Machine Learning Operations (MLOps)

Model Deployment and Operational Management

Zeham Management Technologies BootCamp

by SDAIA

October the 2nd, 2024

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

Objectives

By the end of this module, trainees will have a comprehensive understanding of:

- Install and build your own repository.
- Learn Imputers and Encoders
- Make preprocessing Pipelines.
- Train and validate the model.
- Evaluate and save your model.
- Build an API and connect it with your Machine Learning code

Agenda

Cloud Deployment of the Model

Model Deployment Strategies

Model Monitoring and Alerting

References

Cloud Deployment of the Model

What is deployment?

Deployment is the process of making a machine learning model available for use. This process can involve exposing the model as a service publicly that can be accessed via API calls. Deployment ensures that the model can handle real data and provide predictions in a timely manner.

Deployment Options

There are several options for deploying machine learning models:

- **On-premises:** Deploying models on local servers or private data centers.
- **Cloud:** Utilizing cloud service providers like AWS, GCP(Google Cloud), or Azure to host and manage models.
- **Hybrid:** Combining on-premises and cloud deployments environments.
- **Edge:** Deploying models on edge devices such as IoT devices for low-latency predictions.

Cloud Deployment Options

Cloud deployment offers flexibility, scalability, and managed services. Major cloud providers like AWS, Google Cloud Platform (GCP), and Microsoft Azure offer various services for deploying machine learning models.

AWS Deployment Options

Amazon offers vast of cloud services, the most famous ones for Machine Learning are:

- **Amazon SageMaker:** End-to-end service for building, training, and deploying ML models.
- **AWS Lambda:** Serverless compute service to run code in response to events without managing servers.

AWS Deployment Options

Amazon offers vast of cloud services, the most famous ones for Machine Learning are:

- **Elastic Kubernetes Service (EKS):** Managed Kubernetes service for running containerized applications.
- **Elastic Container Service (ECS):** is a fully managed container orchestration service that helps you to more efficiently deploy, manage, and scale containerized applications.

Google Cloud Deployment Options

- **Vertex AI:** Managed service for training and deploying ML models.
- **Google Kubernetes Engine (GKE):** Managed Kubernetes service for deploying containerized applications.
- **Cloud Functions:** Serverless execution environment for building and connecting cloud services.

Google Cloud

Azure Deployment Options

- **Azure Machine Learning:** Service for building, training, and deploying ML models.
- **Azure Kubernetes Service (AKS):** Managed Kubernetes service for running containerized applications.
- **Azure Functions:** Event-driven serverless compute service.

Best Practices for Cloud Deployment

- **Security:** Ensure data and model security by using encryption and proper access controls.
- **Scaling:** Design for scalability to handle varying loads and large volumes of requests.
- **Monitoring:** Implement monitoring and logging to track model performance and detect issues.
- **Automation:** Use CI/CD pipelines to automate deployment and updates.

Useful Videos

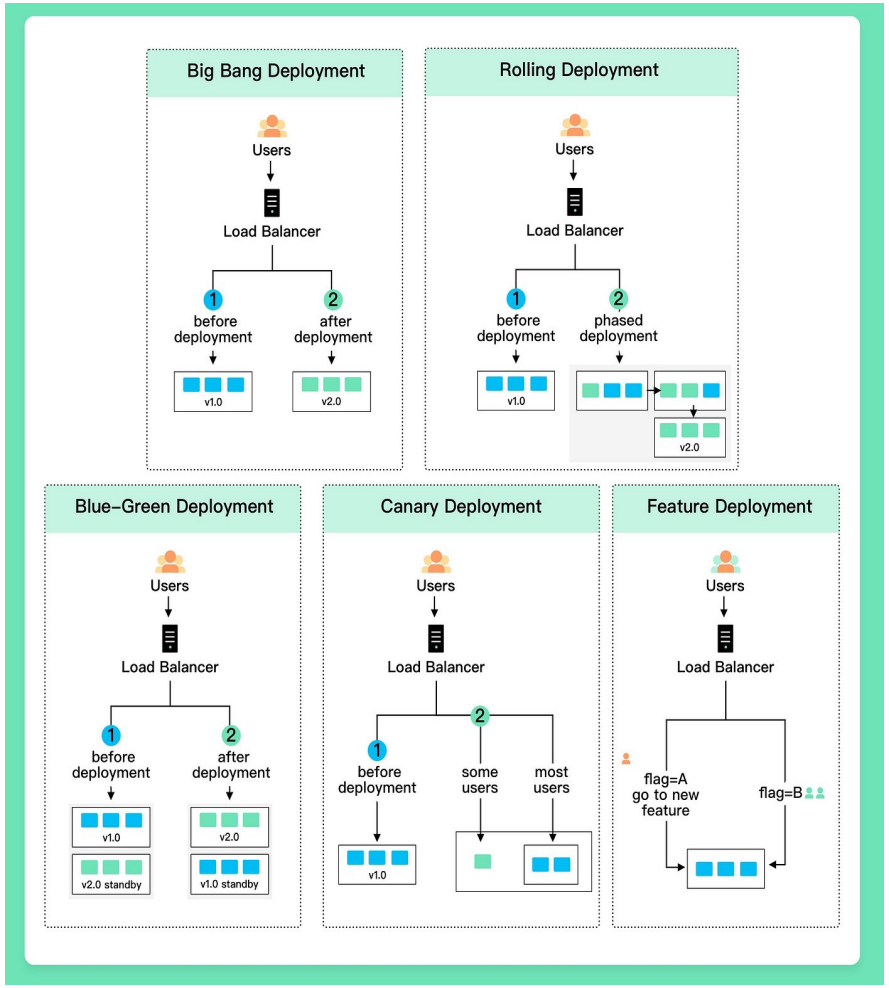
To learn how to deploy on AWS ECS:

- https://www.youtube.com/watch?v=pJ_nCklQ65w&t=1513s

Model Deployment Strategies

What is Deployment strategies?

- A **deployment strategy** is a method used by teams to effectively introduce and implement new versions or features of an application.
- It aids teams in organizing the steps and tools required to efficiently bring code changes into live environments.

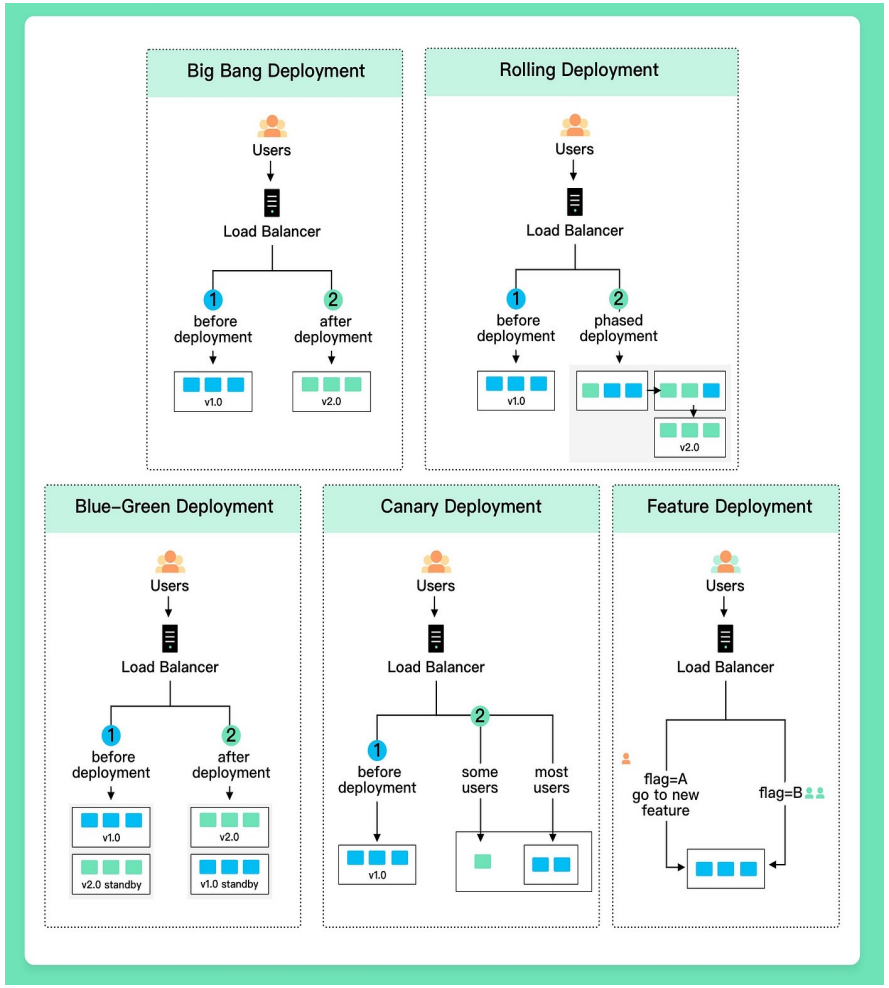


Source

What is Deployment strategies?

- It's important to understand that deployment and release are not the same, even though they might appear to be the same at first.

- A **deployment strategy** is the method by which code is moved from one environment to another to test and verify the software, and eventually make it available to users.

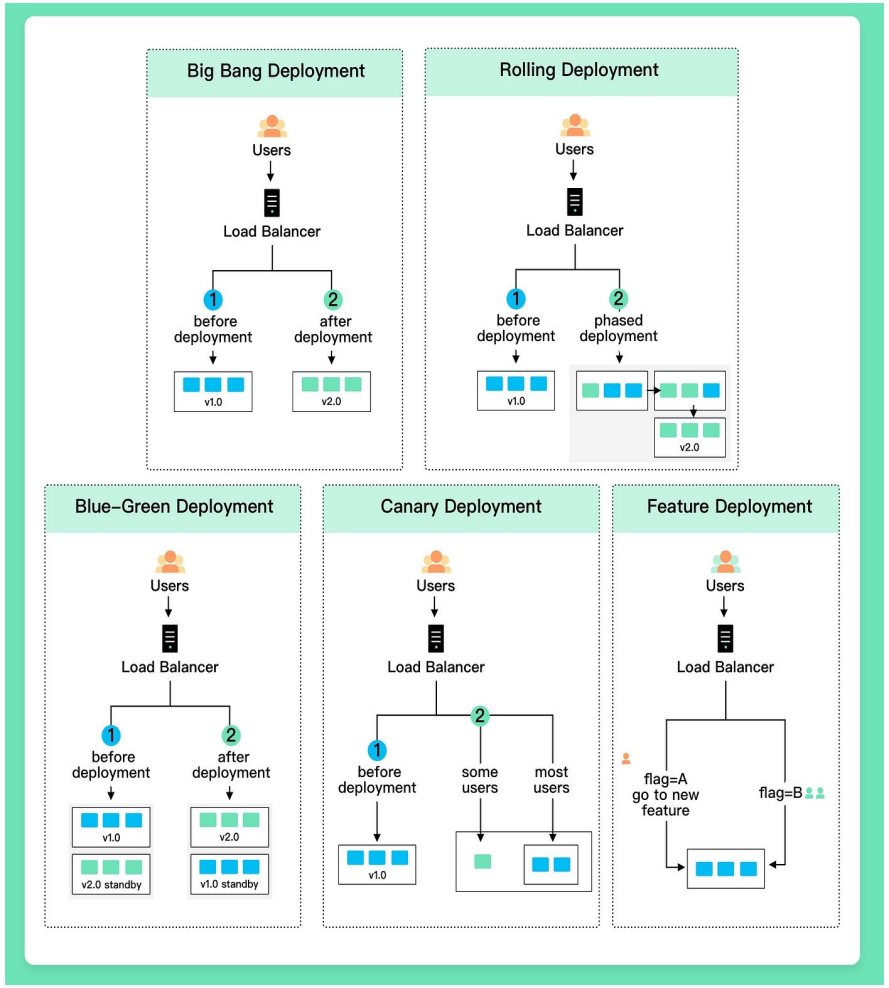


Source

Types of deployment strategies:

- Teams can choose from several deployment strategies, each with its own advantages and disadvantages based on the team's goals.

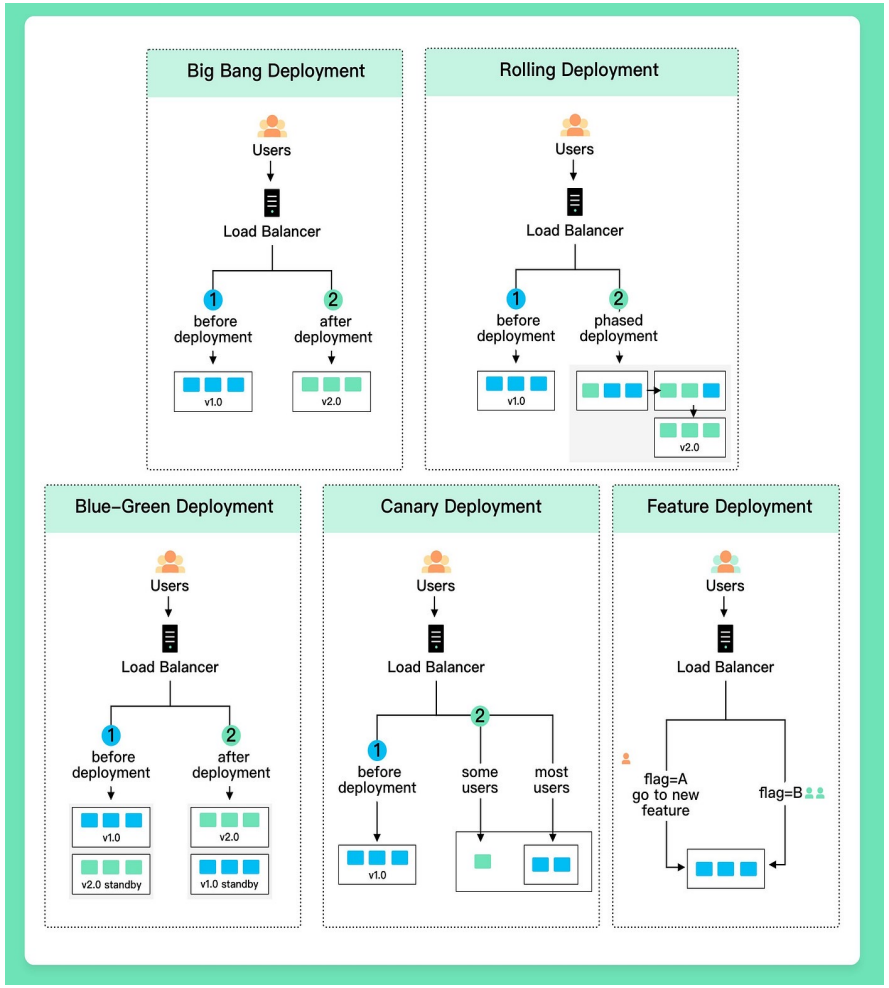
- The deployment strategy chosen by an organization depends on various factors such as team size, available resources, the complexity of the software, and how often deployments or releases occur.



Source

Types of deployment strategies:

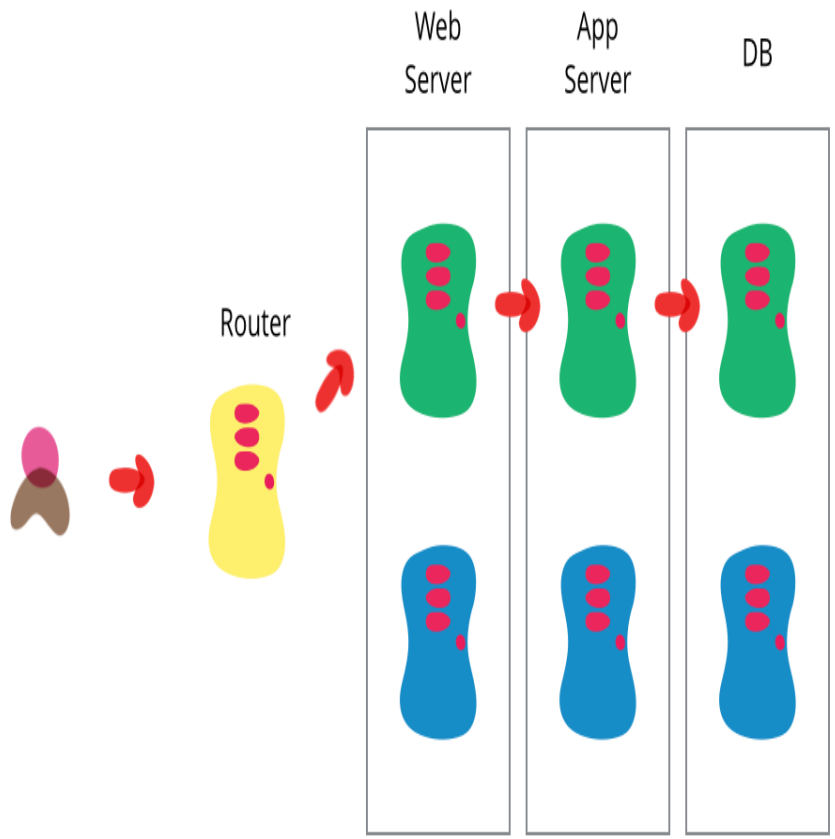
- Blue-Green
- Canary
- A/B testing
- Recreate
- Shadow



Source

What is Blue-Green Deployment?

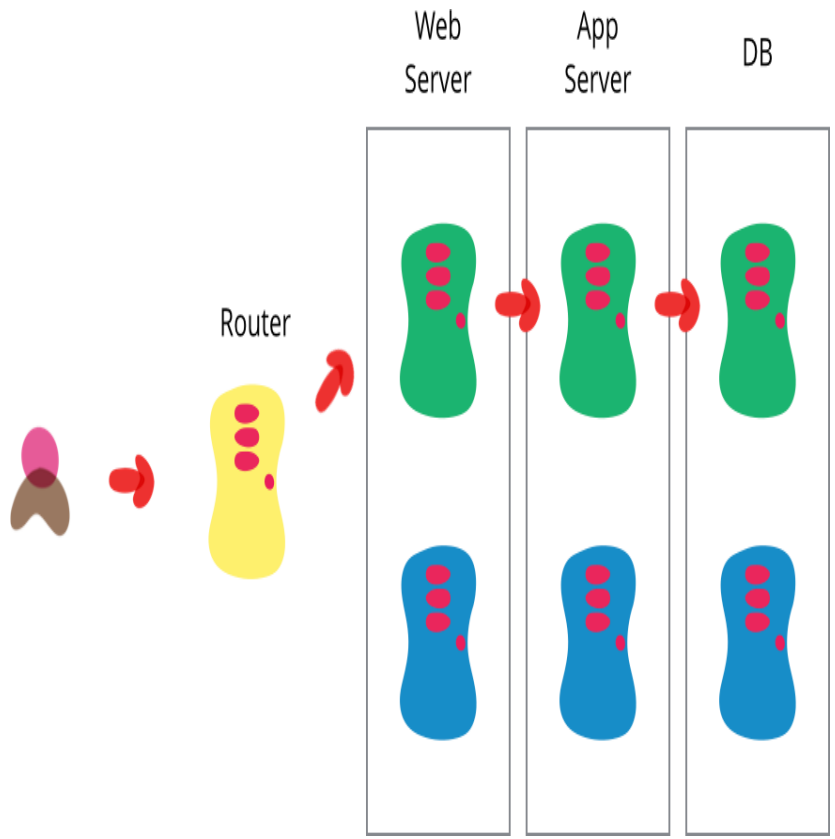
- A blue/green deployment strategy involves having two identical production environments, called "blue" and "green."
- Only one of these environments is live and handling user transactions, while the other is idle.



Source

Blue-Green Deployment:

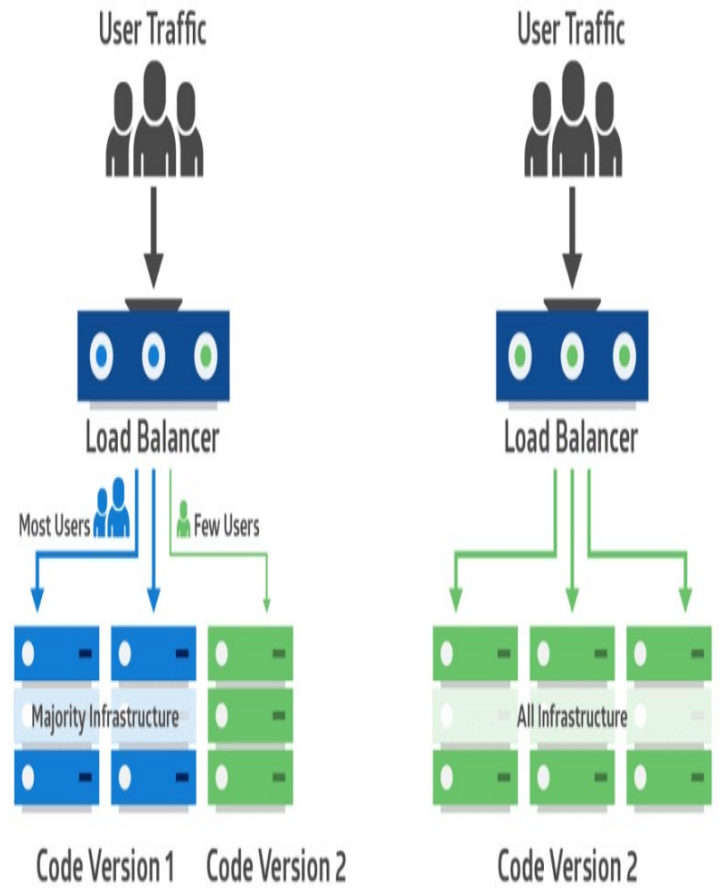
- At any given time, only one environment is live—usually the green environment, which has the new application version.
- The idle blue environment is used as a test or staging area for the final round of testing before releasing a new feature.



Source

What is a Canary Deployment:

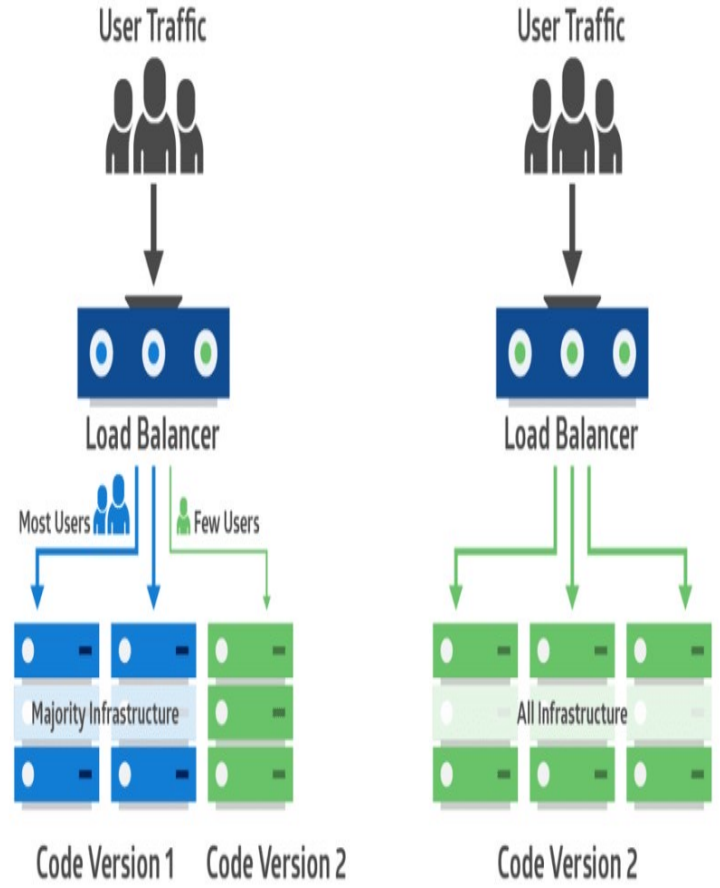
- **Canary deployments** is a strategy that lowers the risk of releasing new software by gradually rolling it out to a small group of users first.
- Using a load balancer or feature flag, traffic is directed to the new version for this subset, while most users continue to use the current version.



Source

Canary Deployment:

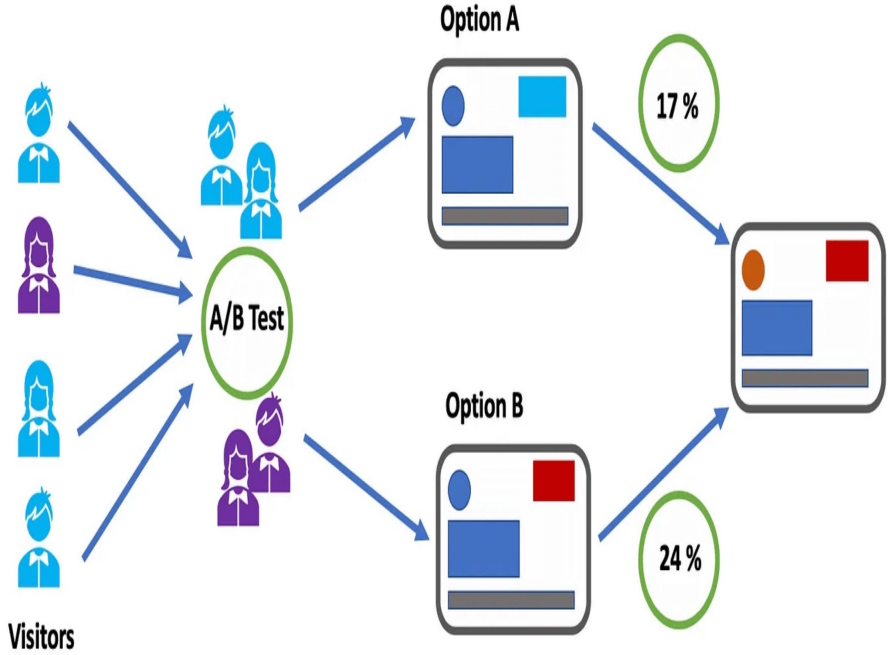
- This small group helps identify bugs, broken features, and confusing elements before the software is widely released.
- These users can be early adopters, a targeted demographic, or a random sample.



Source

What is an A/B Testing Deployment:

- A/B testing, also called split testing, involves comparing two versions of a web page or application to determine which one performs better.
- Users are randomly divided into two groups, with each group seeing a different version (A or B) of the software.

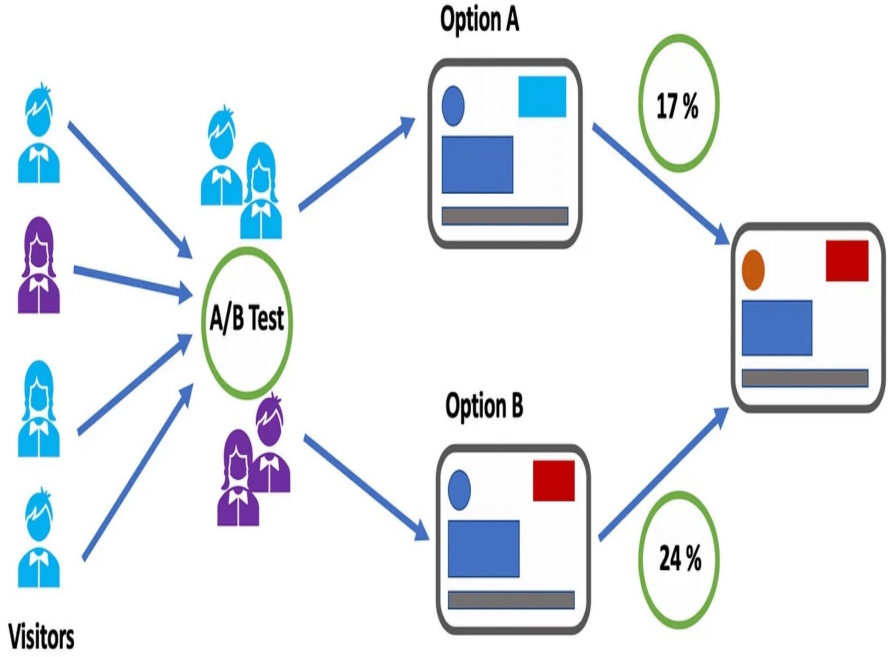


Source

A/B Testing Deployment:

- Statistical analysis of the results determines which version, A or B, performs better based on specific predefined indicators.

- A/B testing helps teams make data-driven decisions by evaluating the performance of each version, allowing them to improve the user experience for better outcomes.



Source

Deployment strategies difference:

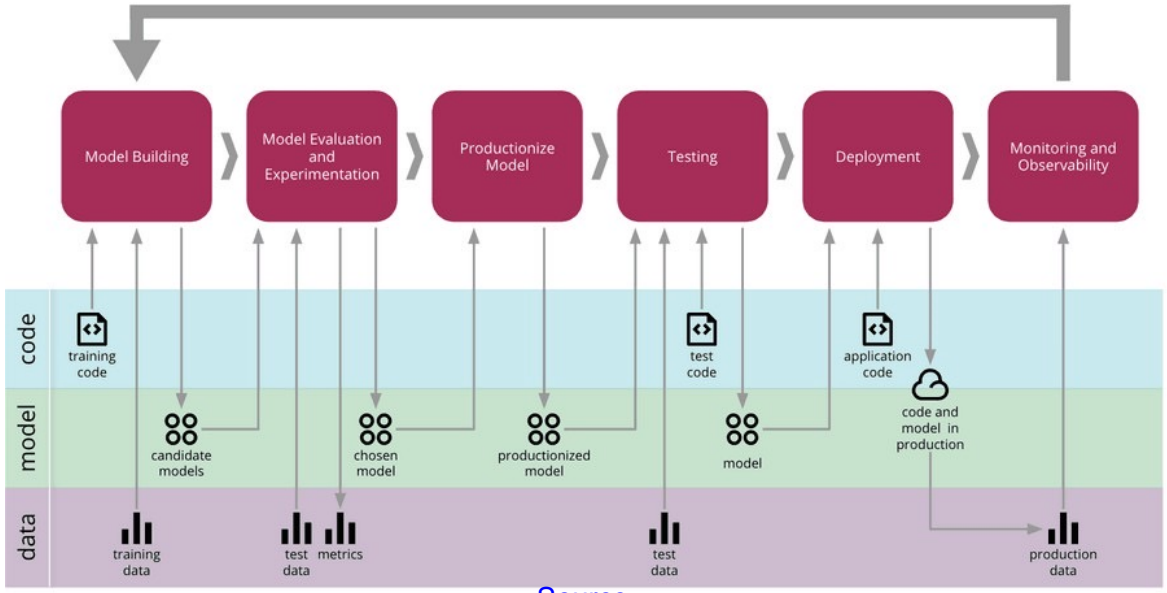
Strategy/ Parameters	No Downtime	Real traffic Testing	User targeting	Infra Cost	Rollback duration	Negative User impact	Complexity
Recreate				$			
Blue/Green				$$$			
Canary				$			
A/B Testing				$			
Shadow				$$$			

Source

Model Monitoring and Alerting

Model Monitoring:

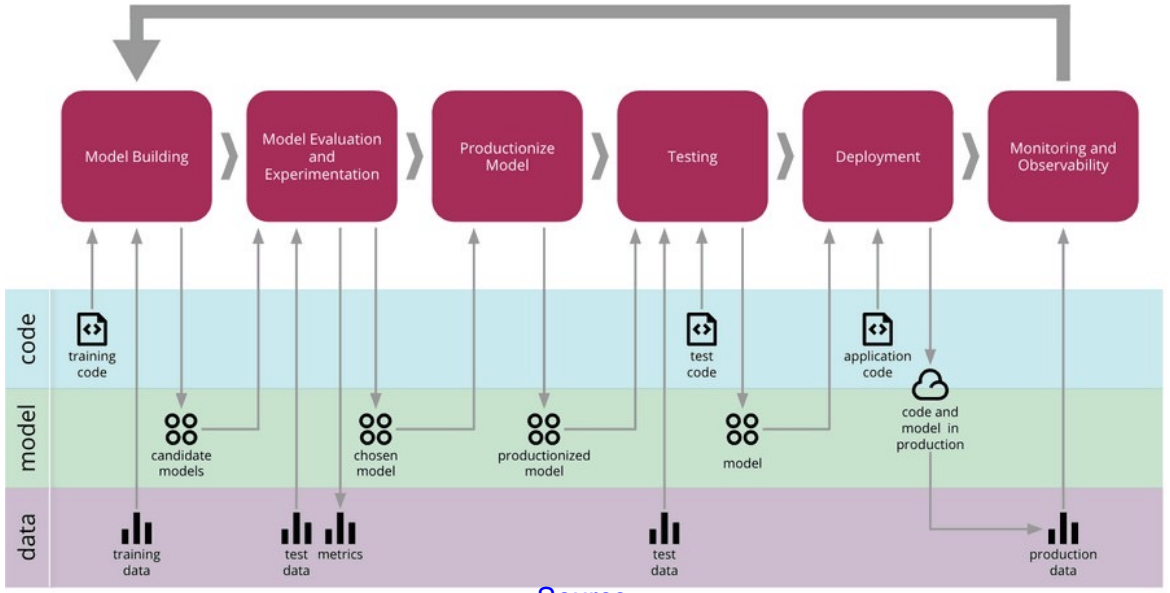
- MLOps Monitoring involves keeping track of machine learning (ML) operations to ensure that ML models perform well, remain reliable, and comply with standards in production environments.



Source

Model Monitoring:

- It includes ongoing monitoring of data quality, model performance, and system behavior.
- MLOps monitoring gives organizations valuable insights into how their ML models are performing, allowing them to identify and fix issues before they become significant problems.



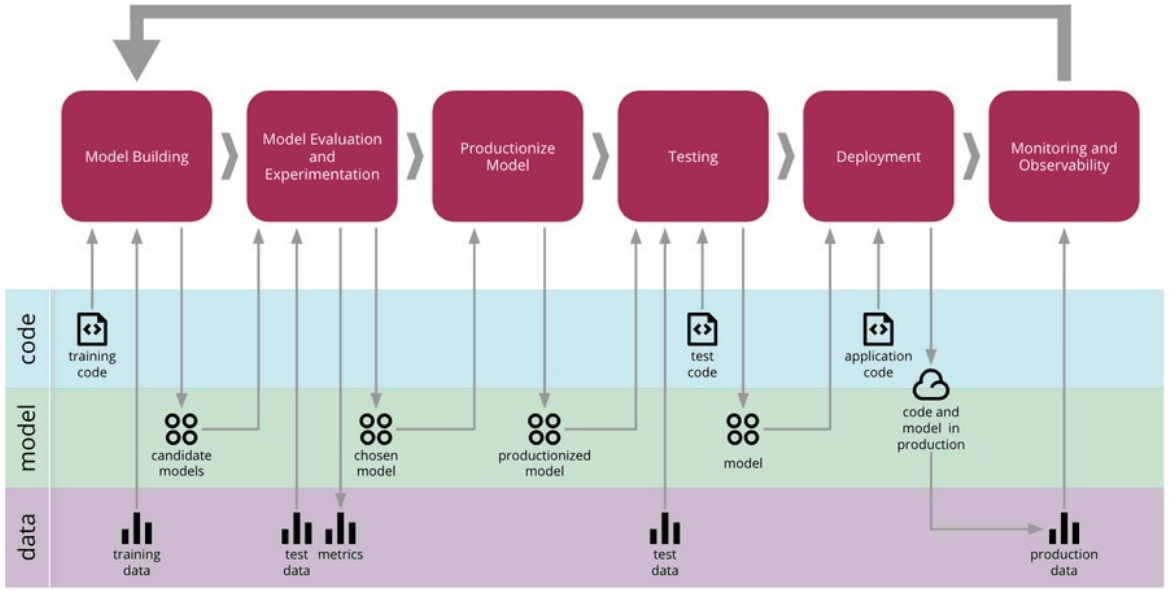
Source

Importance of Model Monitoring:

1. Identifying Data Drift and Data Quality Problems

2. Tracking Model Performance Metrics

3. Early Identification of Model Degradation

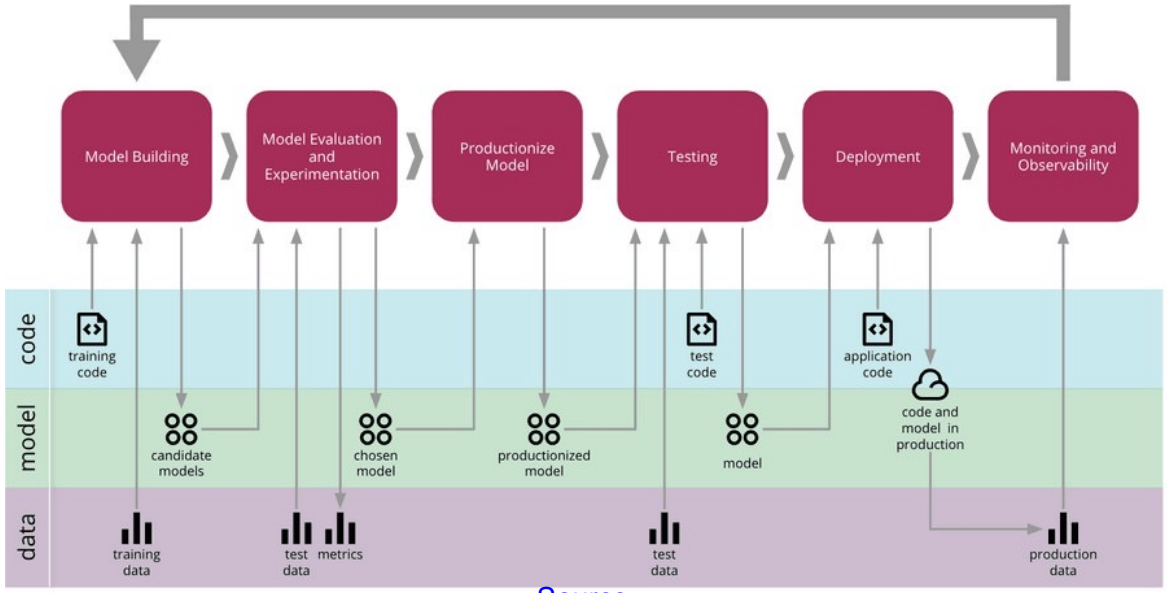


Source

Importance of Model Monitoring:

4. Maintaining Regulatory Compliance and Ethical Standards

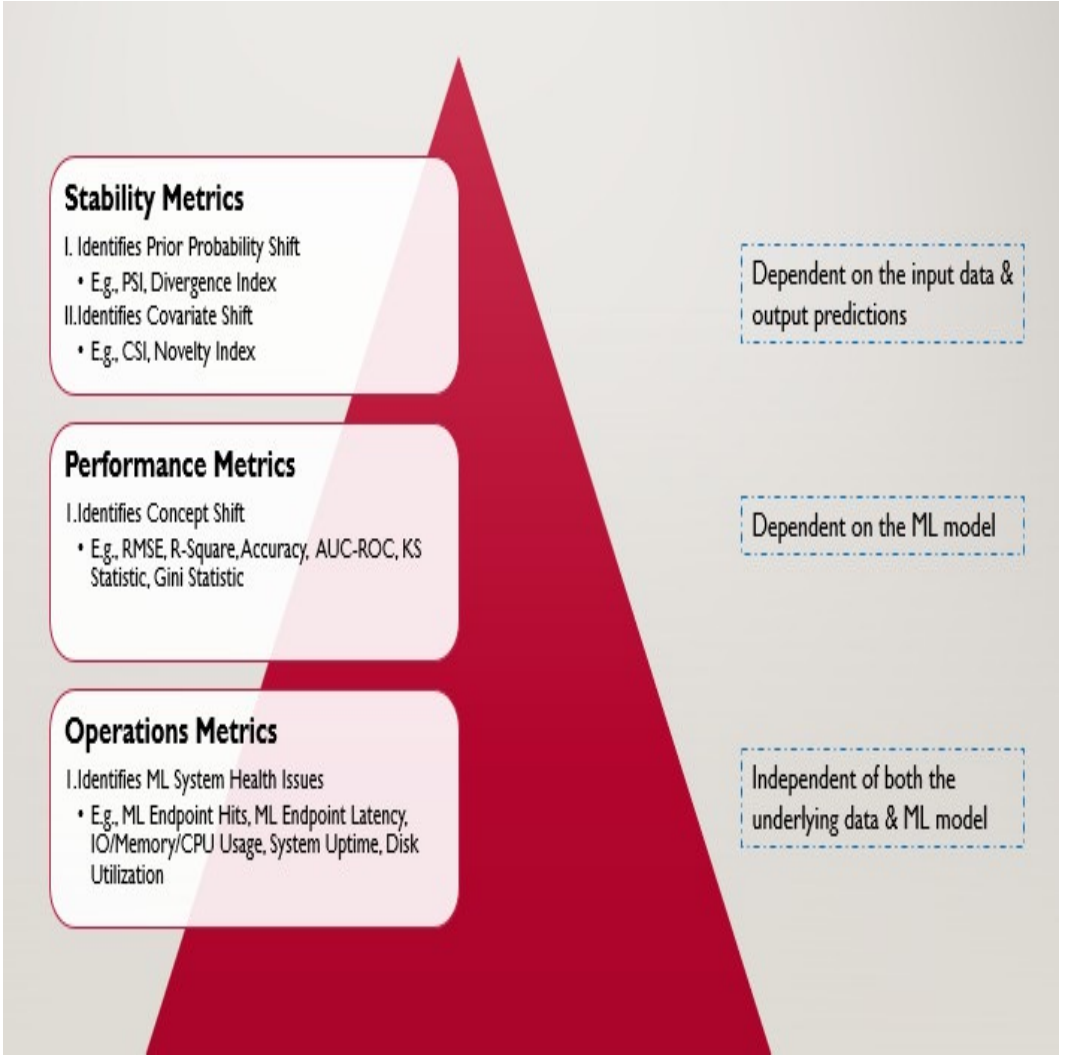
5. Optimizing Performance and Managing Resources



Source

MLOps Monitoring Metrics:

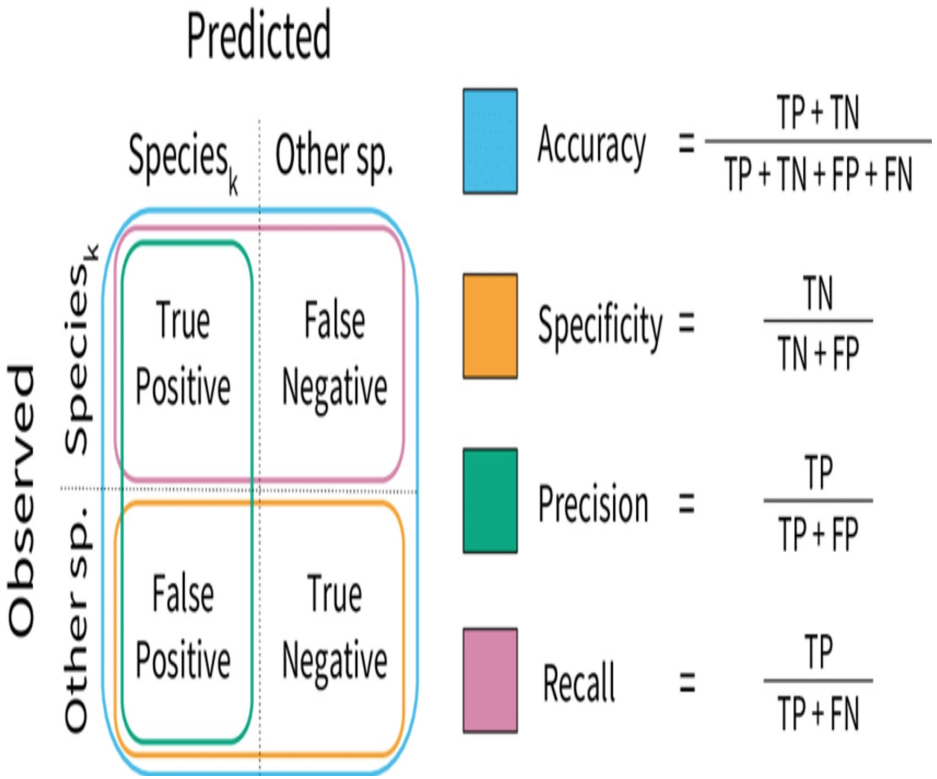
MLOps monitoring involves keeping an eye on various metrics to evaluate the performance and behavior of ML models. These metrics offer important insights into the health and effectiveness of models in production. Here are some commonly tracked MLOps monitoring metrics



Source

Accuracy and performance Metrics:

- Accuracy is a basic metric that measures how correct the predictions made by ML models are.
- It shows the proportion of correct predictions out of the total predictions made.

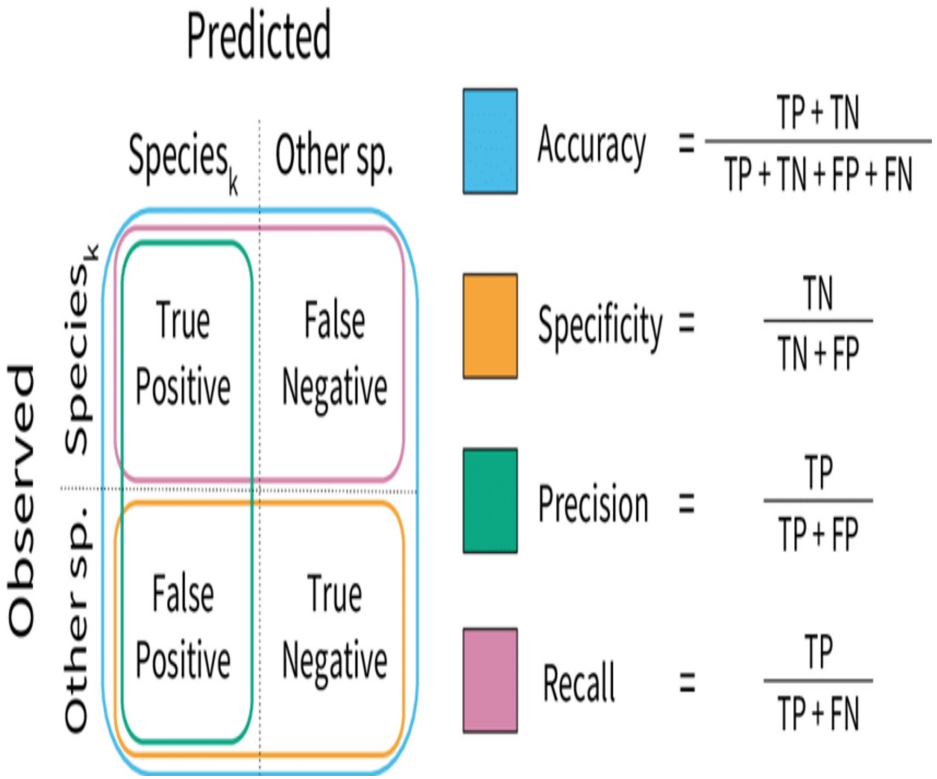


Source

Accuracy and performance Metrics:

- Other commonly used performance metrics include precision, recall, F1 score, and area under the ROC curve (AUC-ROC).

- These metrics help assess the overall performance and effectiveness of ML models.

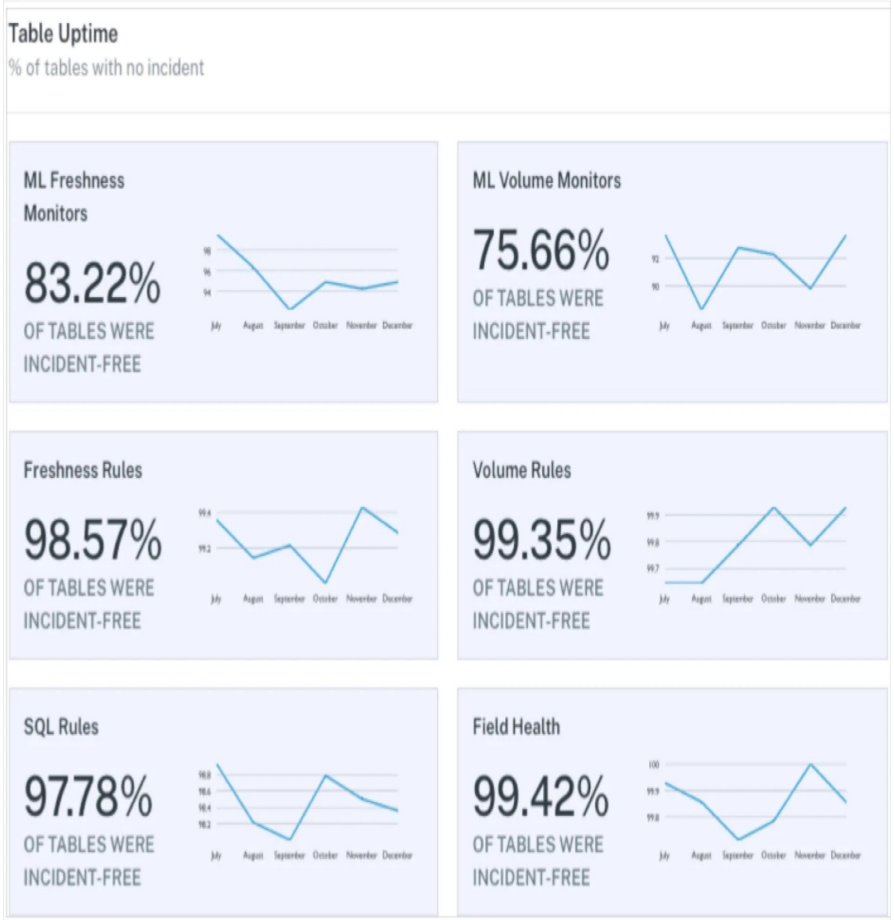


Source

Data Quality metrics:

- Data quality metrics evaluate the quality, completeness, and consistency of input data.

- They help identify issues like missing values, outliers, or inconsistent data distributions.

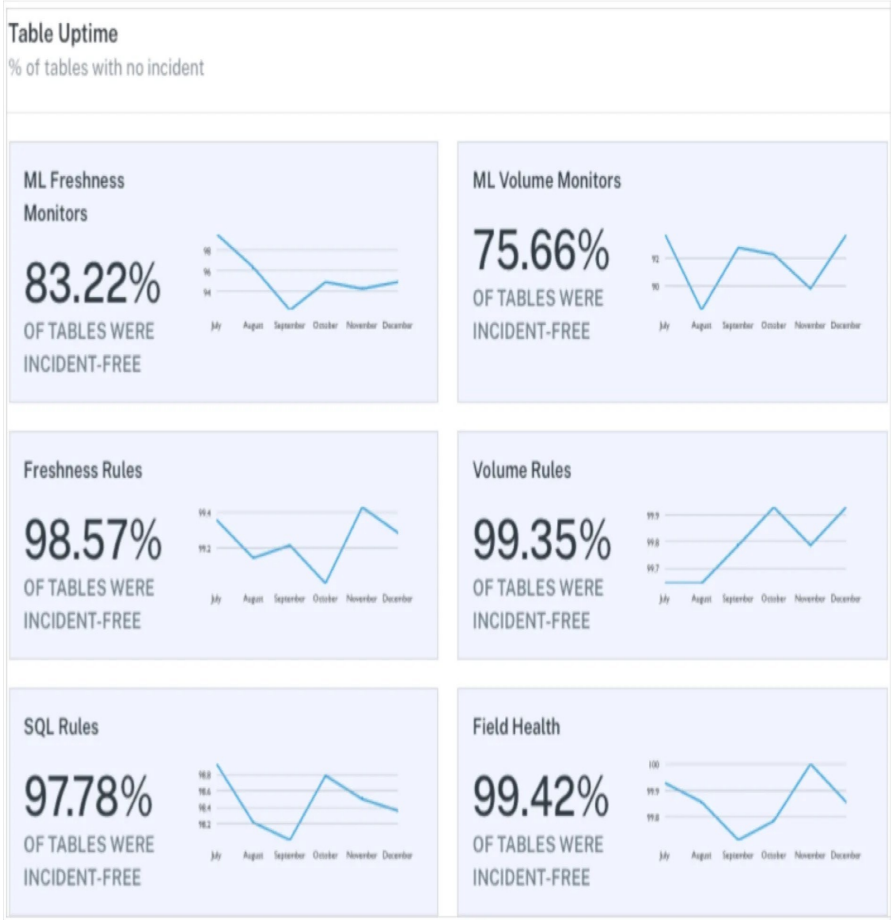


Source

Data Quality metrics:

- Common data quality metrics include data completeness, data consistency, and data distribution similarity.

- Monitoring these metrics is essential to maintain the integrity of input data and avoid biases or drops in model performance.



Source

Model-specific metrics:

- Depending on the type of ML model and the problem domain, there may be specific metrics that are important to monitor.

- For instance, in natural language processing tasks, metrics like BLEU score or perplexity can be used to assess the quality of language generation models.

Indicators	Random Forest	Naïve Bayes	SVM
Accuracy	99.09% +/- 0.66%	96.14% +/- 1.29%	93.69% +/- 1.78%
Recall	99.45% +/- 0.53%	94.27% +/- 1.26%	89.91% +/- 2.87%
Precision	99.10% +/- 1.02%	99.59% +/- 1.05%	100.00% +/- 0.00%
AUC	1.000 +/- 0.001	0.994 +/- 0.017	1.000 +/- 0.000
Sensitivity	99.45% +/- 0.53%	94.27% +/- 1.26%	89.91% +/- 2.87%
Specificity	98.57% +/- 1.45%	99.18% +/- 2.14%	100.00% +/- 0.00%
MAE	0.91% +/- 0.66%	3.86% +/- 1.29%	6.31% +/- 1.78%
-	Neural Net	Deep Learning	KNN
Accuracy	96.52% +/- 1.06%	98.04% +/- 0.93%	71.56% +/- 2.66%
Recall	95.29% +/- 1.28%	97.80% +/- 1.58%	70.07% +/- 5.05%
Precision	99.07% +/- 1.22%	99.08% +/- 1.20%	82.44% +/- 6.31%
AUC	0.994 +/- 0.006	1.000 +/- 0.000	0.805 +/- 0.026
Sensitivity	95.29% +/- 1.28%	97.80% +/- 1.58%	70.07% +/- 5.05%
Specificity	98.55% +/- 1.92%	98.57% +/- 1.90%	74.86% +/- 8.81%
MAE	3.48% +/- 1.06%	1.96% +/- 0.93%	28.44% +/- 2.66%
-	Decision Tree	Auto-MLP	-
Accuracy	95.42% +/- 2.35%	96.22% +/- 1.56%	-
Recall	94.95% +/- 3.81%	94.00% +/- 2.45%	-
Precision	97.63% +/- 1.33%	99.91% +/- 0.28%	-
AUC	0.952 +/- 0.024	0.996 +/- 0.004	-
Sensitivity	94.95% +/- 3.81%	94.00% +/- 2.45%	-
Specificity	96.21% +/- 1.96%	99.87% +/- 0.41%	-
MAE	4.58% +/- 2.35%	3.78% +/- 1.56%	-

Source

Model-specific metrics:

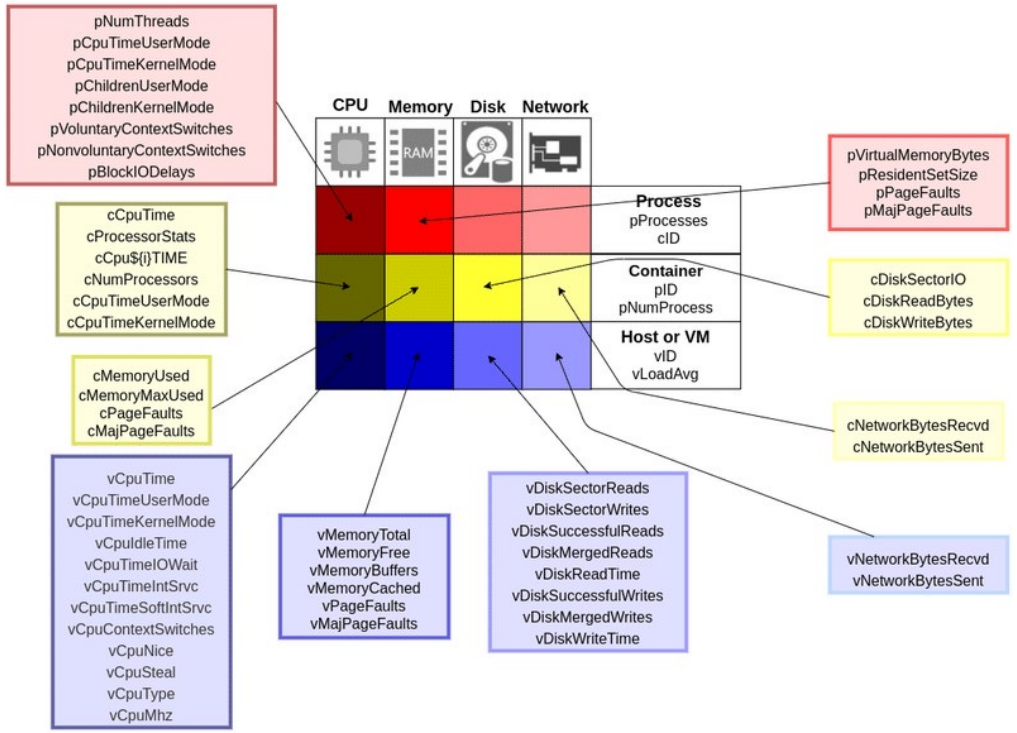
- Similarly, in computer vision tasks, metrics such as mean average precision (mAP) or Intersection over Union (IoU) can be monitored to evaluate the performance of object detection or segmentation models.
- These model-specific metrics help measure the performance and suitability of ML models for a particular tasks.

Indicators	Random Forest	Naïve Bayes	SVM
Accuracy	99.09% +/- 0.66%	96.14% +/- 1.29%	93.69% +/- 1.78%
Recall	99.45% +/- 0.53%	94.27% +/- 1.26%	89.91% +/- 2.87%
Precision	99.10% +/- 1.02%	99.59% +/- 1.05%	100.00% +/- 0.00%
AUC	1.000 +/- 0.001	0.994 +/- 0.017	1.000 +/- 0.000
Sensitivity	99.45% +/- 0.53%	94.27% +/- 1.26%	89.91% +/- 2.87%
Specificity	98.57% +/- 1.45%	99.18% +/- 2.14%	100.00% +/- 0.00%
MAE	0.91% +/- 0.66%	3.86% +/- 1.29%	6.31% +/- 1.78%
-	Neural Net	Deep Learning	KNN
Accuracy	96.52% +/- 1.06%	98.04% +/- 0.93%	71.56% +/- 2.66%
Recall	95.29% +/- 1.28%	97.80% +/- 1.58%	70.07% +/- 5.05%
Precision	99.07% +/- 1.22%	99.08% +/- 1.20%	82.44% +/- 6.31%
AUC	0.994 +/- 0.006	1.000 +/- 0.000	0.805 +/- 0.026
Sensitivity	95.29% +/- 1.28%	97.80% +/- 1.58%	70.07% +/- 5.05%
Specificity	98.55% +/- 1.92%	98.57% +/- 1.90%	74.86% +/- 8.81%
MAE	3.48% +/- 1.06%	1.96% +/- 0.93%	28.44% +/- 2.66%
-	Decision Tree	Auto-MLP	-
Accuracy	95.42% +/- 2.35%	96.22% +/- 1.56%	-
Recall	94.95% +/- 3.81%	94.00% +/- 2.45%	-
Precision	97.63% +/- 1.33%	99.91% +/- 0.28%	-
AUC	0.952 +/- 0.024	0.996 +/- 0.004	-
Sensitivity	94.95% +/- 3.81%	94.00% +/- 2.45%	-
Specificity	96.21% +/- 1.96%	99.87% +/- 0.41%	-
MAE	4.58% +/- 2.35%	3.78% +/- 1.56%	-

Source

Resource utilization metrics:

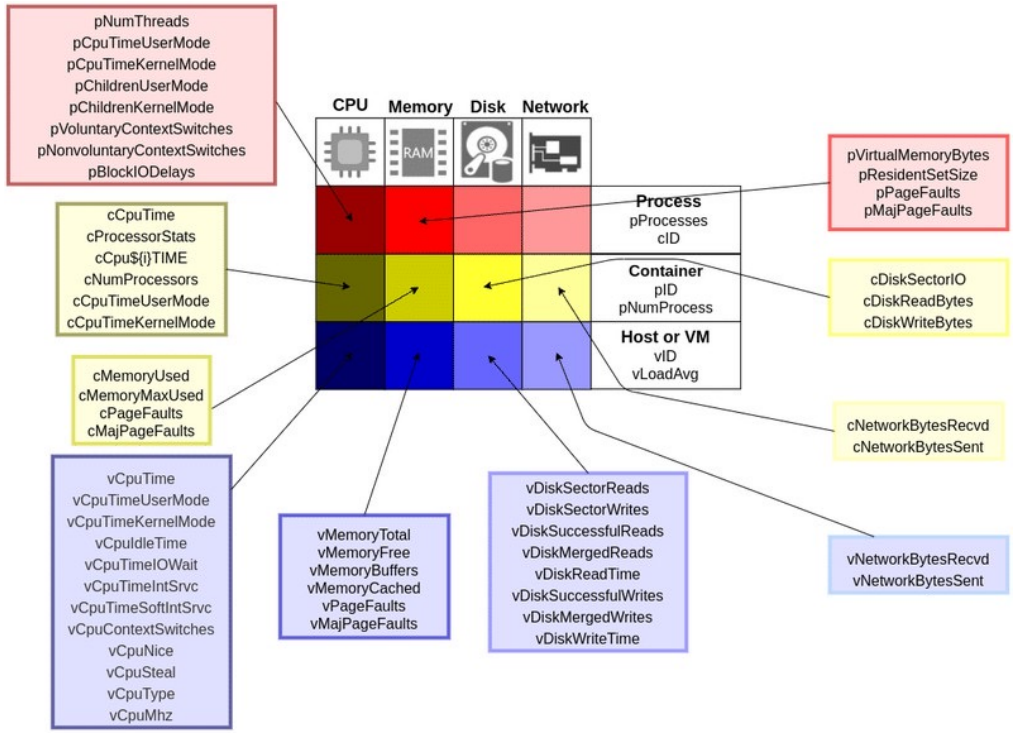
- Monitoring resource utilization metrics helps evaluate the efficiency and cost-effectiveness of ML models.
- These metrics include CPU and GPU usage, memory consumption, and disk I/O.



Source

Resource utilization metrics:

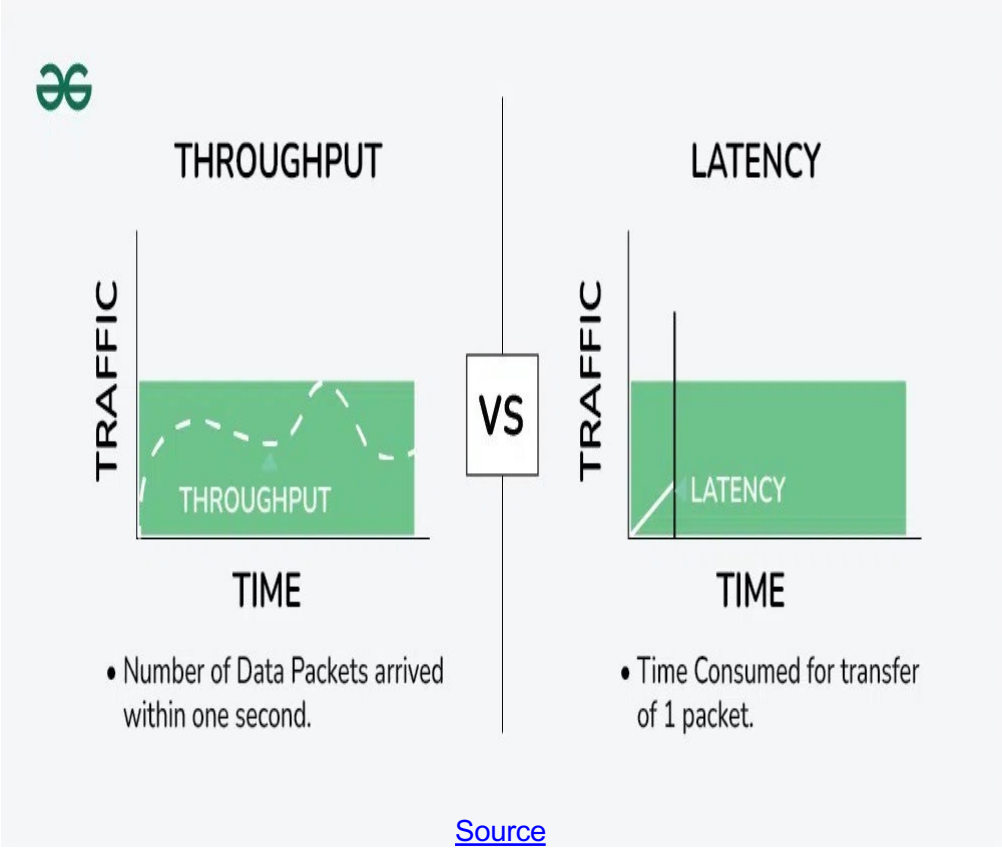
- By keeping track of resource utilization, organizations can identify bottlenecks, optimize resource allocation, and ensure the optimal use of computational resources.



Source

Latency and Throughput metrics:

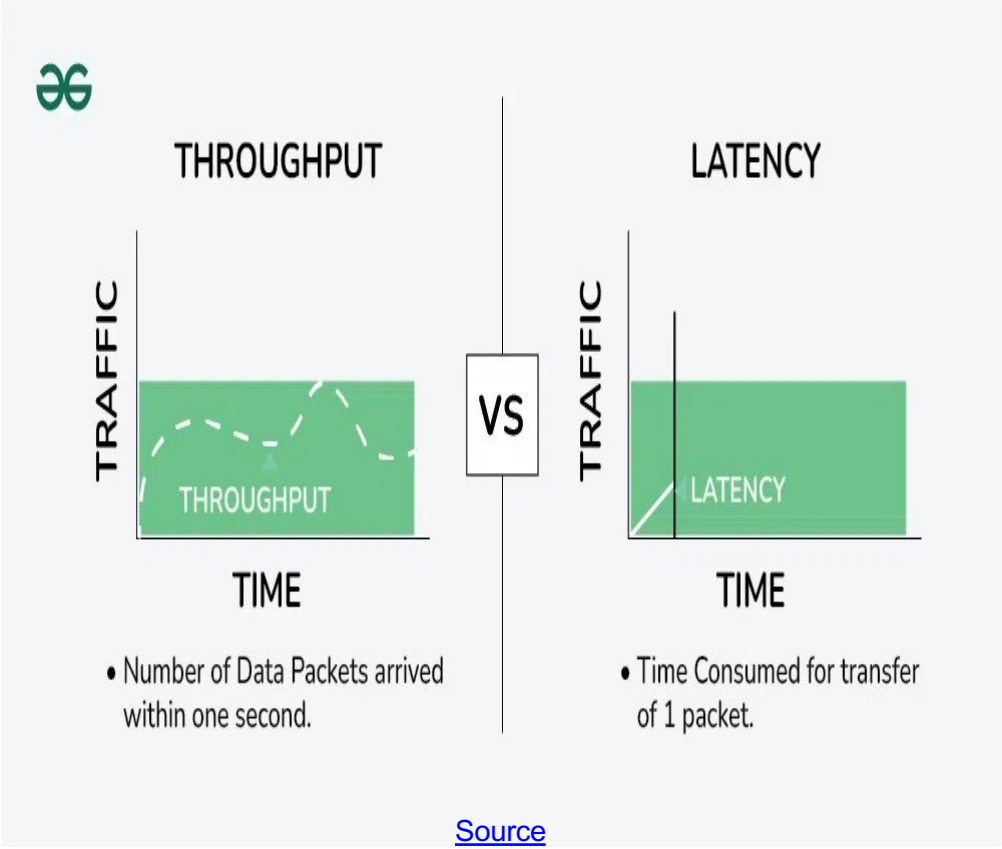
- **Latency and Throughput** metrics assess the response time and processing capacity of ML models.
- **Latency** measures the time a model takes to generate predictions.
- **Throughput** indicates the number of predictions processed in a given time period.



Source

Latency and Throughput metrics:

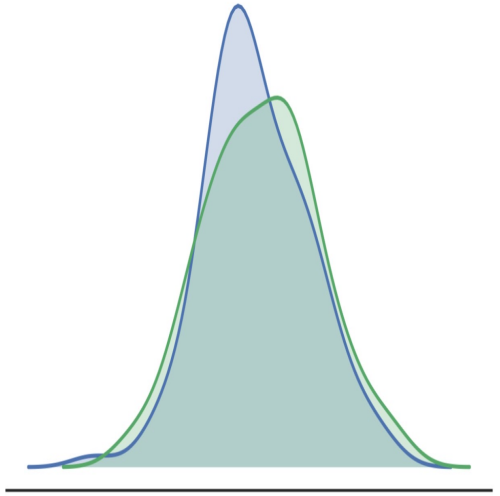
- Monitoring these metrics is crucial to ensure timely and efficient request processing, particularly in real-time or high-throughput applications.

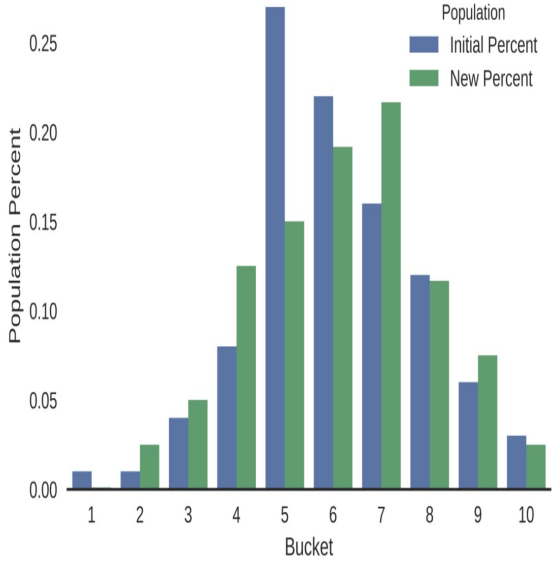


Source

Model drift metrics:

- Model drift metrics help identify changes in the behavior or performance of ML models over time.
- These metrics measure how much the model's predictions or outputs deviate from the expected baseline behavior.

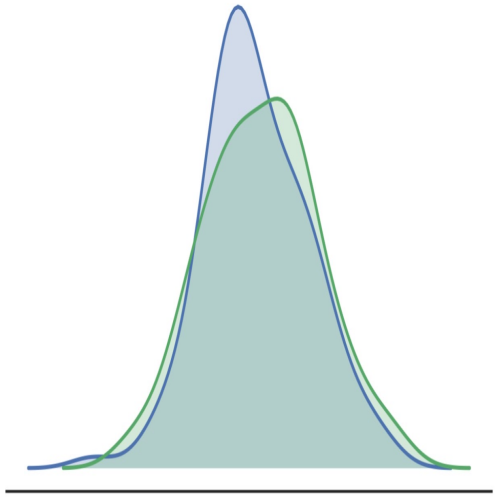


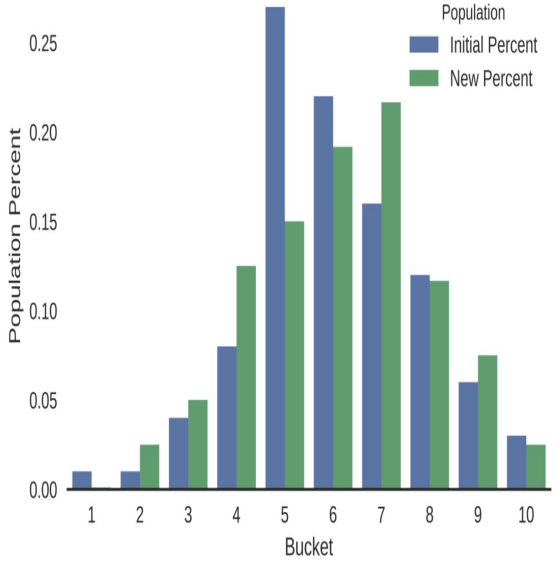


Source

Model drift metrics:

- By monitoring model drift, organizations can detect when models start losing accuracy or becoming less reliable due to shifts in data distributions or other factors.
- Model drift metrics allow for proactive measures to maintain model performance and address the effects of concept drift.

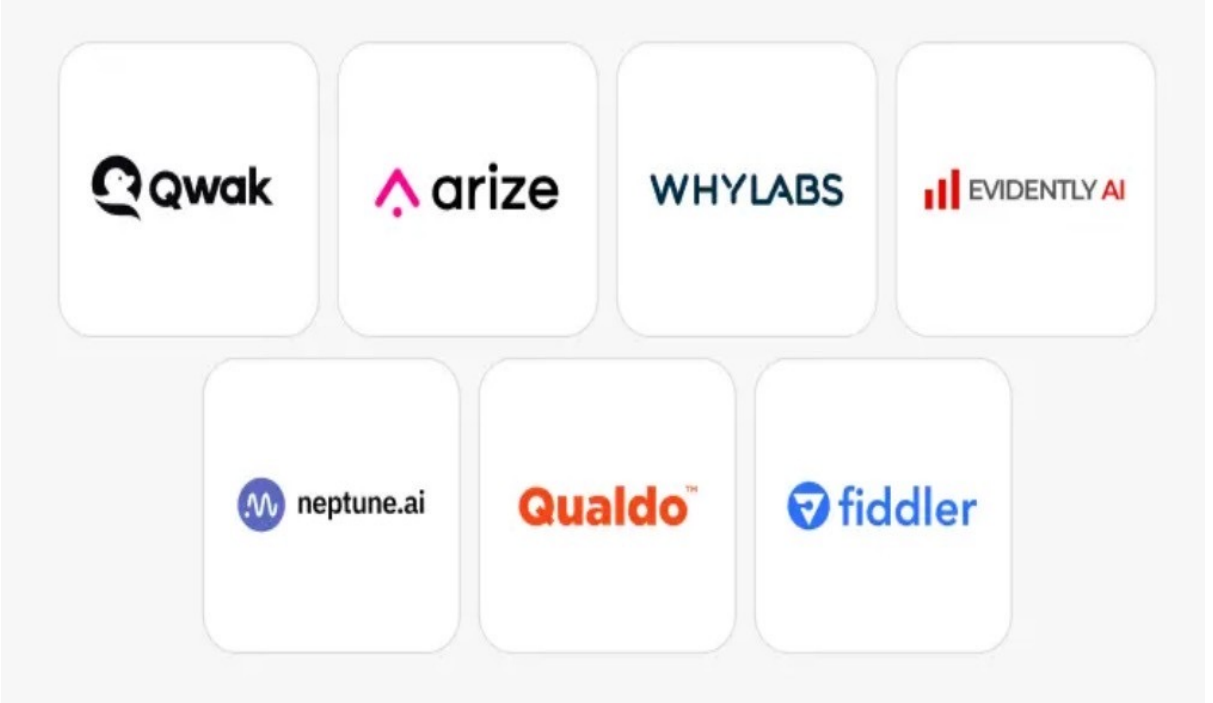




Source

MLOps Model Monitoring FrameWork:

- To implement effective MLOps monitoring, organizations can set up a model monitoring framework that details the processes, tools, and responsibilities for monitoring ML models in production.



Source

Key components of monitoring framework:

- **Define Monitoring Objectives and Metrics:** Clearly outline the goals of monitoring and specify the metrics to evaluate model performance, data quality, and system behavior.

- **Data Collection and Storage:** Implement processes to continuously collect and securely store monitoring data for analysis and historical reference.

Key components of monitoring framework:

- **Real-time Monitoring and Alerting:** Set up real-time monitoring and alerting systems to promptly detect and notify stakeholders of any issues or anomalies.

- **Visualization and Reporting:** Create tools and dashboards for visualizing monitoring data and generate regular reports to track model performance and trends.

Key components of monitoring framework:

- **Model Validation and Retraining:** Regularly validate models against new data and retrain them as necessary to maintain accuracy and effectiveness.

- **Governance and Compliance:** Ensure that monitoring processes comply with relevant regulations and ethical standards, maintaining transparency and accountability.

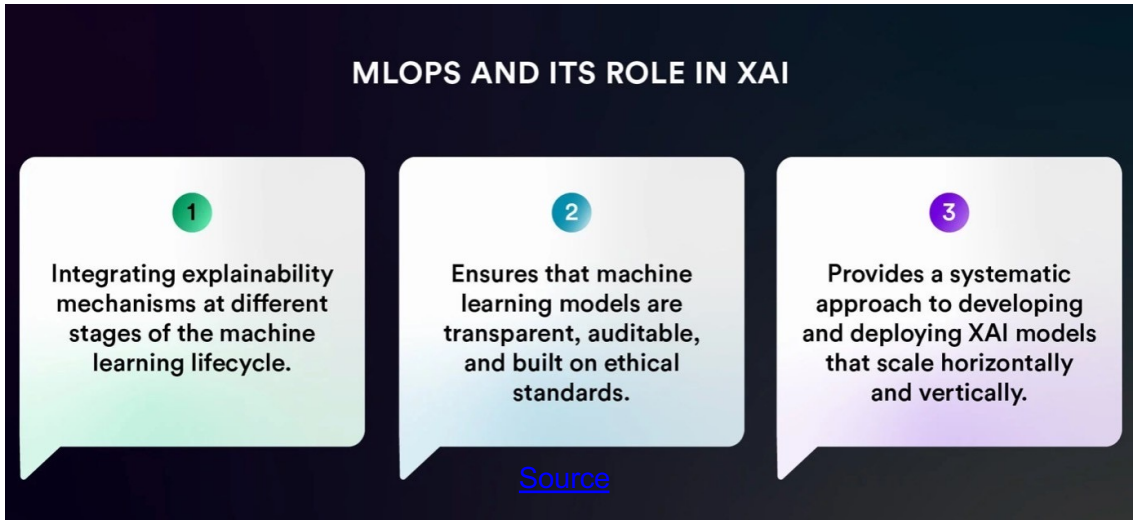
Key components of monitoring framework:

- **Continuous Improvement**: Establish a feedback loop to continuously refine and enhance the monitoring framework based on insights and changing needs.

Model Explainability and Interpretability

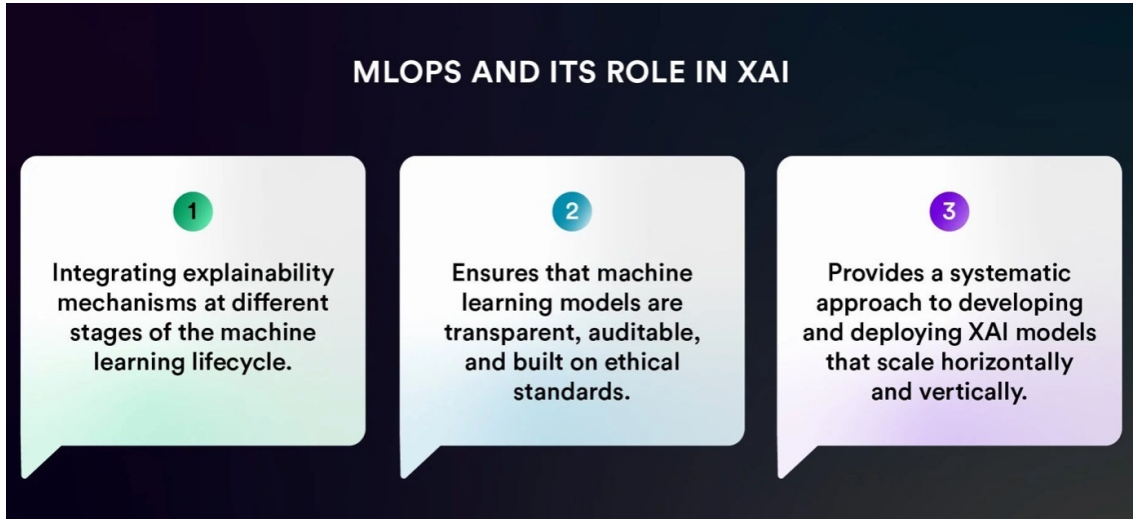
Explainable AI (XAI):

- In recent years, artificial intelligence (AI) has advanced significantly, with applications now widespread in fields such as finance, healthcare, and retail.
- Despite this progress, AI models have faced criticism for being **opaque**, making it difficult to understand their decision-making processes.



Explainable AI (XAI):

- Consequently, Explainable AI (XAI) has emerged as a crucial research area, aiming to make machine learning processes more transparent, accountable, and interpretable.



Source

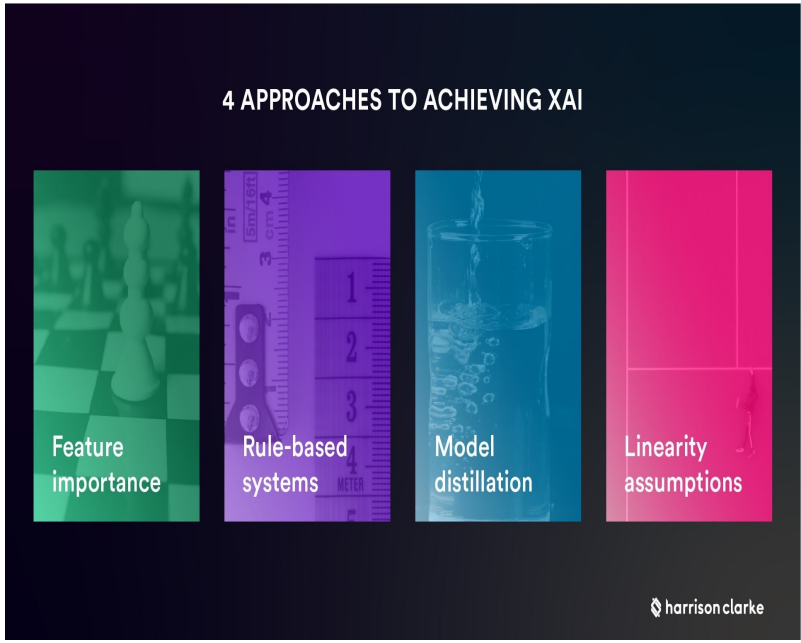
Approaches to achieve XAI

Source

Approaches to achieve XAI

- **Feature Importance:** This method evaluates the impact of each feature on the model's output, helping to clarify the reasoning behind decisions.

- **Rule-based Systems:** These systems use a set of if-then rules to determine how inputs should be classified or predicted. The rules are designed to be easily understandable by users.

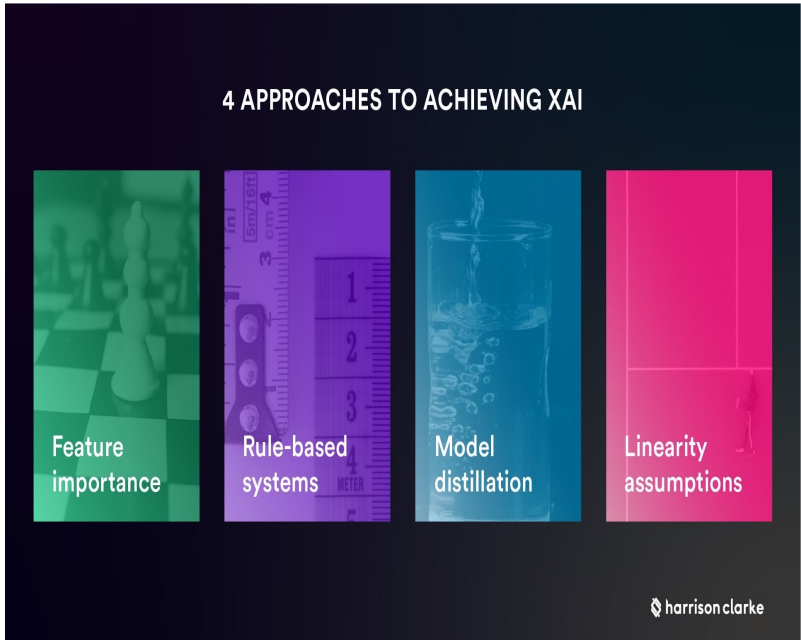


Source

Approaches to achieve XAI

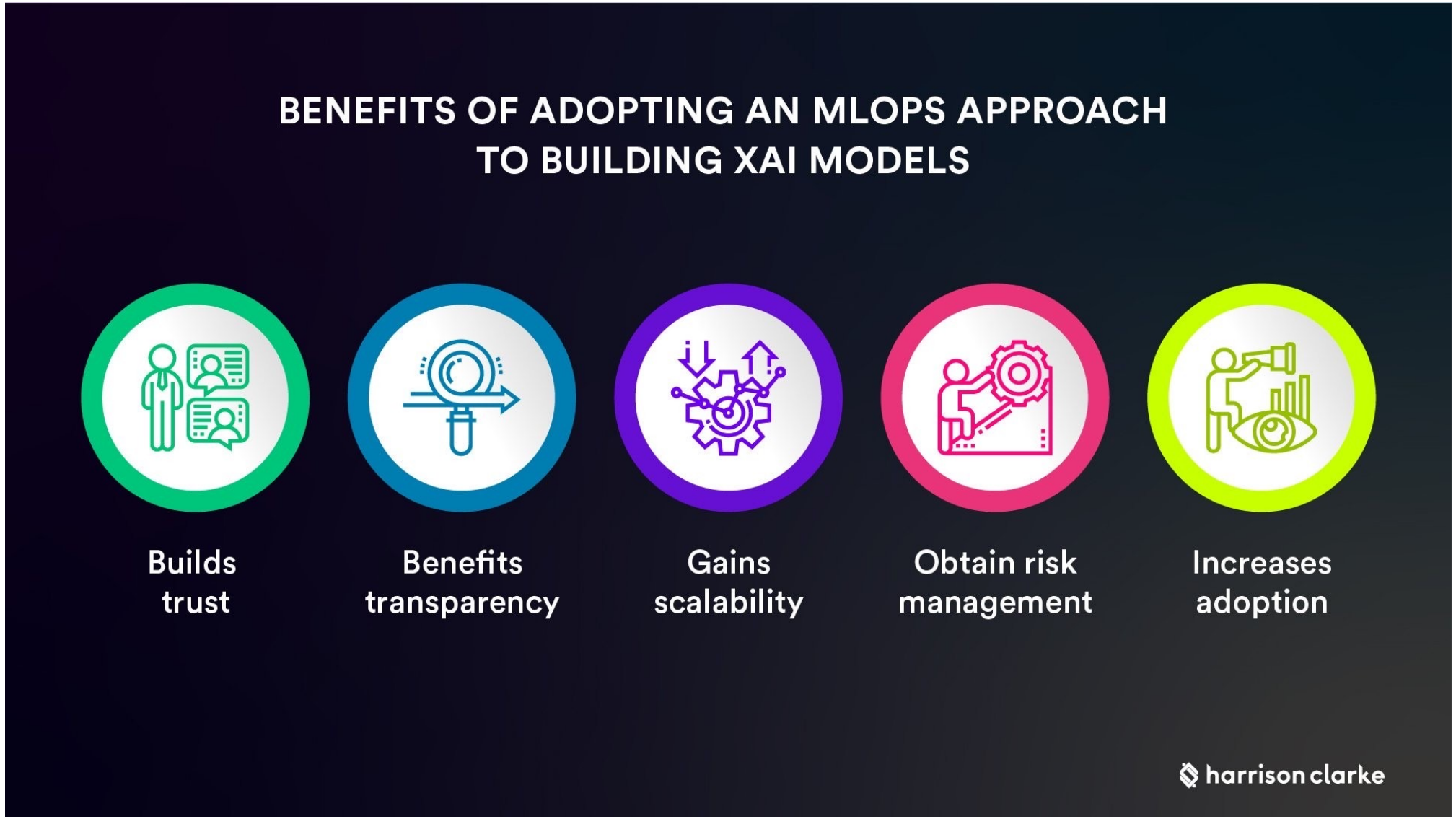
- **Model Distillation:** This approach simplifies a complex model by training a more interpretable one that mimics the predictions of the original model.

- **Linearity Assumptions:** This method is based on the idea that linear models are interpretable and can be used when linear relationships in the data are expected.



Source

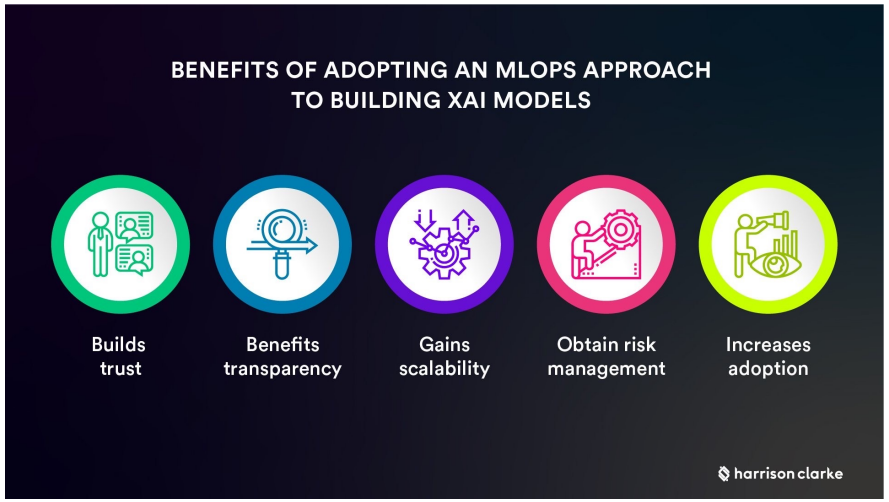
Benefits of MLOps in building XAI models



Source

Benefits of MLOps in building XAI models

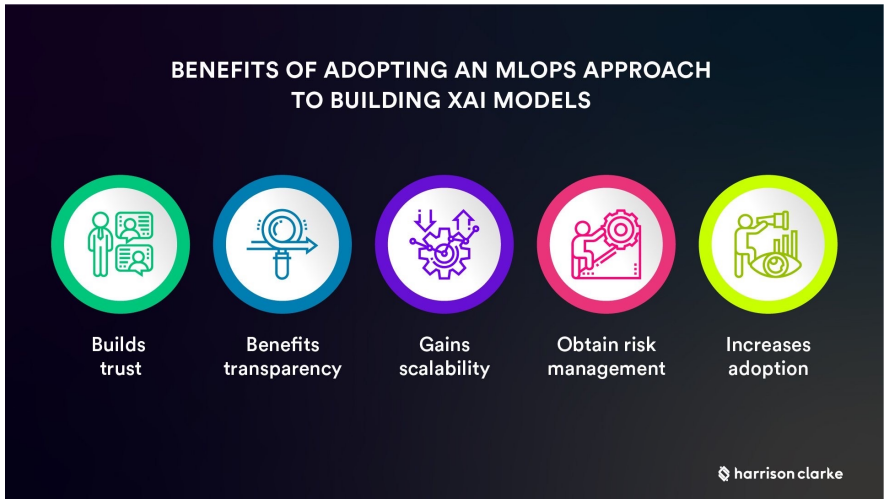
- Adopting an MLOps approach for building XAI models offers benefits beyond just explainability; it also includes scalability, reproducibility, and risk management.
- MLOps provides a comprehensive approach to developing and deploying machine learning models that can be tailored to various business needs.



Source

Benefits of MLOps in building XAI models

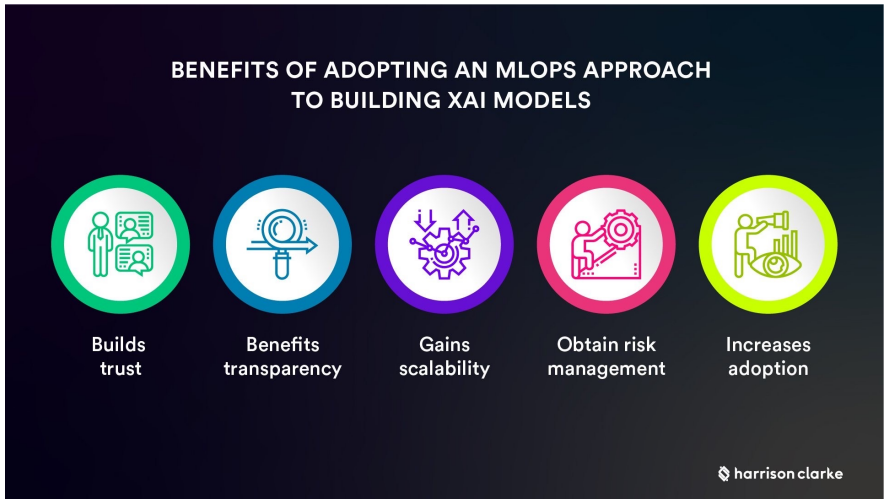
- Utilizing MLOps ensures that models adhere to ethical and legal standards, and the outcomes of the models are both auditable and transparent.



Source

Benefits of MLOps in building XAI models

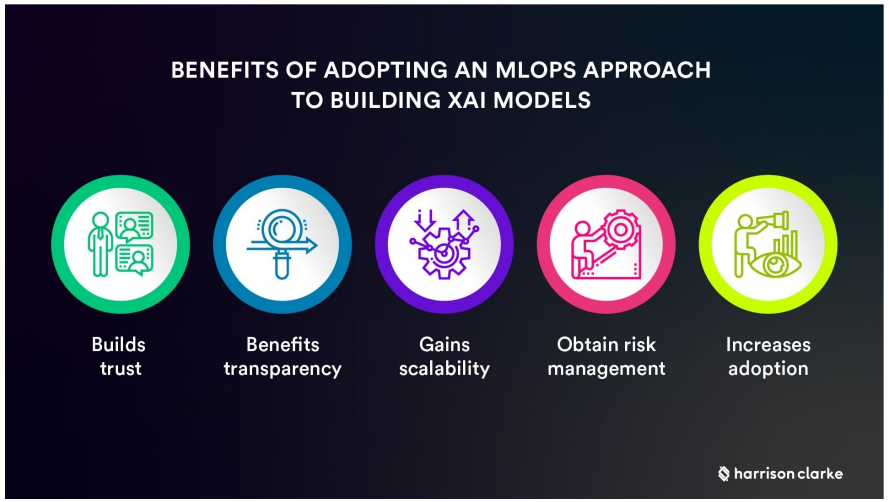
- Utilizing MLOps ensures that models adhere to ethical and legal standards, and the outcomes of the models are both auditable and transparent.



Source

Benefits of MLOps in building XAI models

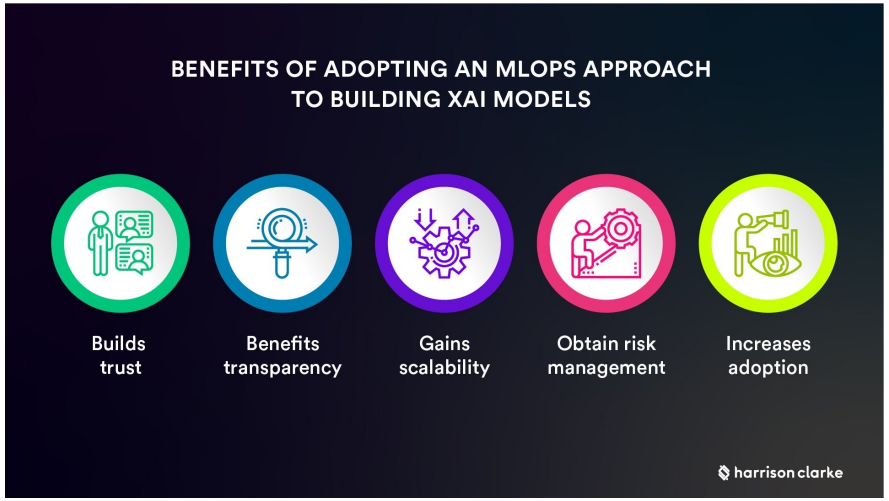
- The development of XAI is crucial for building trust in machine learning models and ensuring they are ethical, fair, and accountable.
- Incorporating MLOps in the development of XAI models can bridge the gap between machine learning developers and stakeholders by offering a systematic approach to their creation and deployment.



Source

Benefits of MLOps in building XAI models

- Organizations that adopt MLOps for building XAI models can enjoy numerous benefits, such as transparency, scalability, and risk management.
- By using an MLOps approach for XAI, organizations can foster trust in machine learning models, improve their performance, and boost their adoption across various industries.



Source

References:

- https://dagshub.com/glossary/mlops-monitoring/
- https://www.abtasty.com/blog/deployment-strategies/#:~:text=In%20that%20sense%2C%20a%20deployment,available%20to%20its%20intended%20users.
- https://www.harrisonclarke.com/blog/the-role-of-mlops-in-explainable-ai-use-cases-and-approaches
- https://www.qwak.com/post/top-ml-model-monitoring-tools