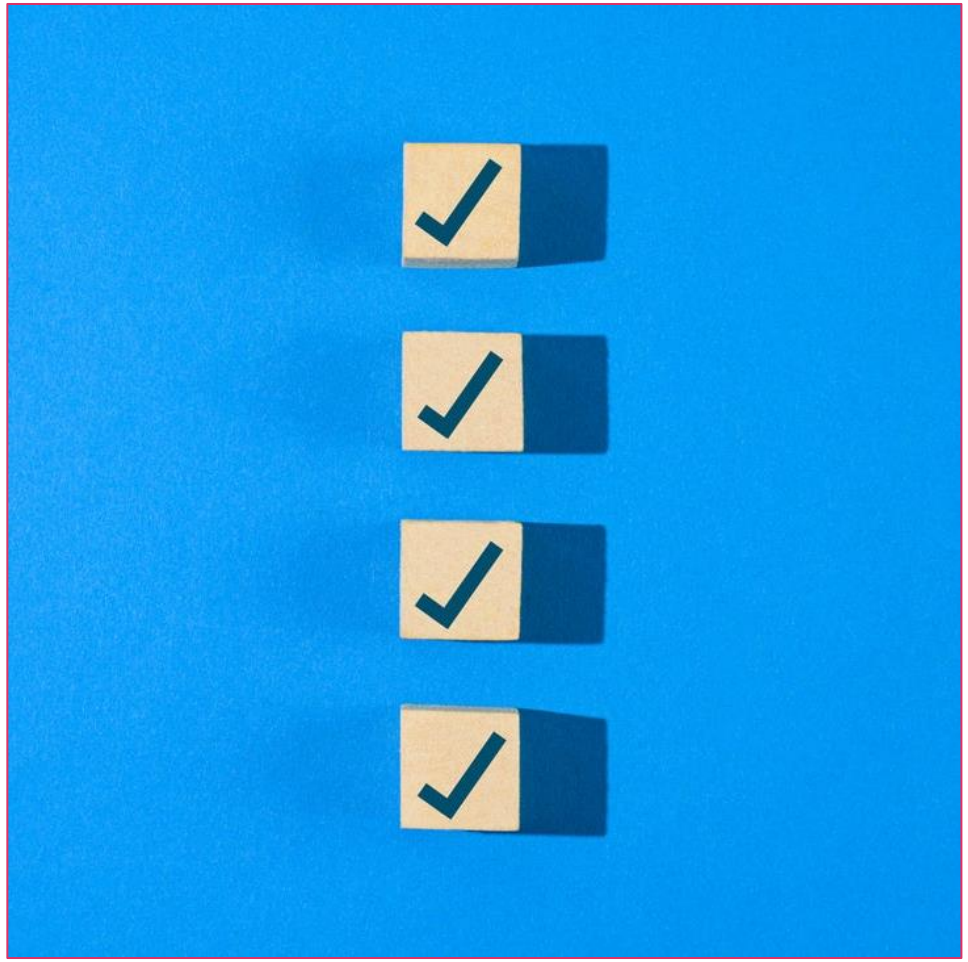
- **Human-in-the-Loop (HITL) workflows:** in some cases, human intervention is necessary to validate the accuracy of the model's predictions
  - AWS offers services like Amazon Augmented AI (Amazon A2I) to incorporate human feedback into the retraining process
- **Bias detection and mitigation**: it is important to monitor models for bias and ensuring fairness in predictions

KEY ASPECTS OF MLOPS RETRAINING

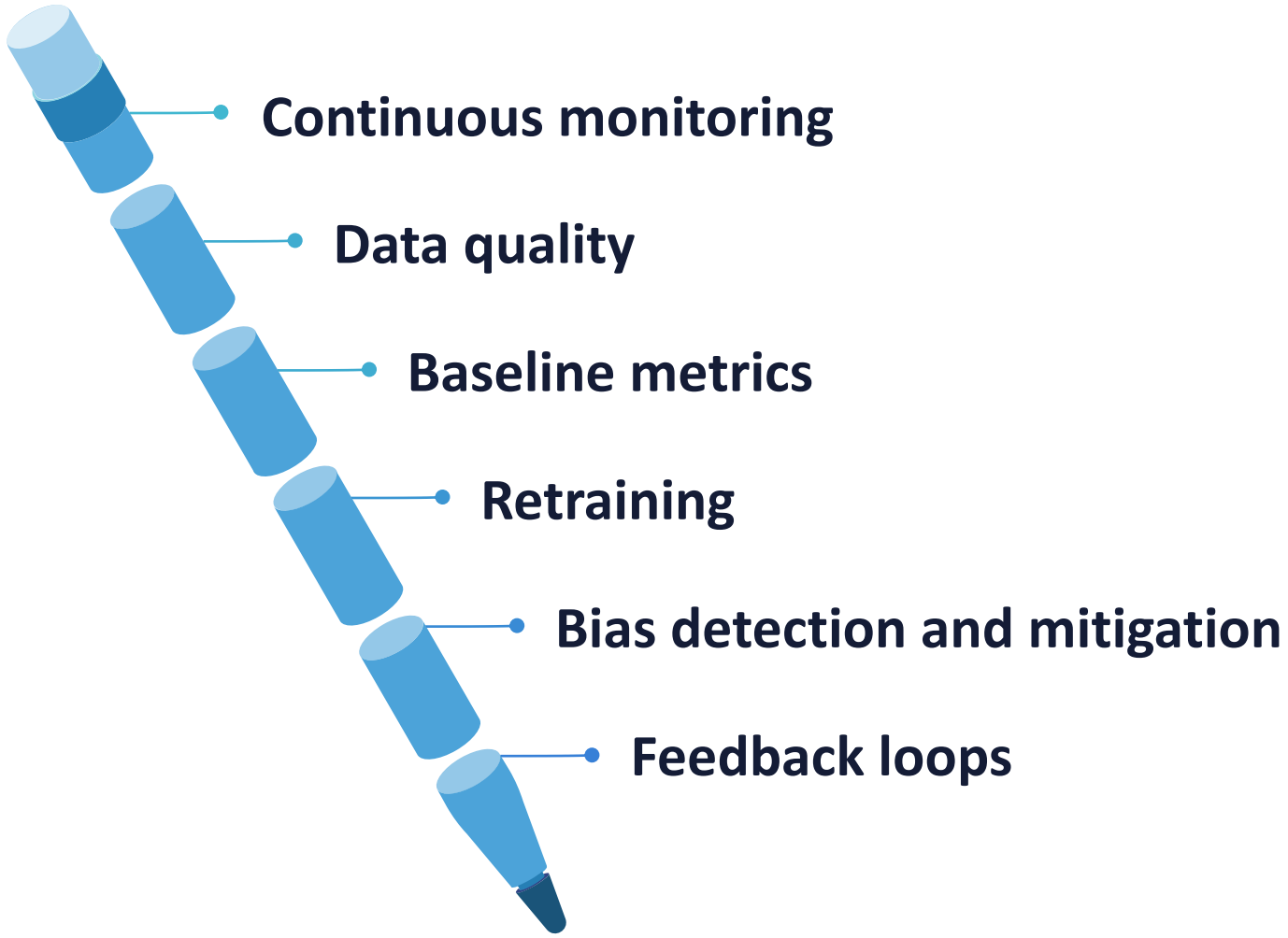
- **Continuous improvement:** retraining is an ongoing process that requires continuous monitoring and updating
- AWS recommends setting up a feedback loop between the production system and the training environment to ensure that models are continuously improved based on new data and feedback

MODEL PERFORMANCE: ACCURACY

- The **accuracy** of MLOps model performance is a critical metric that reflects how well a model's predictions align with actual outcomes
- By following best practices, AWS ensures that ML models maintain high accuracy and continue to deliver reliable predictions in production environments

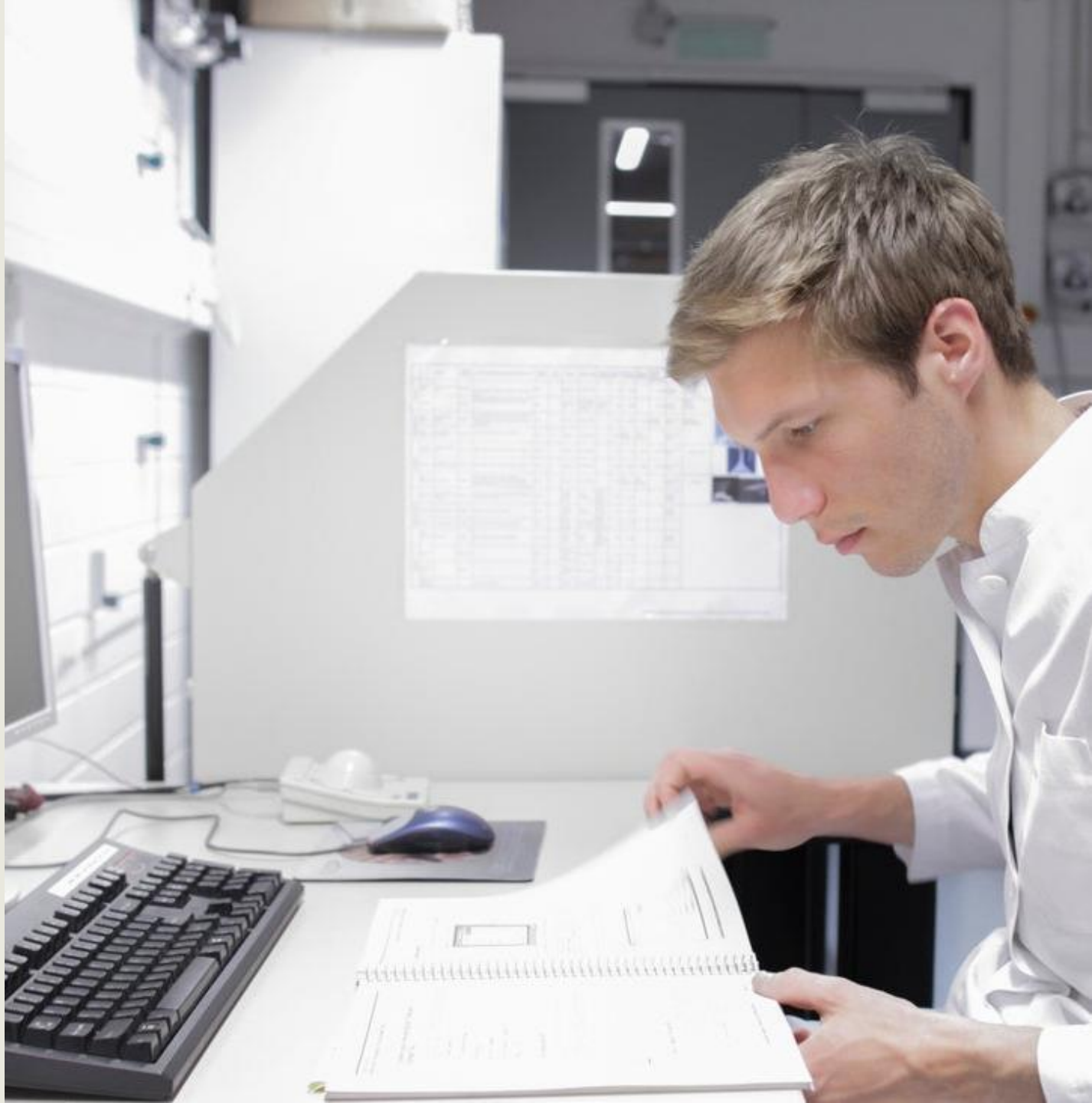


KEY PRACTICES FOR ENSURING MODEL ACCURACY



F1 SCORES

- **The F1 score** is a crucial evaluation metric in ML that balances precision and recall to provide a single measure of a model's performance
- **It is particularly useful in scenarios with imbalanced datasets or when both false positives and false negatives arise**
- By leveraging the F1 score, ML models are evaluated comprehensively, maintaining high performance and fairness throughout their lifecycle



F1 SCORE EXAMPLE

- **Imagine** that you are creating a model to detect a rare disease
- If you focus solely on accuracy, you might believe your model is performing well because it correctly identifies most healthy individuals
- But what about the few instances where it fails to detect the disease?
- **The F1 score becomes essential here as it offers a balanced perspective, ensuring your model is not only accurate but also precise and sensitive enough to identify the critical cases**

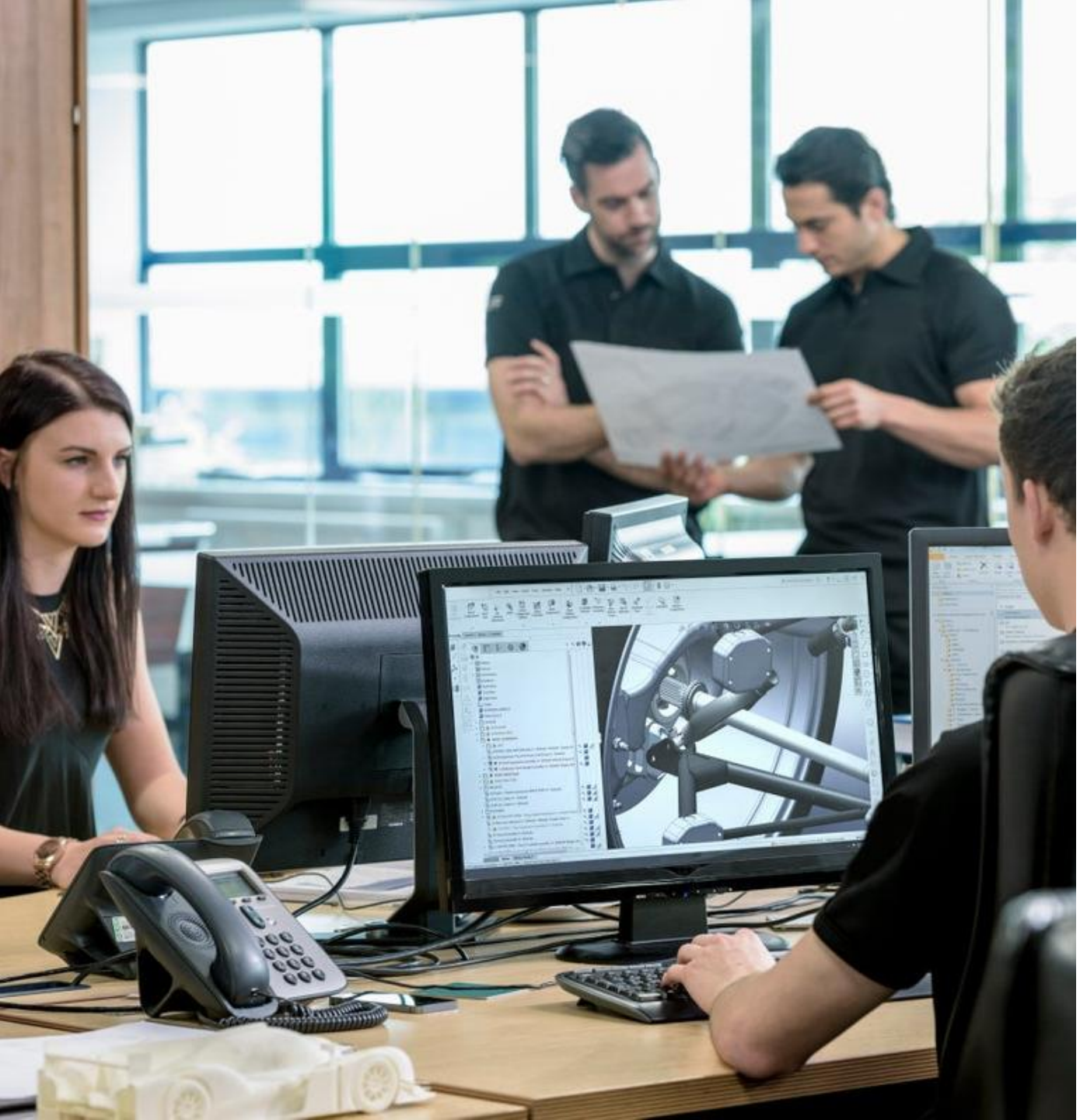
F1 SCORES

The F1 score is the harmonic mean of precision and recall
It is calculated using the formula:

$$F1 = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

Precision: the ratio of correctly predicted positive instances to the total predicted positive instances

Recall: the ratio of correctly predicted positive instances to the total actual positive instances



USES OF F1 SCORES IN MLOPS

- **Model evaluation:** the score is used to evaluate the performance of ML models, especially in classification
  - It provides a balanced measure that considers both precision and recall, making it ideal for assessing models with imbalanced classes
- **Monitoring and maintenance:** in MLOps, continuous monitoring of the F1 score helps in detecting performance degradation over time

USES OF F1 SCORES IN MLOPS

- **Bias detection**: the F1 score can be used in conjunction with other metrics to detect and mitigate bias in ML models
- **Amazon SageMaker Clarify** provides tools to analyze the F1 score across different subgroups to ensure fairness and transparency

MODEL PERFORMANCE: AREA UNDER THE ROC CURVE (AUC)

- The **area under the curve (AUC)** is a key metric used to evaluate the performance of ML models, particularly in binary classification tasks
- The AUC is derived from the Receiver operating characteristic (ROC) curve, which plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings

KEY ASPECTS OF AUC IN MLOPS MODEL PERFORMANCE

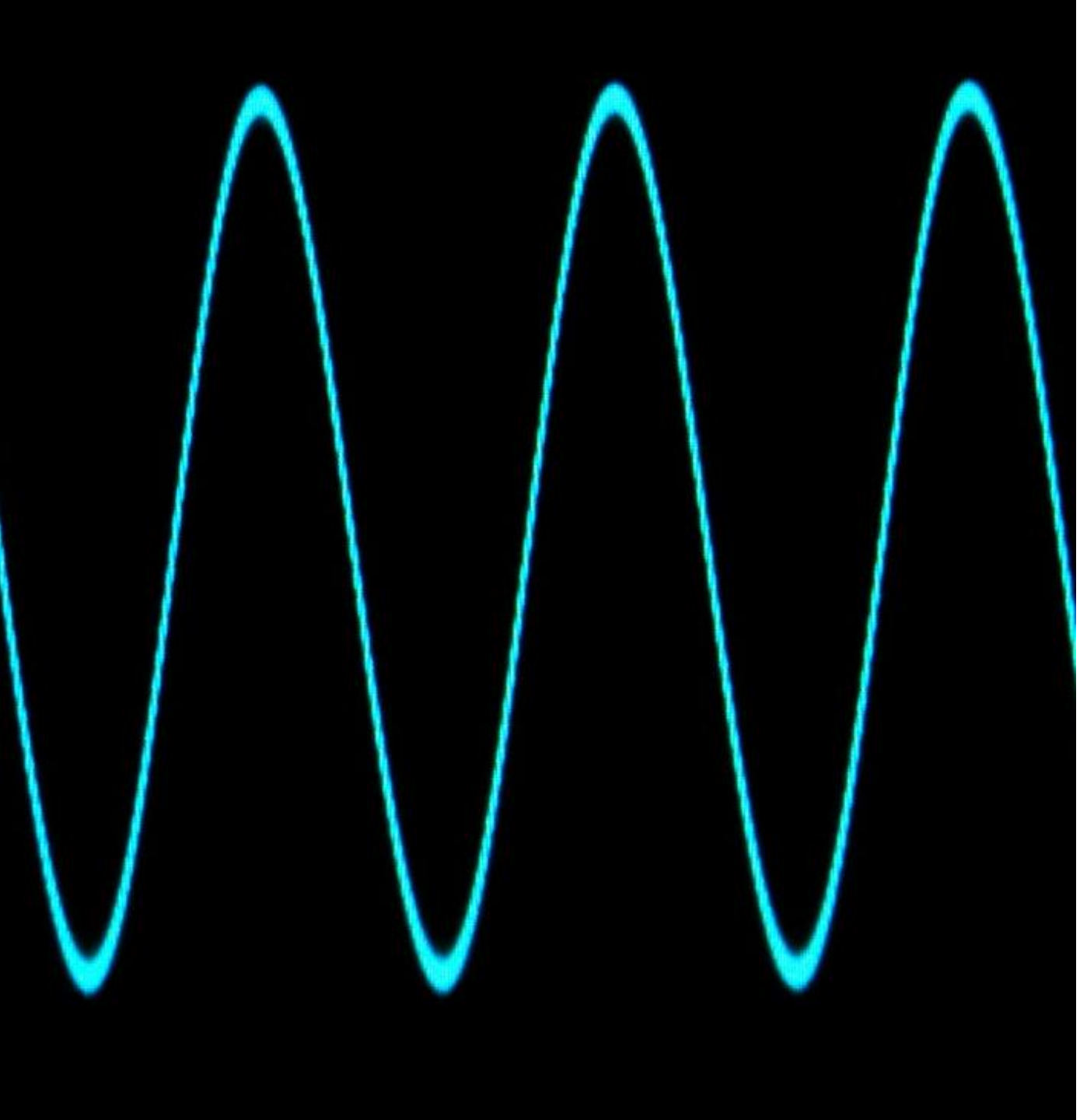
- **Interpretation:** the AUC value ranges from 0 to 1
- An AUC of 1 indicates perfect model performance, where the model correctly classifies all positive and negative instances
- An AUC of 0.5 suggests that the model's performance is no better than random guessing

KEY ASPECTS OF AUC IN MLOPS MODEL PERFORMANCE

- **Aggregate performance measure:** AUC provides an aggregate measure of model performance across all possible classification thresholds
  - Unlike metrics such as accuracy, precision, or F1-score, which depend on a specific threshold, AUC evaluates the model's ability to distinguish between classes over a range of thresholds

KEY ASPECTS OF AUC IN MLOPS MODEL PERFORMANCE

- **Use in imbalanced datasets**: AUC is particularly useful in scenarios with imbalanced datasets, where the number of positive and negative instances is not equal
- It helps in understanding the trade-offs between true positives and false positives across different decision thresholds



KEY ASPECTS OF AUC IN MLOPS MODEL PERFORMANCE

- **Model comparison:** AUC is often used to compare the performance of different models
- By evaluating the AUC scores, data scientists can determine which model performs better in distinguishing between classes
- Continuous monitoring of the AUC score helps in detecting performance degradation over time

BUSINESS METRIC: COST PER USER

- The **cost per user** business metric is used in MLOps to measure the efficiency and cost-effectiveness of ML operations
- This metric helps organizations understand the cost associated with serving each user or customer, which is crucial for budgeting, pricing, and optimizing resource allocation

KEY ASPECTS OF COST PER USER IN MLOPS

- **Resource allocation:** by calculating the cost per user, organizations can allocate resources more effectively
  - This includes optimizing the use of compute instances, storage, and other AWS services to ensure that costs are kept in check while maintaining performance
- **Pricing strategies:** the cost per user metric can inform pricing strategies for ML services

KEY ASPECTS OF COST PER USER IN MLOPS

- **Budgeting and forecasting:** understanding the cost per user helps in budgeting and forecasting expenses
  - Organizations can predict future costs based on user growth and adjust their budgets accordingly
- **Cost optimization:** AWS provides tools like AWS Cost Explorer and AWS Budgets to track and manage costs

COST PER USER IN MLOPS

- **Performance monitoring:** monitoring the cost per user alongside performance metrics ensures that cost optimizations do not negatively impact the quality of service
  - This balance is crucial for maintaining user satisfaction while managing expenses
- **Scalability:** as the number of users grows, the cost per user metric helps ensure that the ML infrastructure scales efficiently

BUSINESS METRIC: DEVELOPMENT COSTS

- The **development cost** business MLOps metric relates to expenses throughout the entire ML lifecycle, from data collection and preprocessing to model training, deployment, and monitoring
- This metric helps organizations understand and manage the financial investment required to develop and maintain ML models
- By understanding this metric, organizations can optimize their ML operations, ensuring cost-effectiveness while maintaining high performance

KEY ASPECTS OF DEVELOPMENT COST

- **Resource utilization:** includes the cost of compute instances, storage, and other AWS services used during the ML lifecycle
  - AWS provides tools like AWS Cost Explorer to track and manage these expenses
- **Data preparation:** costs associated with data collection, cleaning, and preprocessing, including services like AWS Glue for data integration and Amazon S3 for data storage
- **Model training:** expenses related to training ML models, including the use of Amazon SageMaker

KEY ASPECTS OF DEVELOPMENT COST

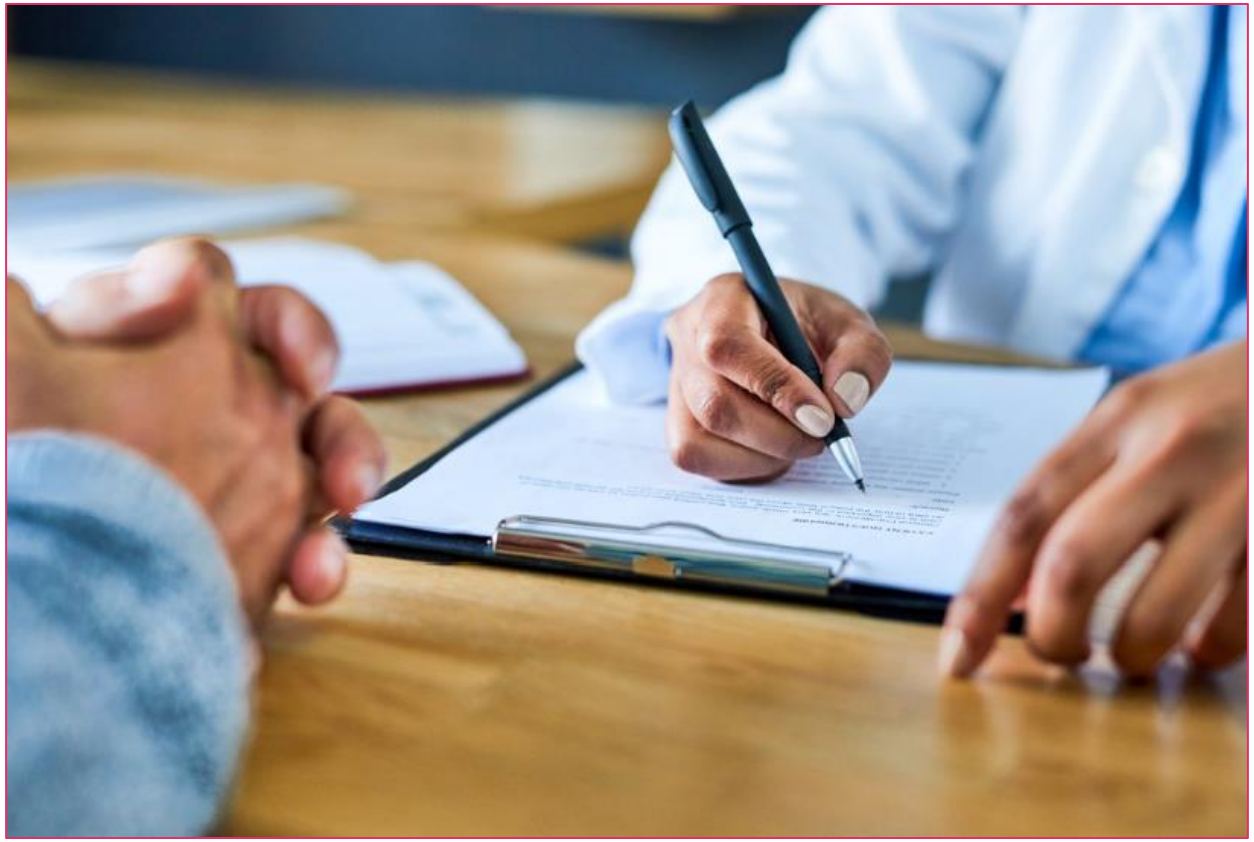
- **Model deployment:** costs incurred during the deployment of models, such as setting up and maintaining endpoints for real-time inference using Amazon SageMaker
- **Monitoring and maintenance:** ongoing costs for monitoring model performance and maintaining the ML infrastructure
  - Includes the use of SageMaker Model Monitor for continuous monitoring and CloudWatch for logging and alerts

KEY ASPECTS OF DEVELOPMENT COST

- **Automation and CI/CD:** expenses related to automating the ML pipeline and integrating with CI/CD workflows
  - AWS CodePipeline and AWS CodeBuild are commonly used for this purpose
- **Scalability:** ensuring that the ML infrastructure can scale efficiently to handle increasing workloads without a proportional increase in costs
  - AWS services are designed to scale automatically based on demand

BUSINESS METRIC: CUSTOMER FEEDBACK

- The **customer feedback** business metric in MLOps is crucial for understanding how well ML models are meeting user needs and expectations
- This metric involves collecting and analyzing feedback from users to improve model performance and ensure that the models deliver value to the business



KEY ASPECTS OF CUSTOMER FEEDBACK IN MLOPS

- **Feedback collection:** gathering feedback from users through various channels such as surveys, user reviews, and direct interactions
  - This feedback provides insights into how users perceive the model's performance and its impact on their experience
- **Sentiment analysis:** analyzing the sentiment of the feedback to understand user satisfaction and identify areas for improvement

KEY ASPECTS OF CUSTOMER FEEDBACK

- **Model performance evaluation:** using customer feedback to evaluate the performance of ML models
  - This includes assessing the accuracy, relevance, and usability of the model's predictions based on user input
- **Iterative improvement:** incorporating customer feedback into the model development process to make iterative improvements

KEY ASPECTS OF CUSTOMER FEEDBACK

- **User engagement:** engaging with users to gather detailed feedback and understand requirements
  - This helps in building models that are more aligned with user needs and provide greater value
- **Monitoring and reporting:** continuously monitoring customer feedback and generating reports to track trends and issues
  - This helps in making data-driven decisions to enhance model performance and user satisfaction

BUSINESS METRIC: RETURN ON INVESTMENT (ROI)

- The return on investment (ROI) business metric in MLOps is used to measure the financial benefits gained from ML operations relative to the costs incurred
- By leveraging the ROI metric, AWS ensures that organizations can assess the financial impact of their ML operations, make informed decisions, and optimize their investments in ML technologies

KEY ASPECTS OF ROI IN MLOPS

- **Cost savings:** ROI considers the cost savings achieved through automation, improved efficiency, and reduced manual intervention in the ML lifecycle
- **Revenue generation:** ROI also accounts for the added revenue from deploying ML models that enhance business processes, improve customer experiences, and create new revenue streams

KEY ASPECTS OF ROI IN MLOPS

- **Time to value:** the speed at which ML models can be developed, deployed, and deliver value (a critical factor in ROI)
- **Scalability**: the ability to scale ML operations efficiently without a proportional increase in costs contributes to a higher ROI
  - AWS services are designed to scale automatically based on demand, ensuring that ML models can handle increasing workloads cost-effectively

KEY ASPECTS OF ROI IN MLOPS

- **Performance improvements:** enhancements in model performance, such as increased accuracy and reduced latency, can lead to better business outcomes and higher ROI
- **Risk mitigation:** effective MLOps practices help mitigate risks associated with model deployment and operation, such as data drift, model degradation, and compliance issues
    - Reducing these risks can prevent costly errors and enhance ROI

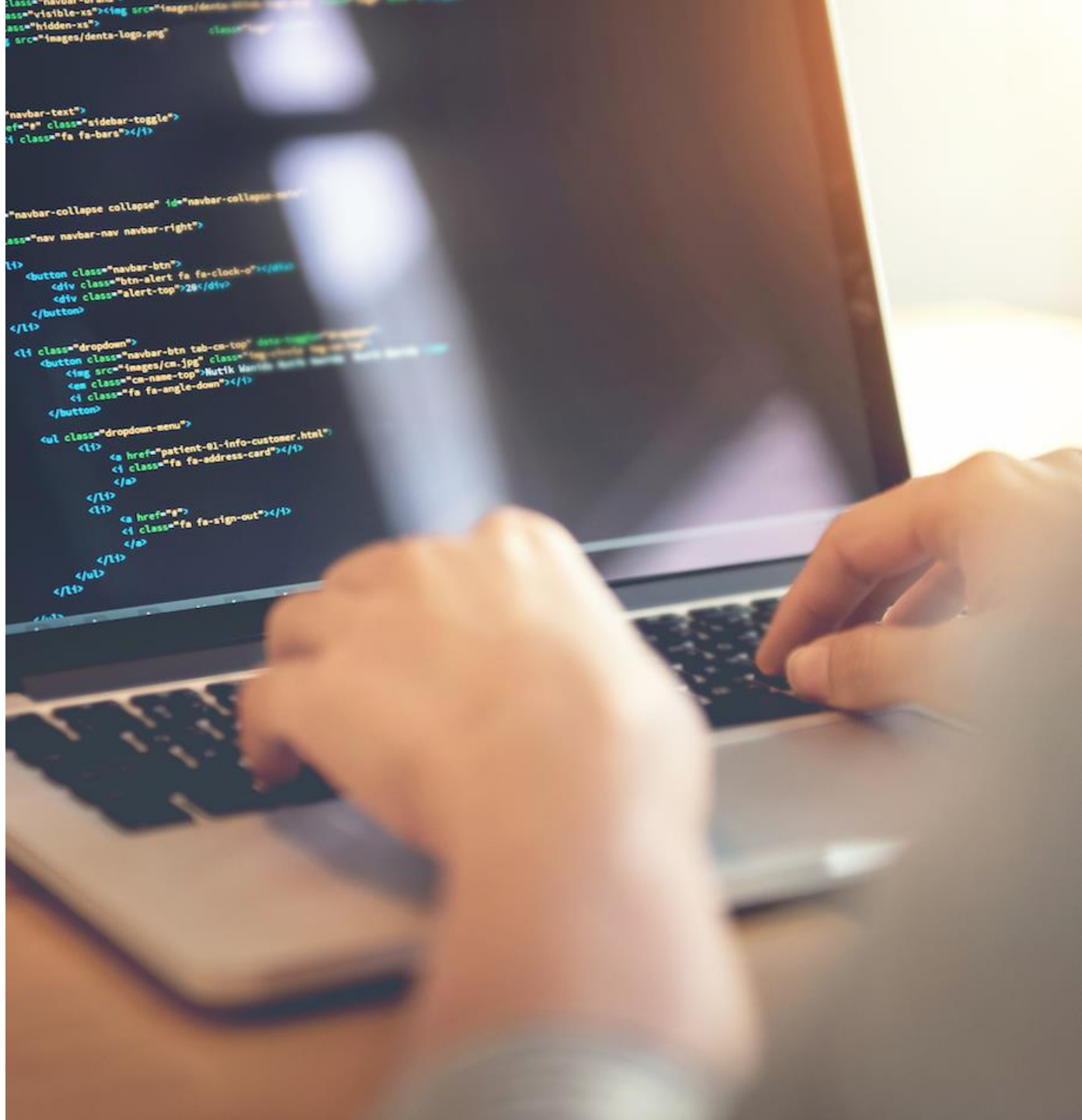
BASIC CONCEPTS OF GENERATIVE AI

Objectives

- Describe foundational generative AI concepts
- Compare foundational generative AI models
- Identify potential use cases for generative AI
- Explore the foundation model lifecycle stages

FOUNDATIONAL GENERATIVE AI CONCEPTS

- AWS describes **generative AI** as a type of artificial intelligence that can create new content and ideas, such as images, videos, conversations, stories, and music
- It can learn human language, programming languages, art, chemistry, biology, or any complex subject matter, and reuse what it knows to solve new problems

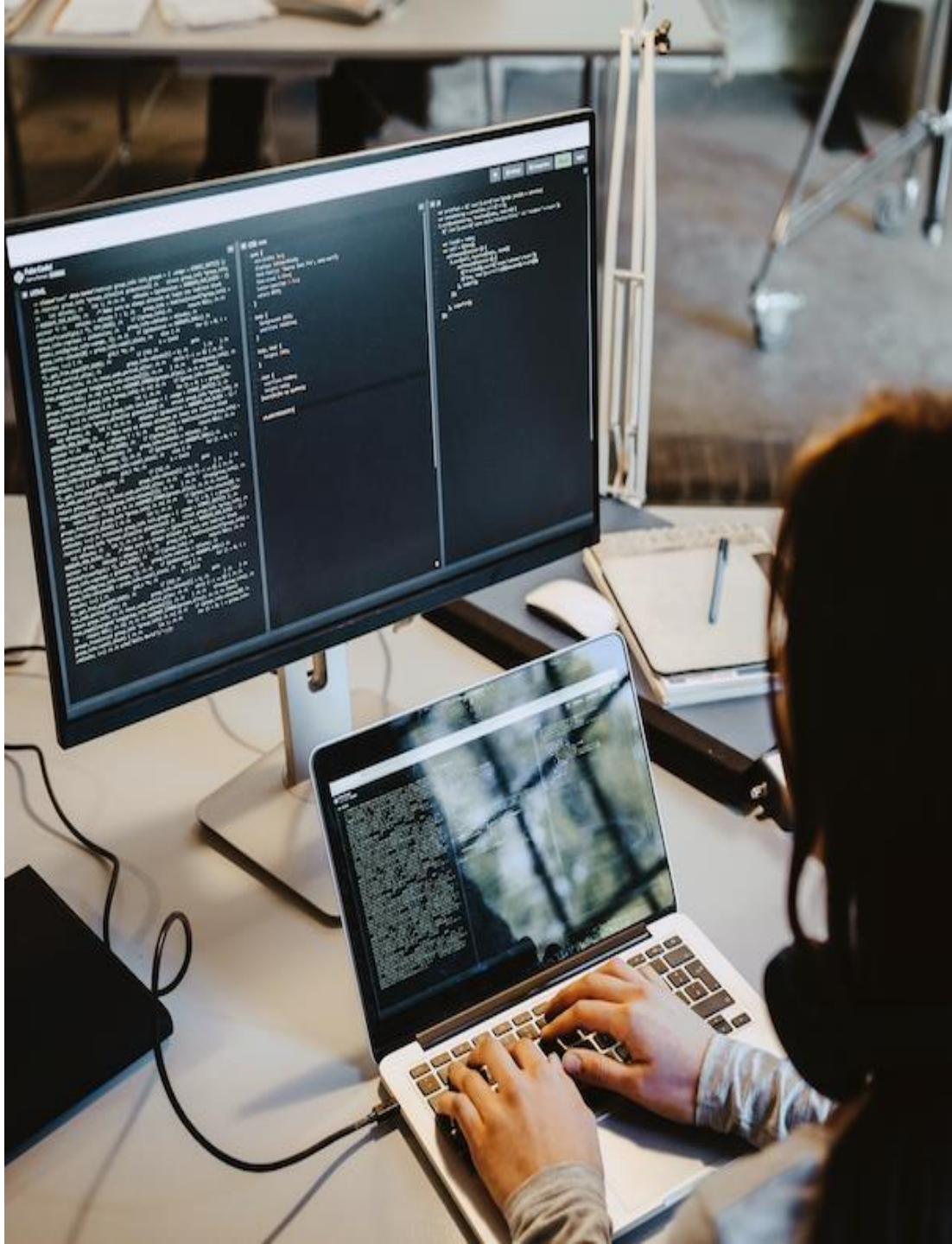


GENERATIVE AI TOKENS

- Generative AI **tokens** are defined as the units of text that a generative AI model processes and generates
- These tokens can be words, parts of words, or even characters, depending on the model's design
- The model uses tokens to understand and generate human-like text, enabling it to create new content, such as conversations, stories, and code

GENERATIVE AI CHUNKING

- **Chunking** in the context of generative AI is the process of dividing large documents or datasets into smaller, manageable sections
  - This is crucial for efficient data processing and retrieval
- Chunking strategies can vary, including fixed-size chunking, which splits content into chunks of a specified number of tokens, and hierarchical chunking, which organizes information into nested structures of parent and child chunks



GENERATIVE AI EMBEDDINGS

- Generative AI **embeddings** are numerical representations of real-world objects that ML and AI systems use to understand complex knowledge domains
- These embeddings capture the inherent properties and relationships between data points, enabling AI models to process and generate new content effectively

GENERATIVE AI VECTORS

- AI **vectors** are numerical representations of data that capture the relationships and properties of the data points
- These vectors are used by models to process and generate new content effectively
- They are essential for understanding and manipulating complex data structures in generative AI applications

GENERATIVE AI PROMPTS

- AWS defines generative AI **prompts** as natural language texts that request the generative AI to perform specific tasks
- These prompts guide the AI to generate desired outputs by providing detailed instructions
- The process of creating and refining these prompts to achieve high-quality and relevant results is known as prompt engineering

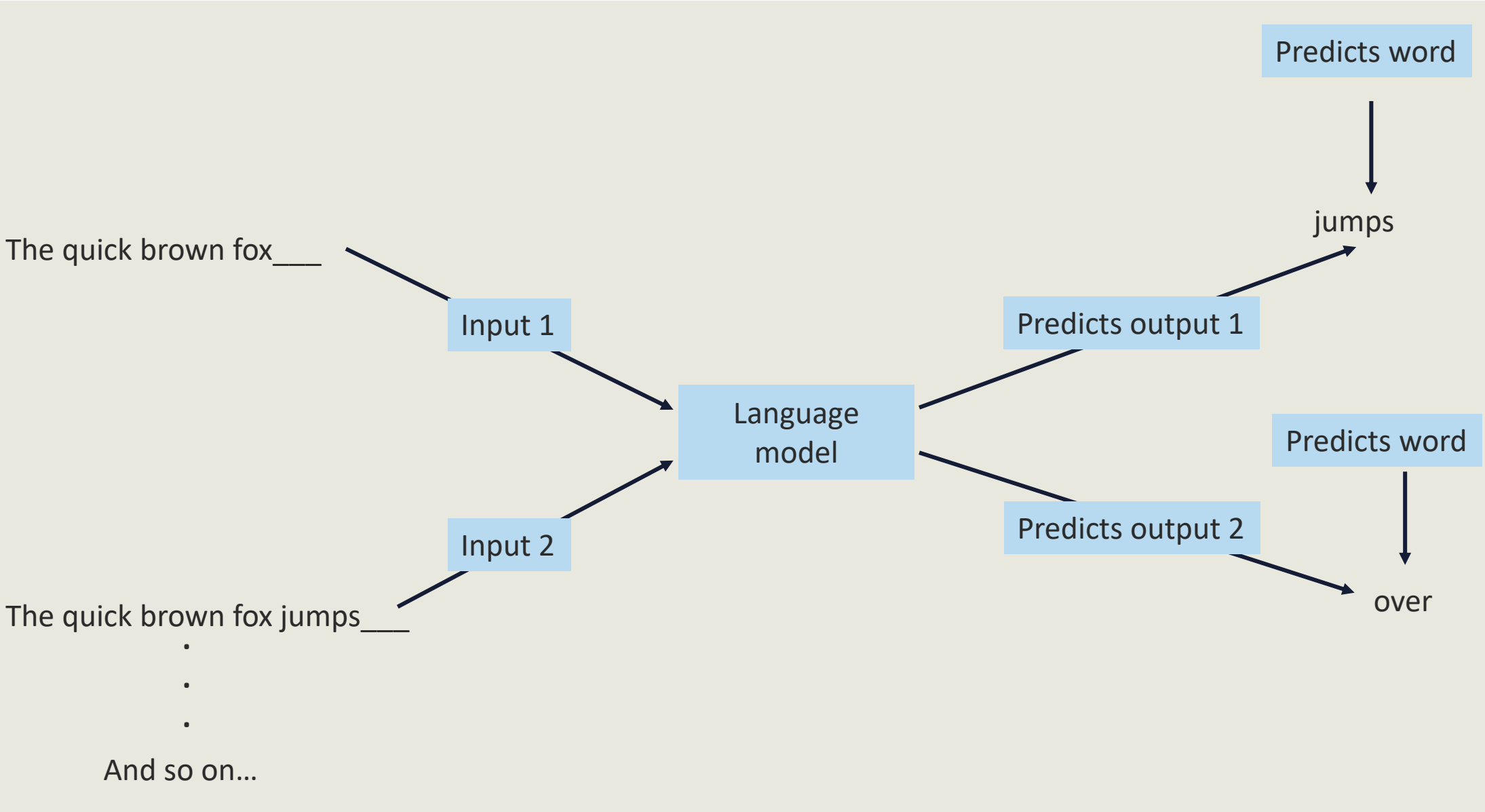
GEN-AI PROMPT ENGINEERING

- **Prompt engineering** is the process of guiding generative AI solutions to generate desired outputs by providing detailed instructions
- This involves choosing the most appropriate formats, phrases, words, and symbols to interact with users meaningfully

Transformer-Based Large Language Models (LLMs)

- **A transformer-based large language model (LLM)** is a type of neural network architecture that processes and generates text by understanding the relationships between words and phrases in a sequence
- These models consist of an encoder and a decoder with self-attention capabilities, allowing them to handle long-range dependencies in text and process entire sequences in parallel

Large Language Models (LLMs)



Foundational Generative AI Models

- **Foundation models (FMs)** are large deep learning neural networks trained on massive datasets
- These models have revolutionized the way data scientists approach machine learning
- Instead of developing AI from scratch, data scientists use foundation models as a starting point to develop ML models more quickly and cost-effectively

Multi-Modal Models

- A **multi-modal model** is an AI system designed to handle and integrate data from various modalities, such as text, images, audio, and video
- These models can understand and analyze different forms of data inputs to achieve a more comprehensive understanding and generate more robust outputs

Diffusion Models

- A **diffusion** model is a type of generative AI model that produces unique photorealistic images from text and image prompts
- These models work by using Gaussian noise to encode an image and then applying a noise predictor along with a reverse diffusion process to recreate the image

Diffusion Models

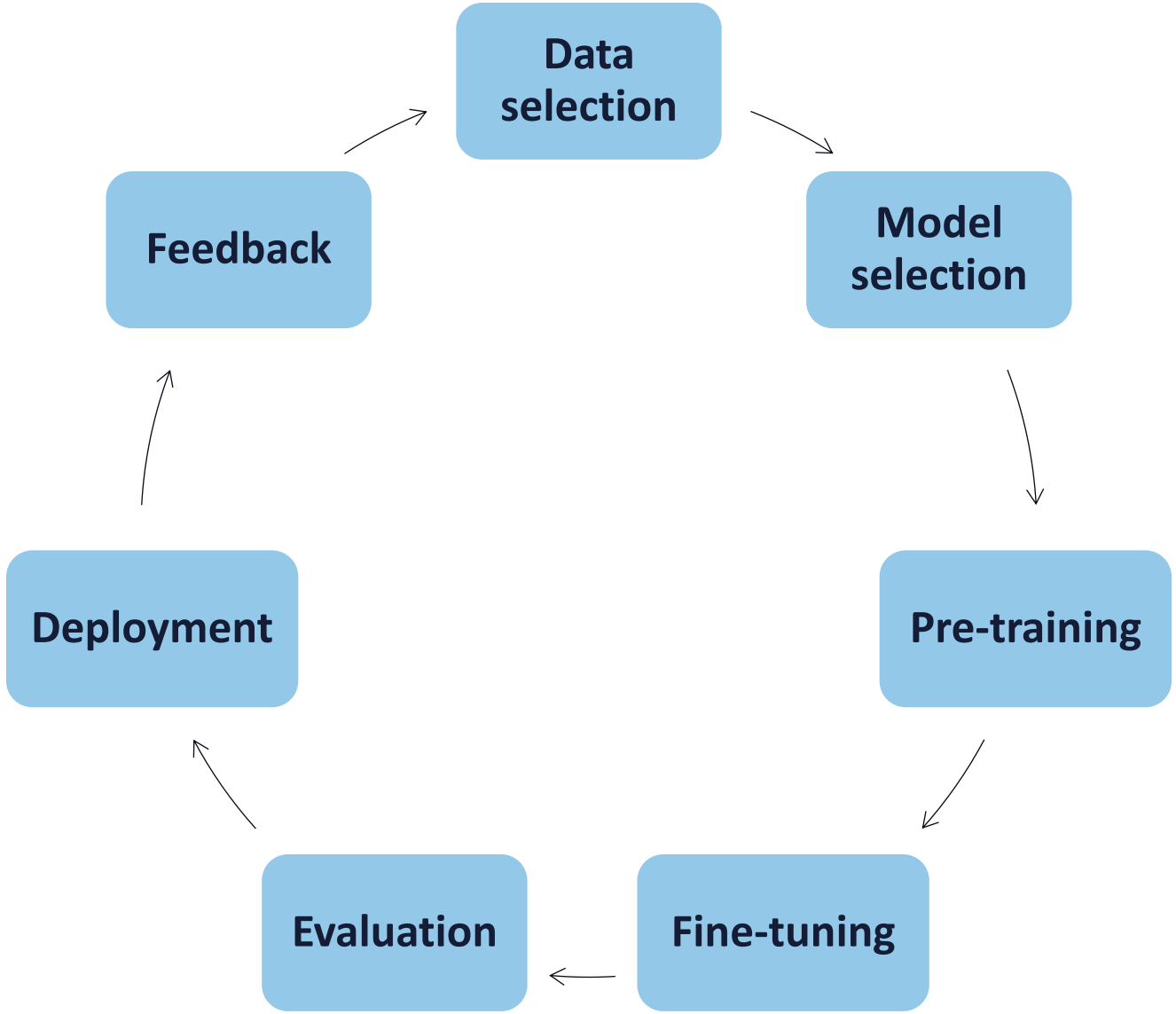
- **Gaussian noise**, also known as white noise, is a type of random noise that follows a normal distribution
- In the context of generative AI, Gaussian noise is often added to input data during the training process
  - This helps the model learn to generate new data that is like the training data, even when the input is not perfect
  - This process allows the model to generate realistic images by gradually refining the noise into a coherent picture

Exploring Potential Use Cases for Generative AI Models

In this demo…

We will explore the capabilities of generative AI models such as image, video, and audio generation; summarization; chatbots; translation; code generation; customer service agents; search; and recommendation engines.

The Foundation Model Lifecycle



The Foundation Model Lifecycle: Data Selection

- The **data selection stage** in the foundation model lifecycle involves choosing the right datasets to train the foundation model
- This stage is crucial because the quality and diversity of the data directly impact the model's performance and generalization capabilities

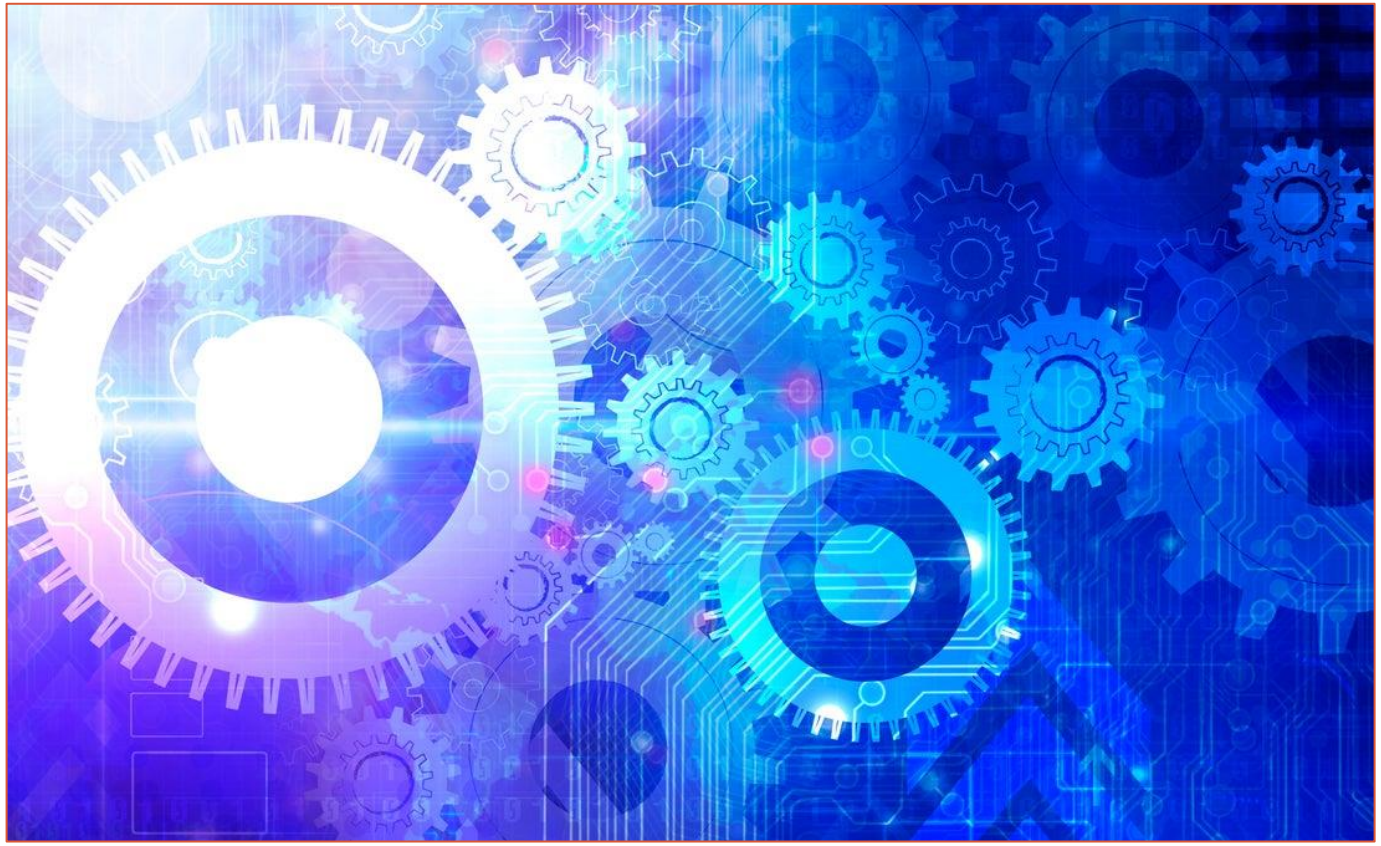
Common Data Selection Stage Techniques

1. Data sourcing
2. Data cleaning
3. Data filtering
4. Data augmentation
5. Data annotation
6. Data balancing

The Foundation Model Lifecycle: Model Selection

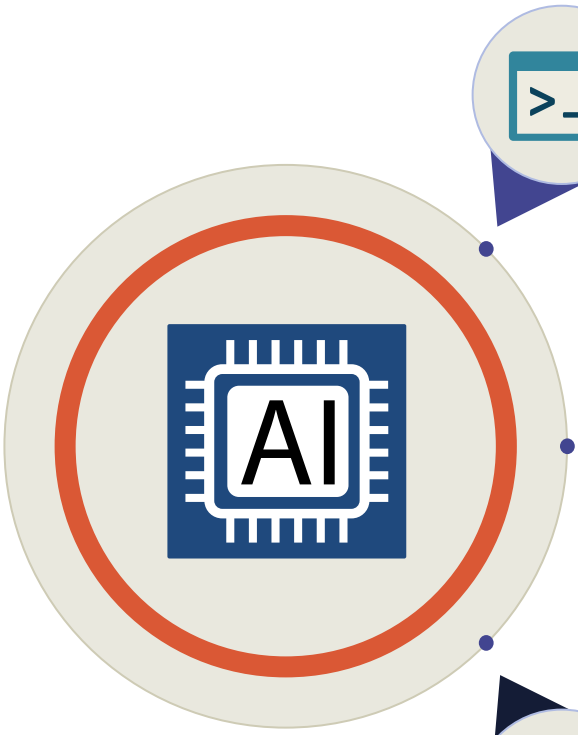
- The **model selection stage** of the foundation model lifecycle involves choosing the most appropriate foundation model for a specific use case
- This stage is critical because the selected model will significantly impact the performance, efficiency, and overall success of the AI application

The Foundation Model Lifecycle: Model Selection



- Factors include the model's customization capabilities, size, latency, licensing agreements, inference options, and context windows
  - **Inference options** refer to the different configurations and settings available for running inference tasks with foundation models, such as latency-optimized inference
  - The **context window** (or attention window) refers to the maximum number of tokens from the input that the model can consider at one time when making predictions

Some Popular Available Models

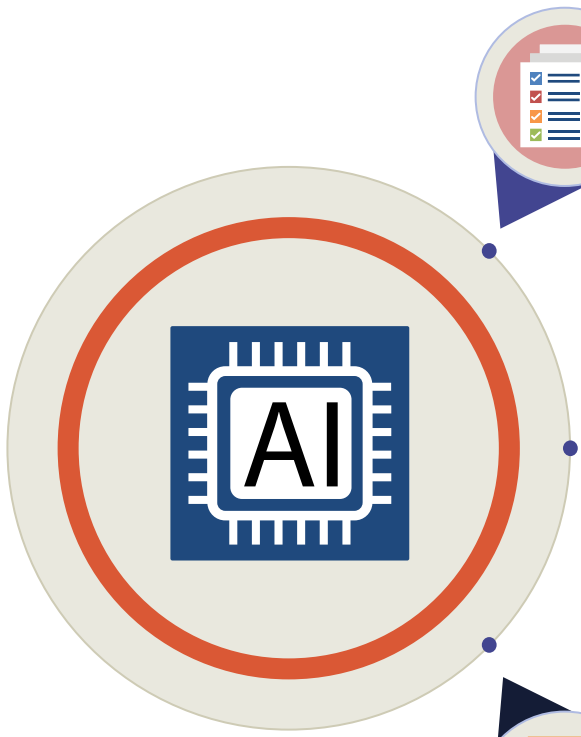


AI21 Labs' Jurassic: known for its language generation capabilities

Anthropic's Claude: focuses on safety and alignment in AI

Cohere's Command and Embed: offers models for text generation and embeddings

Some Popular Available Models



Meta's Llama 2: provides models for various AI tasks

Stability AI's Stable Diffusion: specializes in image generation

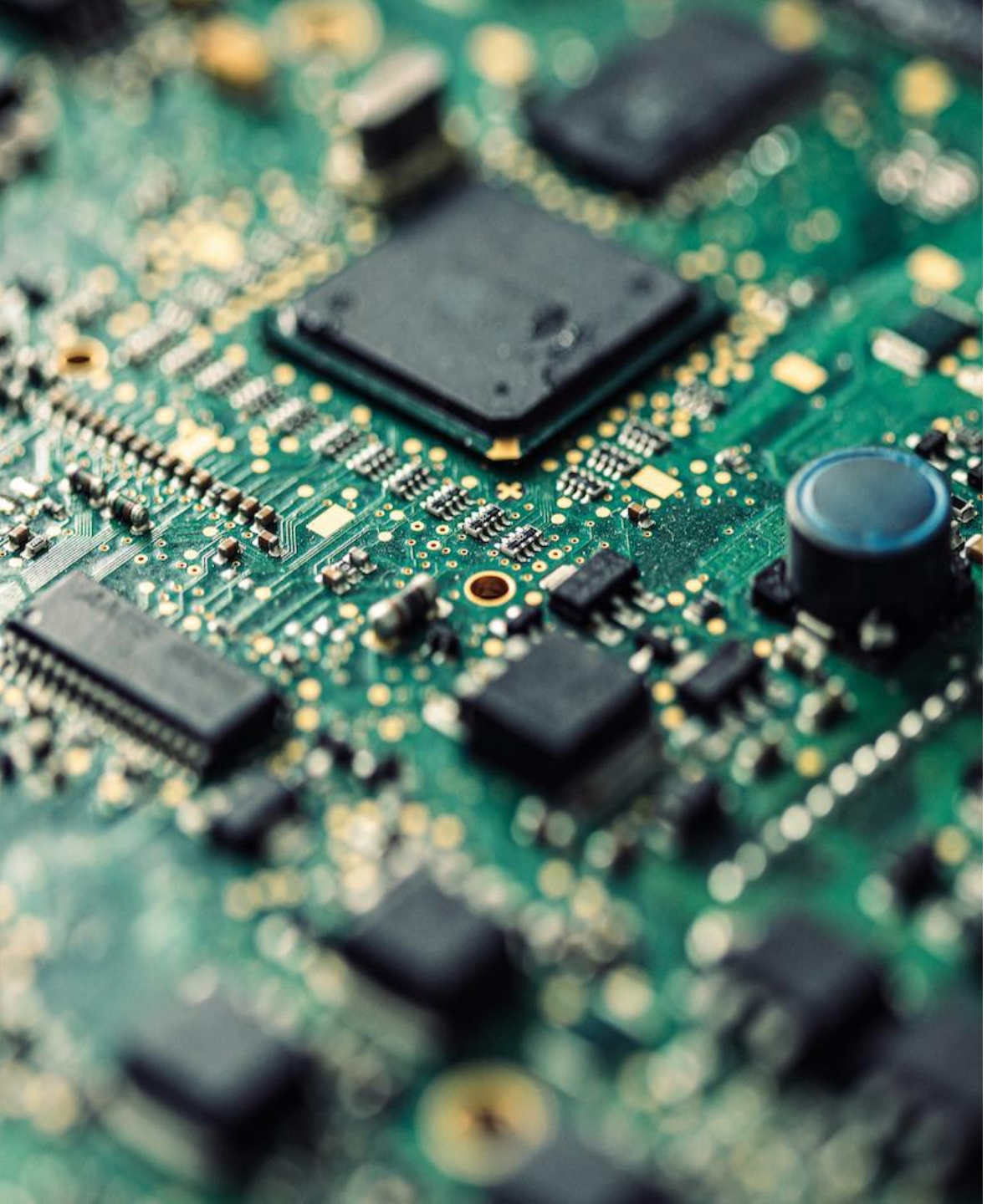
Amazon Titan models: developed by AWS for a range of applications

The Foundation Model Lifecycle: Pre-Training

- The **pre-training stage** of the foundation model lifecycle involves preparing the foundation model for training by selecting and curating the appropriate datasets
- This stage is crucial as it ensures the model is exposed to diverse and high-quality data, which is essential for effective learning and generalization

The Foundation Model Lifecycle: Fine-Tuning

- The **fine-tuning stage** of the foundation model lifecycle involves further training of a pre-trained foundation model on a specific task or domain using a smaller, relevant dataset
- This process helps the model better understand and generate text tailored to the particular context, resulting in improved performance and accuracy



Fine-Tuning Techniques

- **Transfer learning:** leveraging pre-trained models and fine-tuning them on a smaller, task-specific dataset to improve performance and reduce training time
- **Hyperparameter tuning:** optimizing the model's hyperparameters using techniques like grid search, random search, or Bayesian optimization to enhance performance
- **Data augmentation:** enhancing the training dataset with additional variations to improve the model's robustness and generalization capabilities

Fine-Tuning Techniques

- **Domain adaptation:** adjusting the model to better handle data from a specific domain by incorporating domain-specific knowledge and data
- **Regularization techniques:** applying methods like dropout, weight decay, and early stopping to prevent overfitting and improve the model's generalization

The Foundation Model Lifecycle: Evaluation

- The **evaluation stage** of the foundation model lifecycle involves assessing the performance and effectiveness of foundation models
- This stage is critical for ensuring that the models meet the desired standards and can perform well in real-world applications

Common Evaluation Types

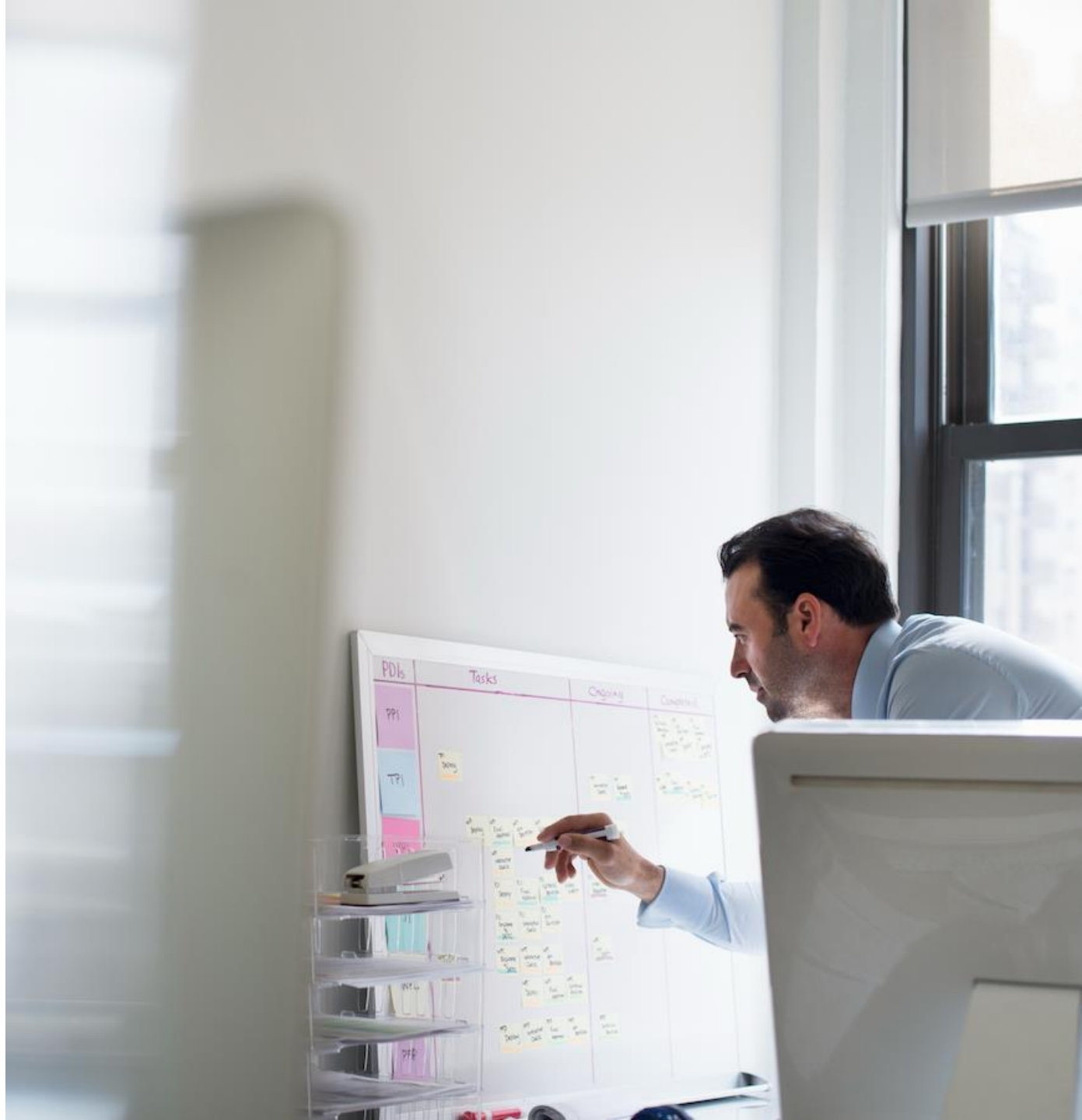
- **Automatic evaluations:** these involve running predefined tests and metrics to generate calculated scores that help assess the model's performance
  - This can include tasks like text generation, classification, question answering, and summarization
- **Human-based evaluations:** these involve a team of human evaluators who provide ratings and preferences based on specific metrics
  - This approach helps capture subjective aspects of model performance that automatic evaluations might miss

Common Evaluation Types

- **Task-specific evaluations:** evaluating the model's performance on specific tasks relevant to its intended use case, such as language understanding, image generation, or other domain-specific tasks
- **Robustness testing:** assessing the model's ability to handle noisy or adversarial inputs to ensure it performs well under various conditions
- **Bias and fairness assessment:** evaluating the model for potential biases and ensuring it performs fairly across different demographic groups

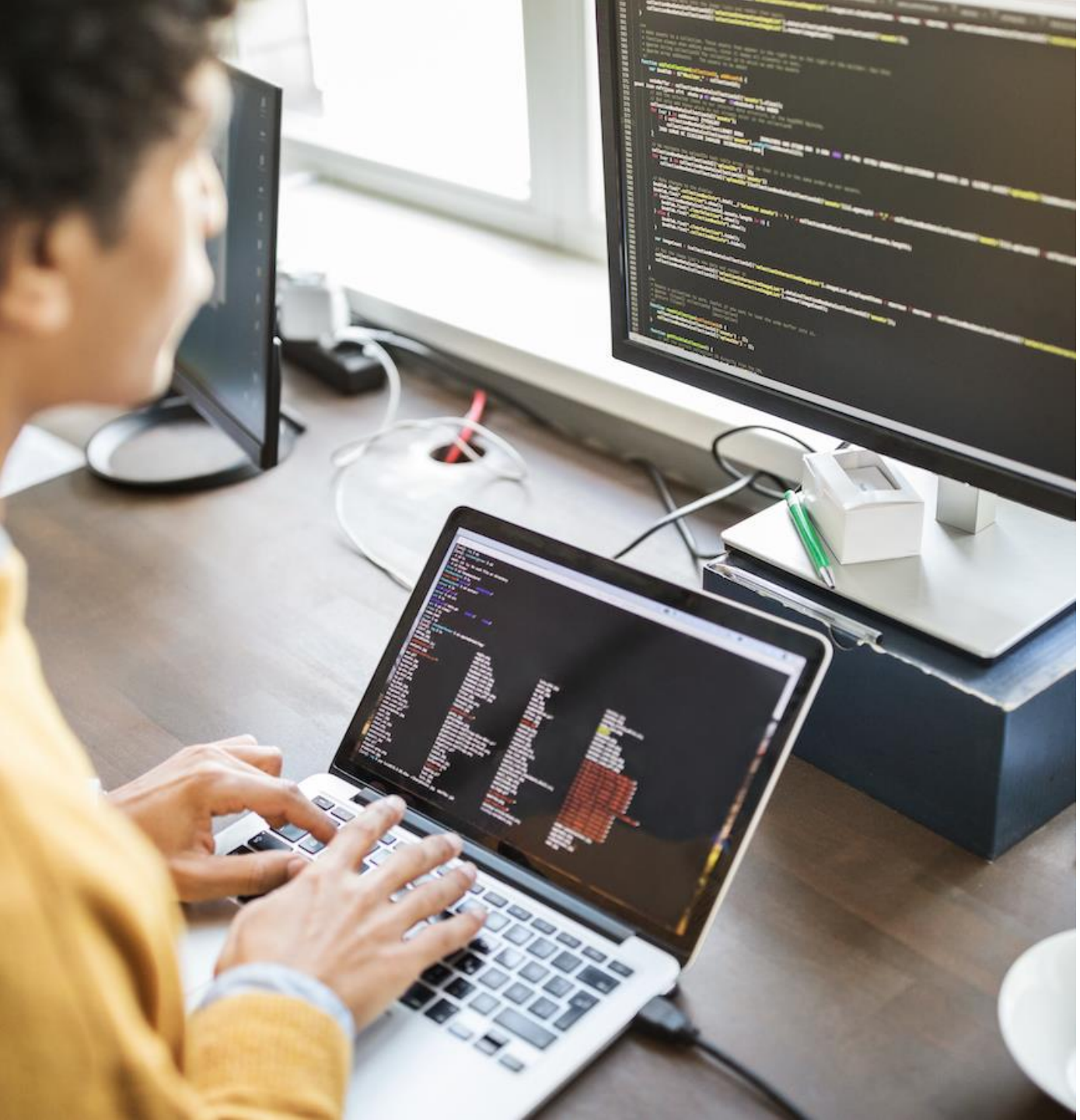
The Foundation Model Lifecycle: Deployment

- The **deployment stage** of the foundation model lifecycle involves making the foundation model available for use in production environments
- This stage includes setting up the necessary infrastructure, configuring the model for optimal performance, and ensuring that it can handle real-world data and workloads



Deployment Stage Tools and Techniques

- **AWS Lambda:** this serverless compute service lets you run code without provisioning or managing servers, making it easier to deploy and scale models
  - **AWS Step Functions:** this orchestration service lets you combine AWS Lambda functions and other AWS services to build and run applications
- **Amazon ECS and EKS:** these managed container services allow you to run and scale containerized applications, providing a flexible and scalable environment for deploying models



Deployment Stage Tools and Techniques

- **AWS CodePipeline:** this continuous integration and continuous delivery (CI/CD) service offers fast and reliable application and infrastructure updates
- **Amazon CloudWatch:** this monitoring and observability service provides data and actionable insights to monitor applications, respond to system-wide performance changes, and optimize resource utilization

The Foundation Model Lifecycle: Feedback

- The **feedback stage** of the foundation model lifecycle involves collecting and analyzing user feedback to improve the foundation model's performance and usability
- This stage is crucial for identifying areas where the model can be enhanced and ensuring it meets user needs effectively

Common Feedback Stage Metrics

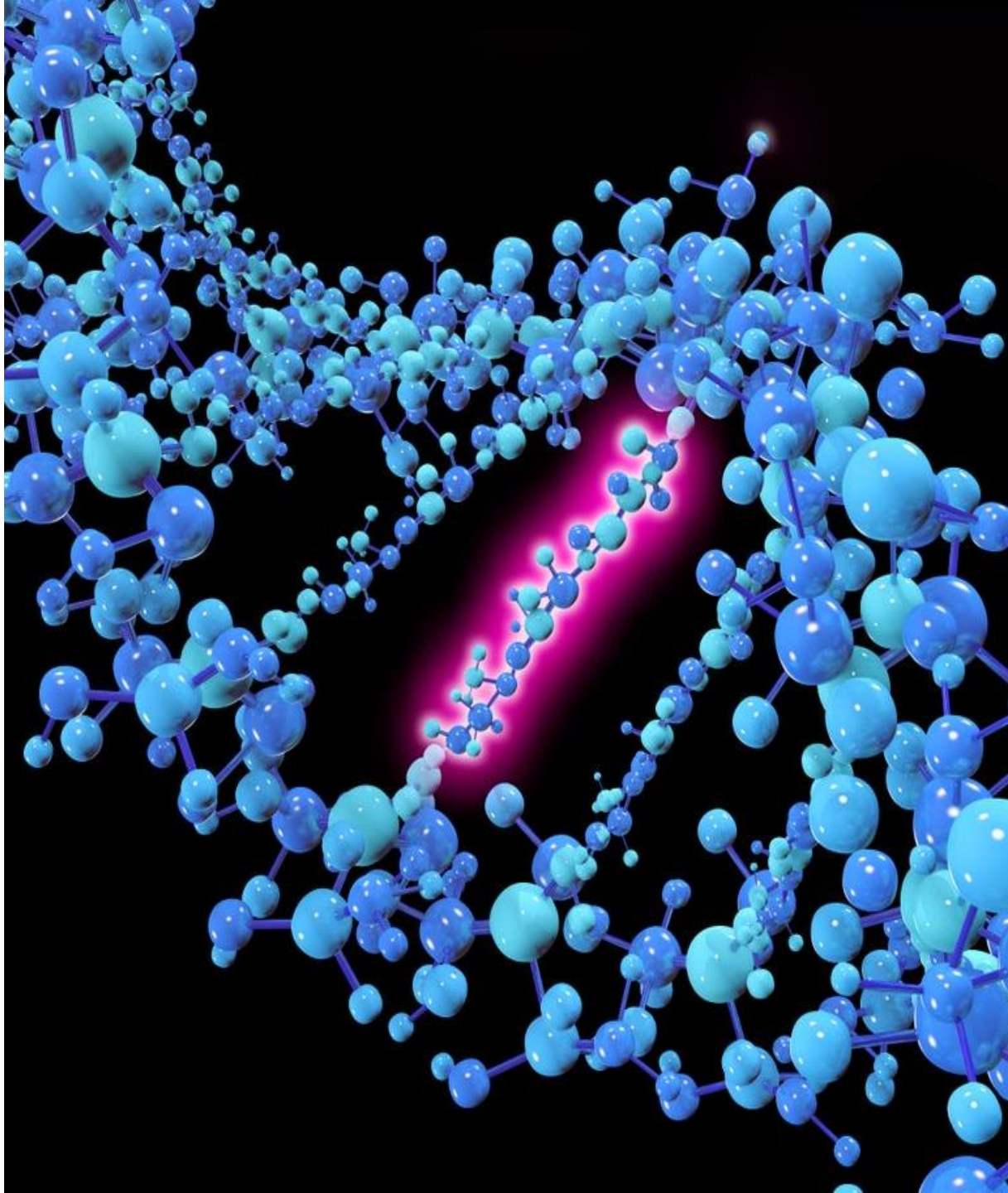
- **Accuracy**: measures how often the model's predictions are correct
- **Precision**: the ratio of true positive predictions to the total number of positive predictions made by the model
- **Recall**: the ratio of true positive predictions to the total number of actual positive instances in the dataset

Common Feedback Stage Metrics

- **F1 score:** the harmonic mean of precision and recall, providing a balance between the two
- **AUC-ROC (area under the receiver operating characteristic curve):** evaluates the model's ability to distinguish between classes
- **Mean squared error (MSE):** measures the average squared difference between the predicted and actual values, commonly used for regression tasks

Common Feedback Stage Metrics

- **Confusion matrix:** a table used to describe the performance of a classification model by showing the true positives, false positives, true negatives, and false negatives
- **Log loss:** measures the performance of a classification model where the output is a probability value between 0 and 1
- **Perplexity:** used for evaluating language models, it measures how well a probability distribution or model predicts a sample



Capabilities and Limitations of Generative AI

Objectives

- Compare the generative AI advantages of foundational generative AI like adaptability, responsiveness, and simplicity
- Compare the generative AI disadvantages such as hallucinations, interpretability, inaccuracy, and nondeterminism
- Select the appropriate generative AI models
- Explore values and metrics for generative AI

Advantages of Generative AI: Adaptability

- **Adaptability** refers to the ability to adjust to new conditions, environments, or situations
- It involves being flexible and open to change, allowing individuals or systems to effectively respond to challenges and opportunities
- Adaptability is a crucial trait for success in various aspects of life, including personal growth, professional development, and technological advancements

Advantages of Generative AI: Responsiveness

- **Responsiveness** broadly refers to the ability to react quickly and effectively to changes, requests, or stimuli
- It involves being attentive and prompt in addressing needs, concerns, or situations, ensuring timely and appropriate actions
- This trait is essential in various contexts, such as customer service, healthcare, and technology, where quick and efficient responses can significantly impact outcomes

Simplicity Is Crucial in Information Technology

Ease of use	Maintenance	Security	Scalability
Simple systems are easier for users to understand and operate, reducing the learning curve and increasing productivity	Simple systems are easier to maintain, troubleshoot, and update, leading to lower maintenance costs and fewer errors	Complex systems are often more vulnerable, so simplicity helps by reducing potential points of failure	Simple designs are easier to scale and adapt to changing needs, ensuring that the system can grow with the organization

Disadvantages of Generative AI: Hallucinations

- An AI hallucination refers to the phenomenon where an AI system generates outputs that are not based on real-world data or facts
- These outputs can be entirely fabricated or contain significant inaccuracies
- AI hallucinations occur because the AI model predicts responses based on patterns in the data it was trained on, **rather than understanding** the information

Disadvantages of Generative AI: Hallucinations

- AWS has introduced tools like **automated reasoning** checks to combat these hallucinations
- These tools use mathematical and logic-based verification processes to ensure that the AI's output aligns with known facts, thereby reducing the likelihood of hallucinations

Disadvantages of Generative AI: Interpretability

- According to AWS, the "**interpretability**" disadvantage of generative AI refers to the challenges in understanding and explaining how these models make decisions and generate outputs
- Models, like generative adversarial networks (GANs) and variational autoencoders (VAEs), often operate as "black boxes," meaning their internal workings are not transparent to users

Disadvantages of Generative AI: Interpretability

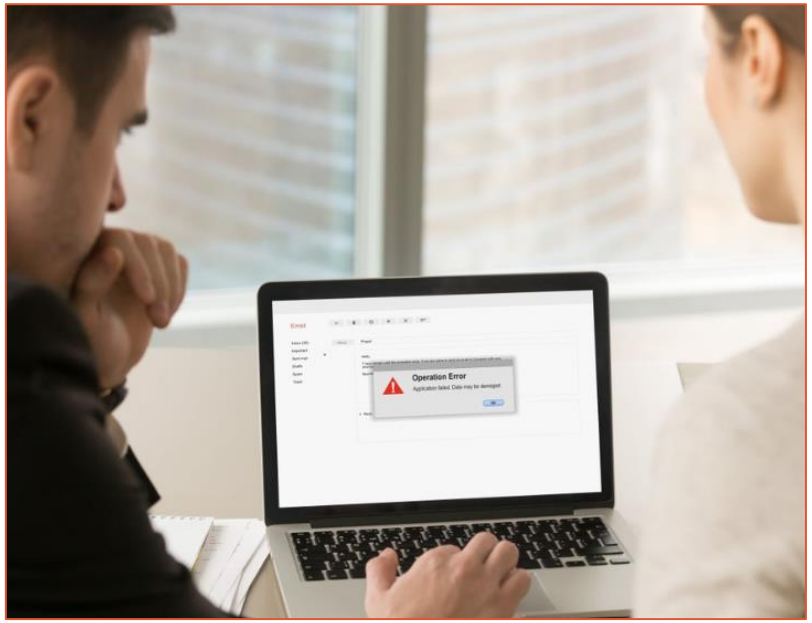
- A lack of transparency can make it difficult to trace the origins of specific outputs and understand the pathways through which data is processed and transformed
- The complexity of these models, the vast and diverse datasets used for training, and the optimization algorithms employed further complicate their interpretability

Disadvantages of Generative AI: Inaccuracy

- The "**inaccuracy**" disadvantage of generative AI refers to the potential for these models to produce outputs that are factually incorrect or misleading
- Inaccuracies can lead to various negative consequences, such as loss of customers, damage to brand reputation, and legal issues

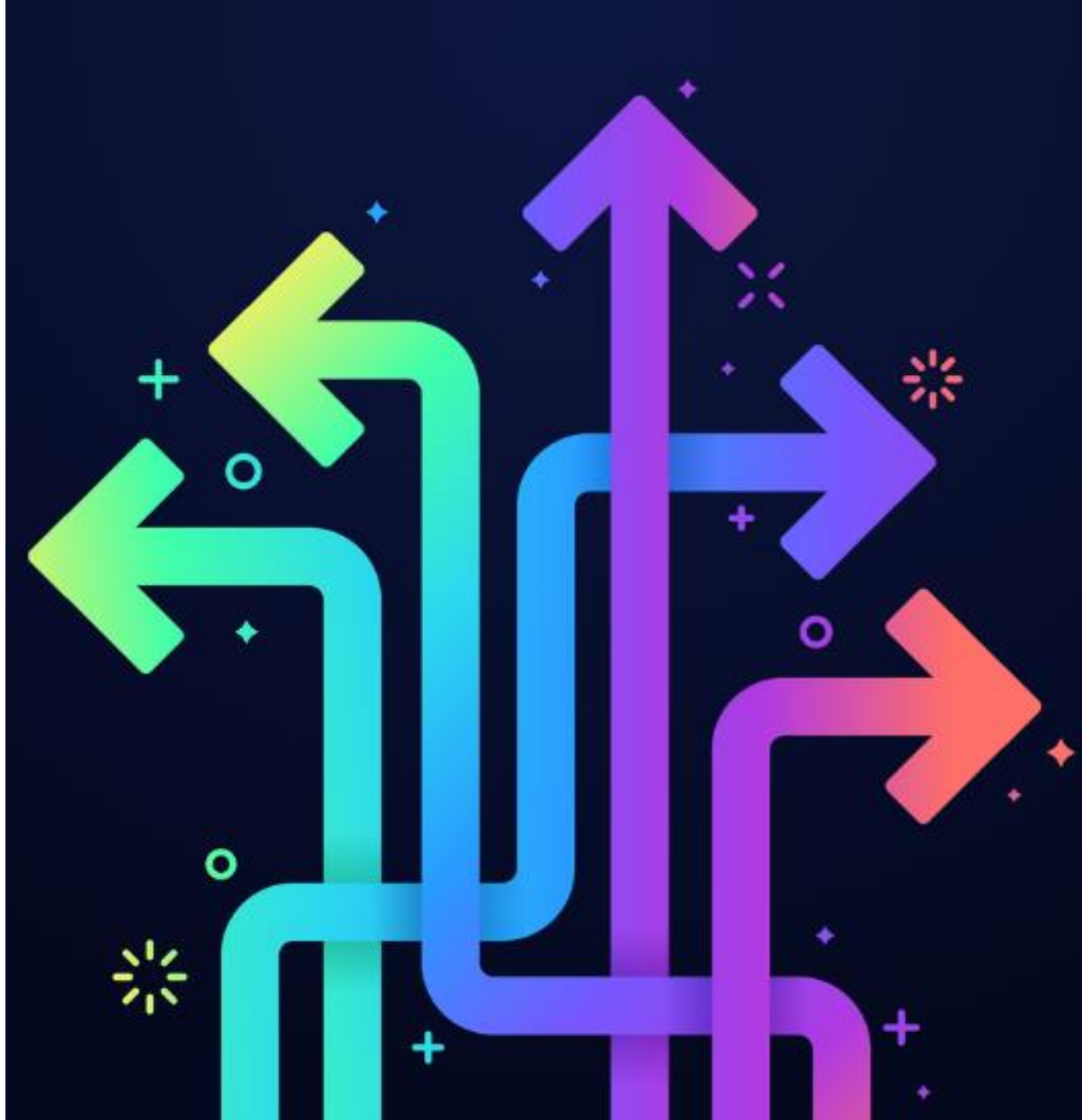
Disadvantages of Generative AI: Inaccuracy

- Inaccuracies in generative AI can happen due to several reasons:
  - The statistical nature of the models
  - Limited generalization skills
  - Non-deterministic outputs
  - Issues with the quality and currency of their training data

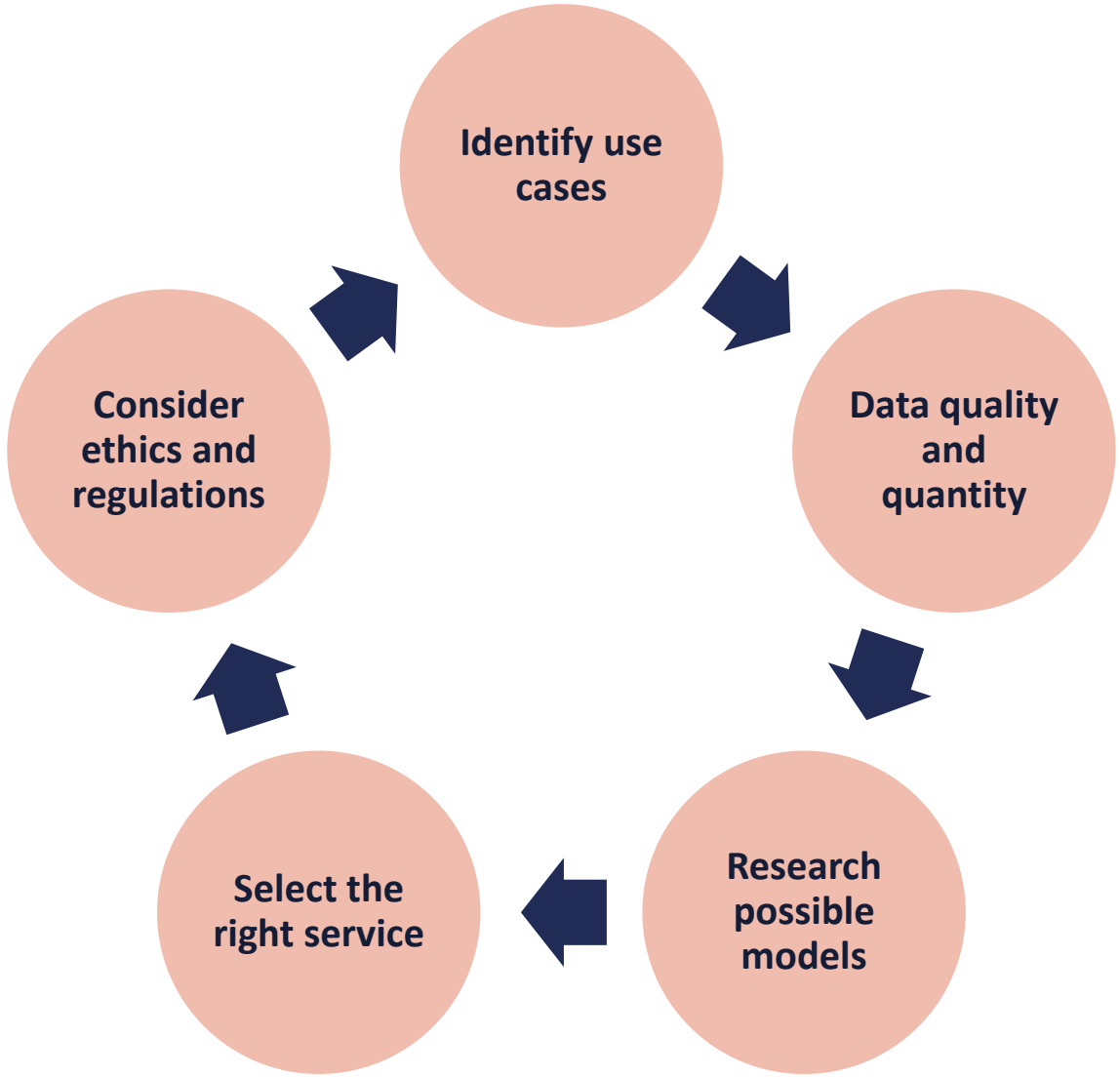


Disadvantages of Generative AI: Nondeterminism

- The "**nondeterminism**" disadvantage of generative AI refers to the inherent unpredictability of these models
- Nondeterministic systems can produce different outputs from the same input under varying conditions
- While this variability can foster creativity and adaptability, it also poses challenges in applications that require consistent, repeatable outcomes



Choosing the Appropriate Gen AI Model Type



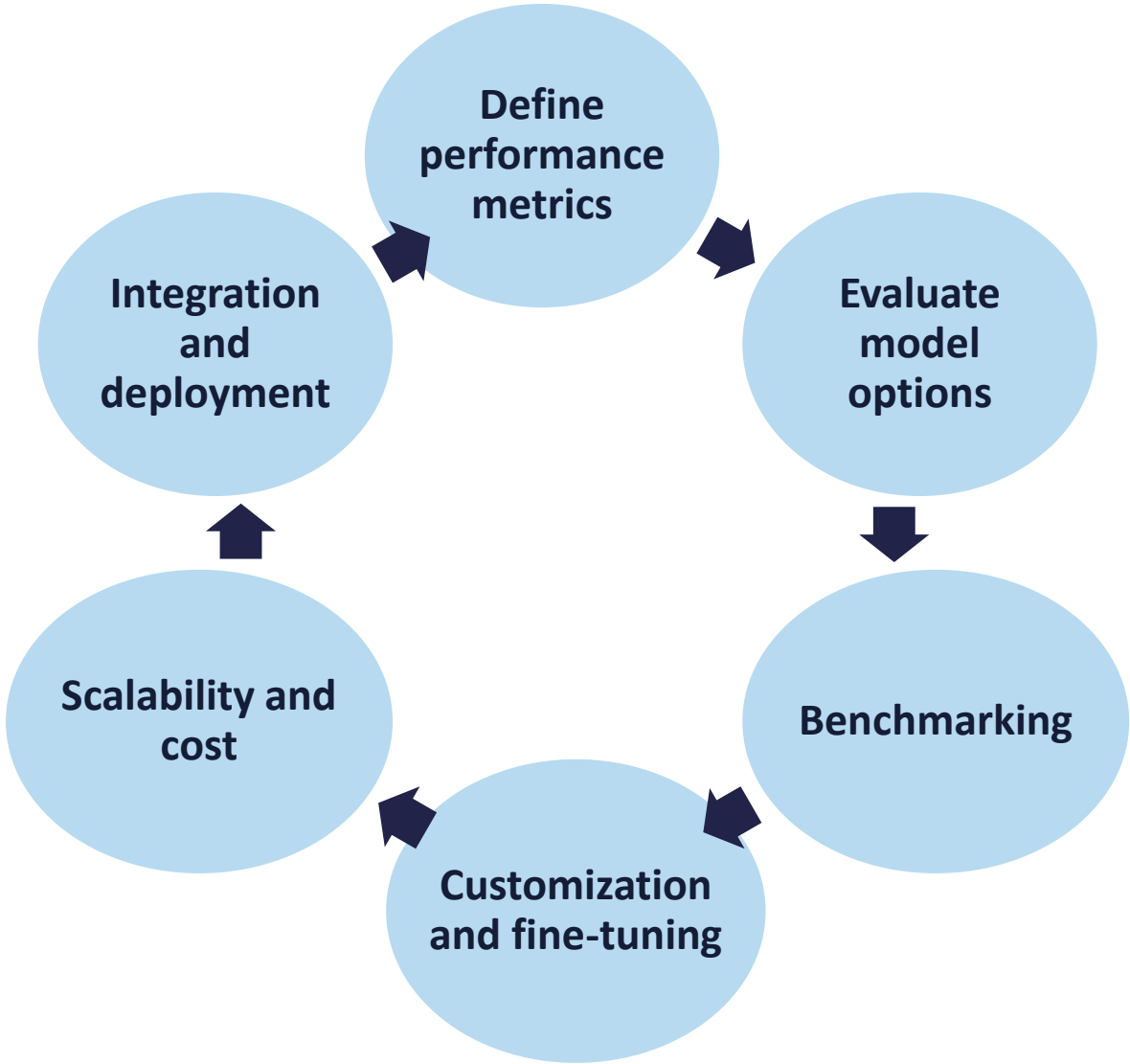
Choosing the Appropriate Model Type

- **Identify use cases**: determine the specific use cases for using generative AI like text summarization, image generation, or virtual assistants
- **Data quality and quantity**: assess the quality and quantity of your data
  - High-quality, diverse datasets are crucial for training effective generative AI models
- **Research possible models**: explore different generative AI models, both commercial and open source

Choosing the Appropriate Model Type

- **Select the right AWS services**: choose the AWS services and supporting infrastructure that best fit your needs
  - AWS offers a range of generative AI services, including Amazon Bedrock, Amazon SageMaker, and Amazon Q
- **Ethical and regulatory considerations**: ensure that the generative AI applications comply with ethical guidelines and regulatory requirements
  - This includes addressing issues like bias, privacy, and transparency

Choosing the Appropriate Model Based on Performance Requirements



Choosing Based on Performance Requirements

- **Define performance metrics**: identify the specific performance metrics that are critical for the application
  - This could include factors like accuracy, latency, throughput, and resource utilization
- **Evaluate model options**: assess different models available on AWS, such as Amazon Bedrock, Amazon SageMaker, and Amazon Q
  - Each model has unique strengths and capabilities that may align with the performance requirements

Choosing Based on Performance Requirements

- **Benchmarking:** conduct benchmark testing to compare the performance of different models against defined metrics
  - AWS offers tools and services to facilitate this process
- **Customization and fine-tuning:** consider the level of customization and fine-tuning required for each model
  - Some models may offer pre-trained capabilities that meet needs, while others might require additional training to achieve optimal performance

Choosing Based on Performance Requirements

- **Scalability and cost**: evaluate the scalability and cost implications of each model
  - Ensure that the chosen model can handle the expected workload and fits within budgetary constraints
- **Integration and deployment**: assess how easily the model can be integrated into the existing infrastructure and deployed in the production environment
  - AWS offers various services to streamline this process and ensure seamless integration

Choosing Based on Capabilities and Constraints

- Determine the specific use cases for which generative AI is needed
  - This could include tasks like text summarization, image generation, or virtual assistants
- Assess the capabilities of different generative AI models
  - Consider factors such as the types of data they can process (text, images, audio), their ability to generate high-quality outputs, and their performance in specific tasks

Choosing Based on Capabilities and Constraints

- Understand and identify any constraints that may impact the model choice
  - These could include computational resources, budget limitations, data availability, and regulatory requirements
- Explore the various generative AI models available on AWS, such as Amazon Bedrock, Amazon SageMaker, and Amazon Q
  - Each model has unique strengths and capabilities that may align with the specific requirements

Choosing the Appropriate Model Based on Compliance

- Determine the specific regulatory requirements that apply to the industry and use case
  - This could include data privacy laws, industry-specific regulations, and international standards
- Assess the compliance capabilities of different generative AI models
  - Consider factors such as data handling practices, security measures, and auditability

Choosing the Appropriate Model Based on Compliance

- Develop and implement the established governance strategies to manage compliance risks
- Finally, establish continuous monitoring processes to ensure ongoing compliance
  - Use automated tools to collect evidence and generate assessment reports, helping to stay compliant with evolving regulations
  - Use AWS Audit Manager

Business Value and Metrics for Generative AI Applications

- **Cross-domain performance**: this metric refers to the ability of a generative AI model to perform well across different domains or types of tasks
  - It evaluates how effectively a model can generalize its capabilities beyond the specific domain it was trained on, ensuring it can handle a variety of tasks with high accuracy and efficiency

Business Value and Metrics for Generative AI Applications

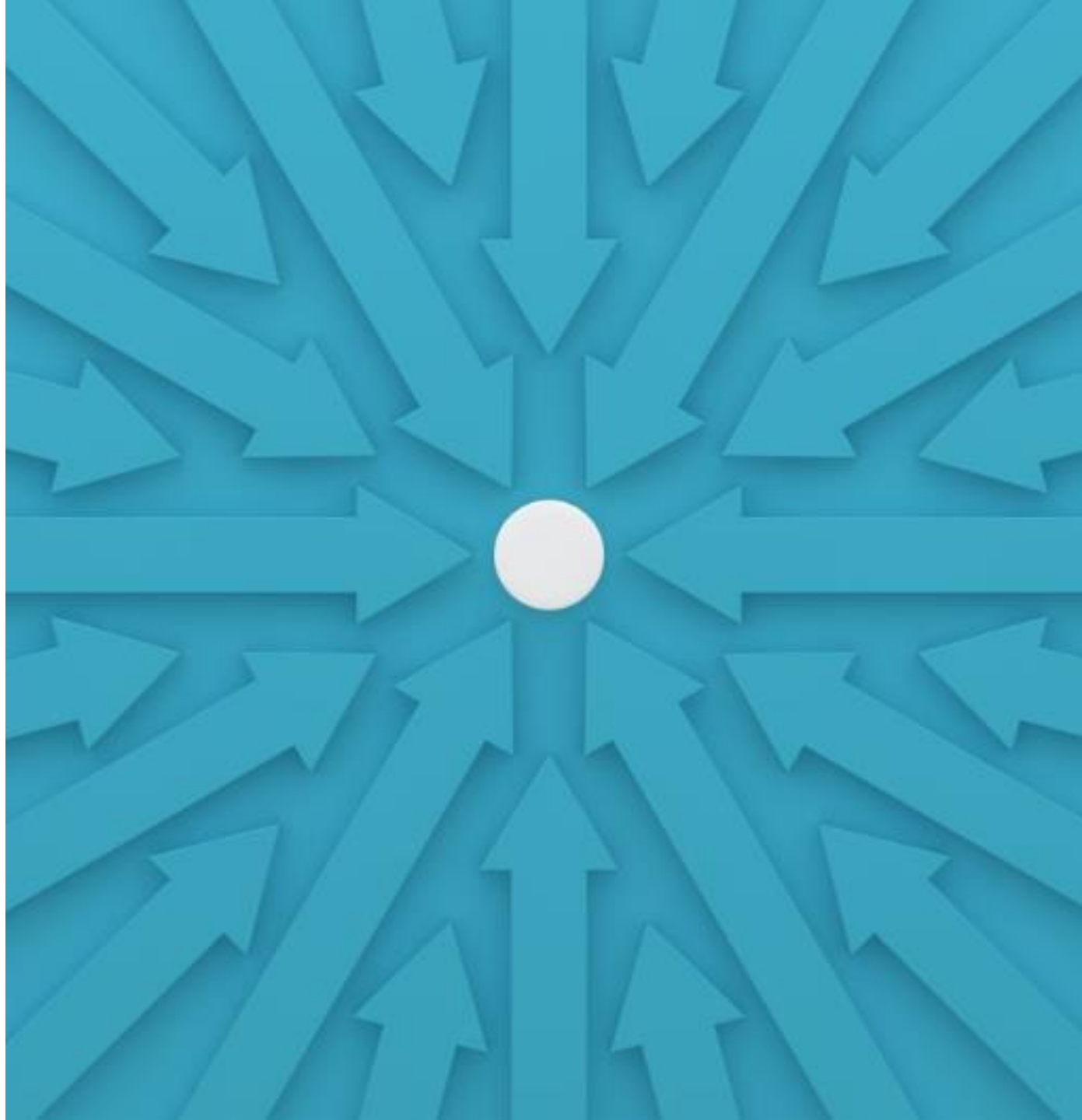
- **Conversion rate:** this measures the effectiveness of the AI in achieving desired outcomes, such as user engagement, sales, or other key performance indicators
  - This metric evaluates how well the generative AI model influences user behavior and drives specific actions, such as making a purchase, signing up for a service, or completing a task

Business Value and Metrics for Generative AI Applications

- **Average revenue per user (ARPU):** this measures the average revenue generated from each active user over a specific period
  - A crucial metric for understanding the financial performance and profitability of generative AI applications
  - Helps businesses in assessing how effectively they are monetizing their user base and identify opportunities for growth and optimization

Business Value and Metrics for Generative AI Applications

- **Accuracy**: according to AWS, this metric for generative AI evaluates how closely the AI-generated outputs match the expected or true results
  - This metric is crucial for assessing the quality and reliability of generative AI models, especially in tasks like summarization, question answering, and classification



Key Aspects of the Accuracy Metric

- **Ground truth comparison**: the AI's outputs are compared against a set of known correct answers (ground truth) to determine the accuracy of responses
- **Evaluation algorithms**: specific algorithms, such as BERTScore, ROUGE, and F1 score, are used to measure accuracy
- **Continuous improvement**: regular evaluation and fine-tuning of models based on accuracy metrics

AWS Certified AI Practitioner

We will see you in 20 hours for Session 2

Thank You!

Michael Shannon
and Eian Clair

Class will begin again tomorrow at 10:00 am Central Standard Time