

THANK YOU FOR ENROLLING WITH US!!

JOIN THE TRIBE OF MLOPS AND ENGAGE WITH INSTRUCTOR ON LIVE SESSIONS FOR FREE !!



Manifold AI Learning - Workshops

"Unleash the Power of AI with Manifold Learning Workshops - Transforming Minds, One Workshop at a Time!"

Language: English

Instructors: Manifold AI Learning

Price (Limited Period Offer) :

ACCESS THE WORKSHOPS NOW

Link - <https://www.manifoldailearning.in/courses/Webinar-Learning-Materials--Slides-64df2bc5e4b0267d331f9998>

About Instructor

- **Nachiketh Murthy** is a MLOps Architect working in a Top Consulting Firm and sharing his knowledge through Manifold AI Learning. His Day-to-Day work includes:

- Leading a Team of 25 People and is responsible for End-to-End Implementation of Data Science, Machine Learning & Deep Learning Projects
- Manage the End-to-End Pipeline starting from Experimentation to Operationalization of Machine Learning Models in the Production Environment, including the maintenance of Machine Learning Models using DevOps Principles.
- Worked on Various domains like – Retail, Healthcare, Banking & Insurance.

- **Impacts Made:**

- Successfully Led & delivered multiple Data Science Projects from POC to Production with overall revenue close to 100m\$
- Leader on MLOps Approach in his Organization
- Author of 10 Top selling e-learning content on Data Science & MLOps
- Trained more than 50k learners through Live Bootcamps
- Trained more than 200k+ learners through self paced courses

Connect with Instructor

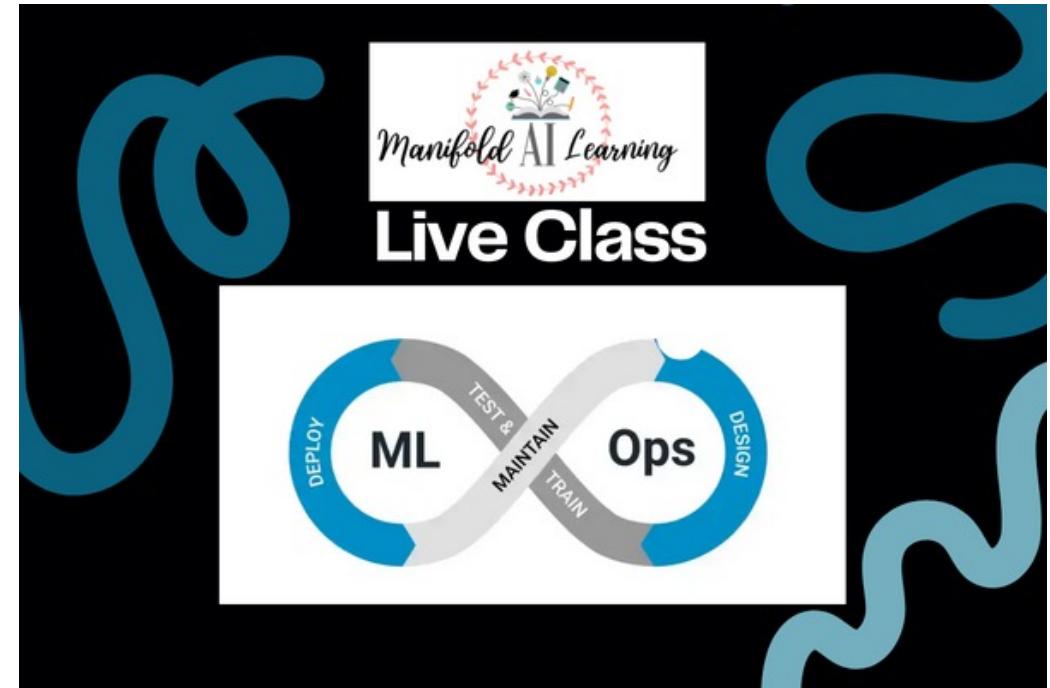


Nachiketh Murthy

<https://www.linkedin.com/in/nachiketh-murthy/>

Learn with Expert Live

Bootcamp for Limited Learners



[Link - https://www.manifoldailearning.in/courses/MLOps---LLMOps-AIOps-Bootcamp---A-Private-Community-65c63808e4b09cf855132de4](https://www.manifoldailearning.in/courses/MLOps---LLMOps-AIOps-Bootcamp---A-Private-Community-65c63808e4b09cf855132de4)

Python for MLOps

FOUNDATION OF PYTHON
PROGRAMMING FOR MLOPS

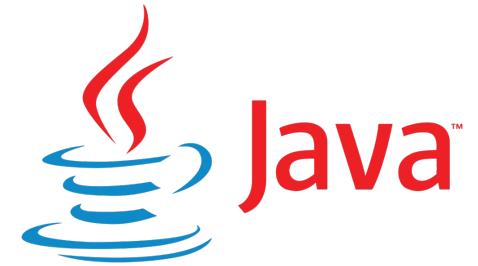
Manifold AI Learning - Registered Learners Only

How Do You Think
It Work ?



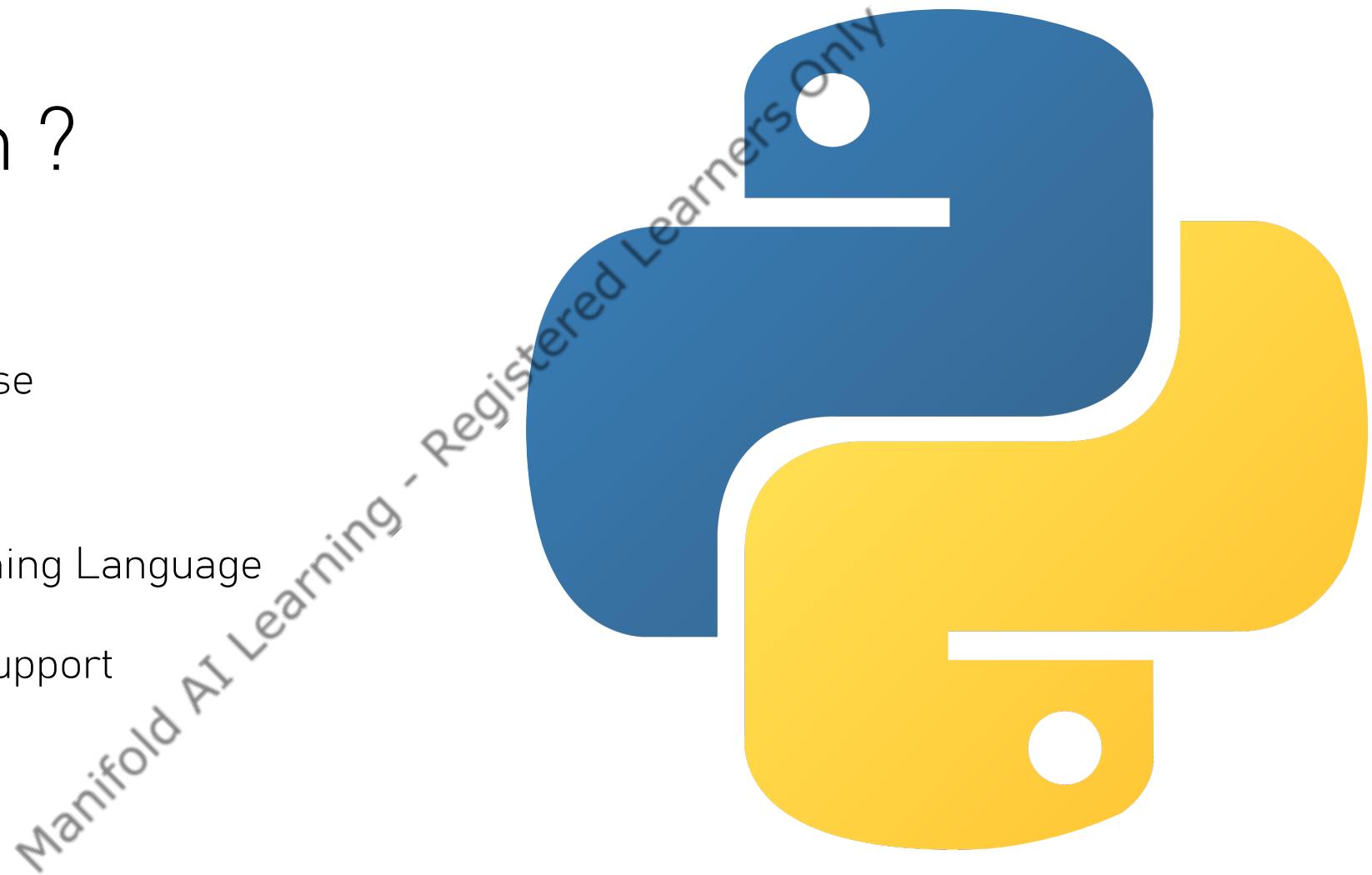
Programming Language

A way to interact with computer

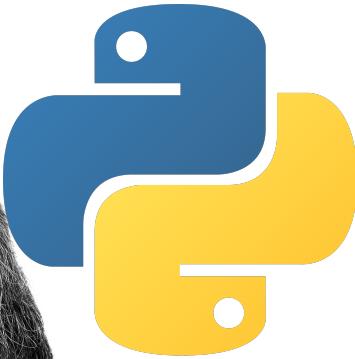


Why Python ?

- Easy to Learn and Use
- Popular
- Powerful Programming Language
- Extensive Library Support
- Huge Community



What is Python?



- Powerful general-purpose programming language.
- Created by Guido Van Rossum in 1991
- A Beginner-friendly Language which is easy to learn & read compared to other programming languages
- Named after a sitcom – Monty Python Flying Circus

Companies Using Python



- Python is used by Intel, IBM, NASA, Pixar, Netflix, Facebook, JP Morgan Chase, Spotify and a number of other companies.
- It's one of the four main languages at Google, and Google's YouTube is largely written in Python.
- Reddit, Pinterest & Instagram is also largely written in Python

Getting System Ready

INSTALL JUPYTER NOTEBOOK

Manifold AI Learning - Registered Learners Only

Variables in Python

Manifold AI Learning - Registered Learner

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object

operation = "MIRROR_X"
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False

operation = "MIRROR_Y"
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False

operation = "MIRROR_Z"
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add
modifier.select= 1
modifier.select=1
context.scene.objects.active
("Selected" + str(modifier))
modifier.select = 0
bpy.context.selected_objects
data.objects[one.name].select=1
print("please select exactly one object")
- OPERATOR CLASSES -->
types.Operator):
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
context):
    context.active_object is not None
MANIFOLD AI LEARNING ®
```

Variables in Python

- Variables are containers for storing the data values, with which we can further reference and manipulate the values.

A = 10

• B = 5

• C = A + B i.e C = 15

• Variable_name = Value_to_be_assigned

Rules of Creation of Variables



- Variable name Should start with letter or an underscore



- Variable Name can be alphanumeric



- Variable names are case sensitive



- Variable name should not be a **reserved word**

Literals

- " Literals are values used in Python. "
- Types are: Numeric, Boolean, Collection, String, Special

Operators in Python

- Arithmetic Operators
- Comparison Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Identity Operators
- Membership Operators

Manifold AI Learning - Registered Learners Only

Keywords in Python

- Reserved words with Pre-defined value

```
: import keyword
print(keyword.kwlist)

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from',
'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

LEARNING WITH CODING

Hands On

Manifold AI Learning - Registered Learners Only

Control Statements in Python

- Conditional Statements
- Looping Statements

Manifold AI Learning



Functions in Python

- Group of Statements that performs a specific task.

syntax

```
1. def function_name(parameters):  
2.     """docstring"""  
3.     statement(s)
```

Modules in Python

- A Module is a file containing the python definitions & statements
- Modules are the files with .py extensions
- Helps us to logically organize the code

Classes and Objects in Python

- Object Oriented Programming

Manifold AI Learning - Registered Learners Only



Understand Terminology

Classes

A Class is a user-defined blueprint or prototype from which objects are created.

Class Attributes

Reflects the properties of an Objects

Objects

An instance of a class – with values

Class Methods

Reflects the Behavior of Objects

Manifold AI Learning - Registered Learners Only

Libraries in Python

Manifold AI Learning - Registered Learner

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object

operation = "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False

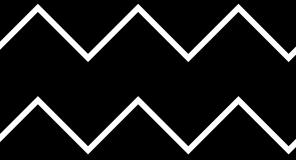
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False

operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

selection at the end -add
modifier.select= 1
modifier.select=1
context.scene.objects.active
("Selected" + str(modifier))
modifier.select = 0
bpy.context.selected_objects
data.objects[one.name].select

print("please select exactly one
operator class")
- OPERATOR CLASSES -->

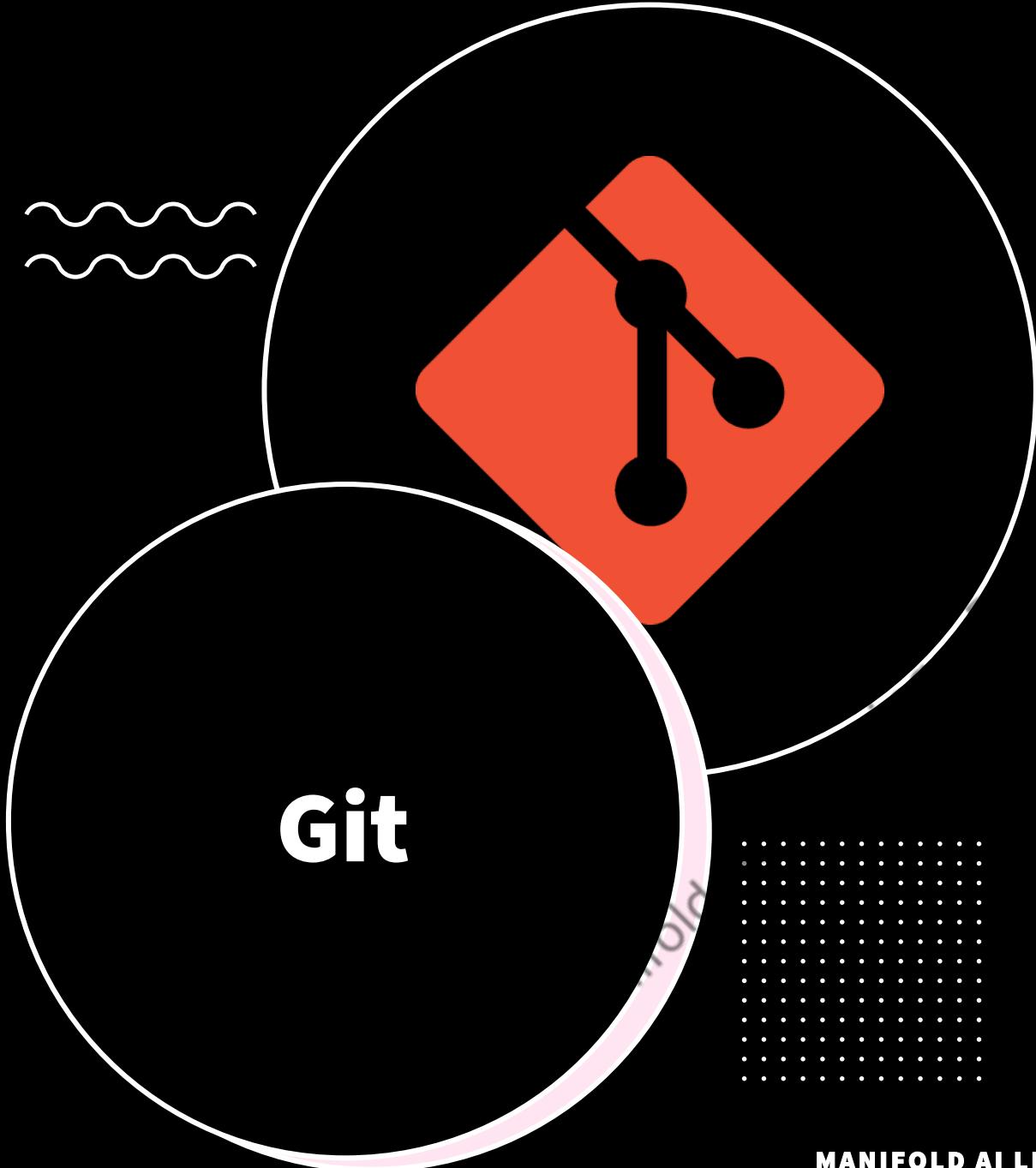
types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    for X"
context):
    context.active_object is not
MANIFOLD AI LEARNING ®
```



GIT FOR MLOPS

LEARNING FROM SCRATCH

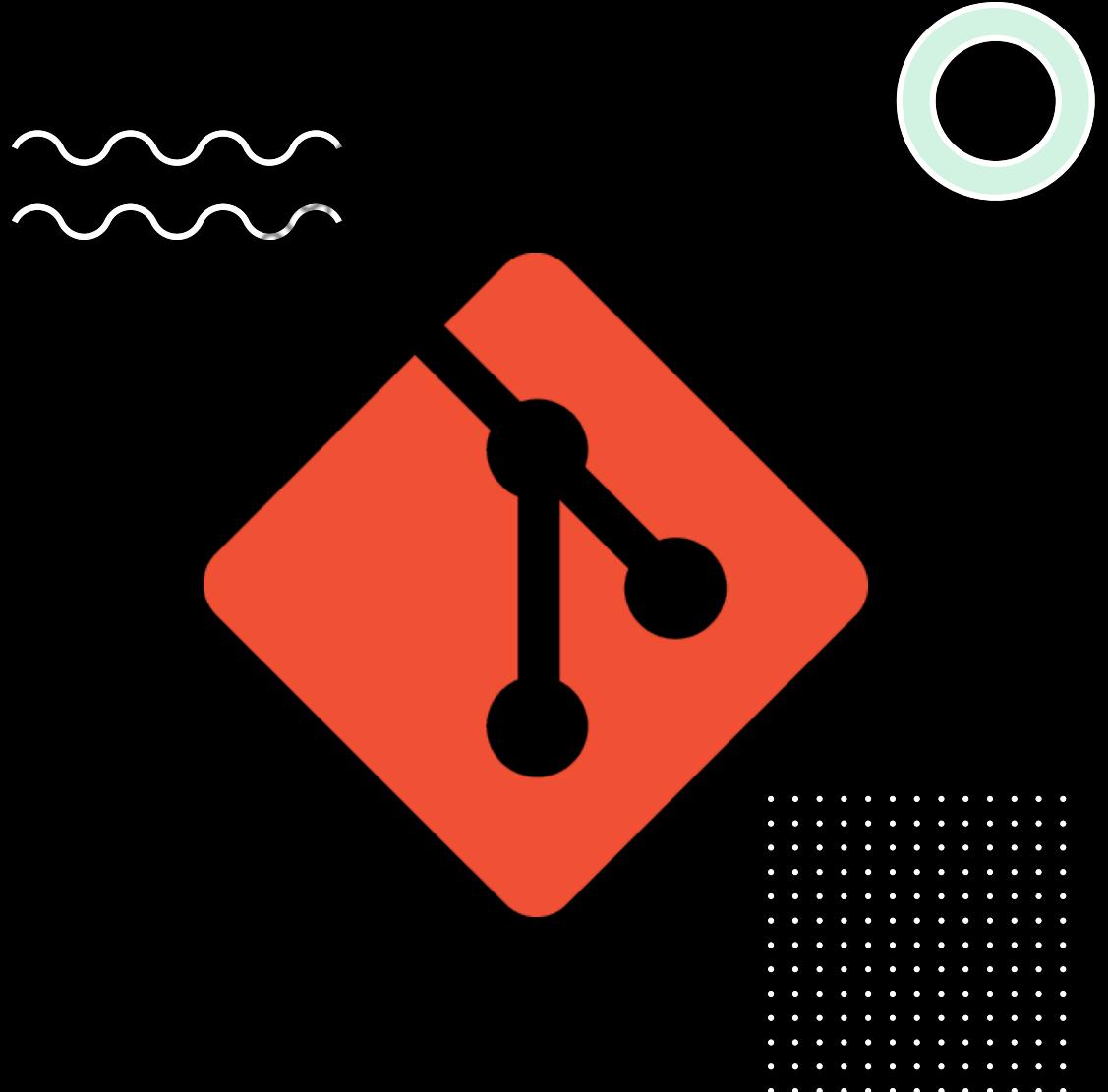




- **Git is an open-source Distributed Version Control System (DVCS).**
- **A version control system allows you to record changes to files over a period.**
- **Git is used to maintain the historical and current versions of source code.**

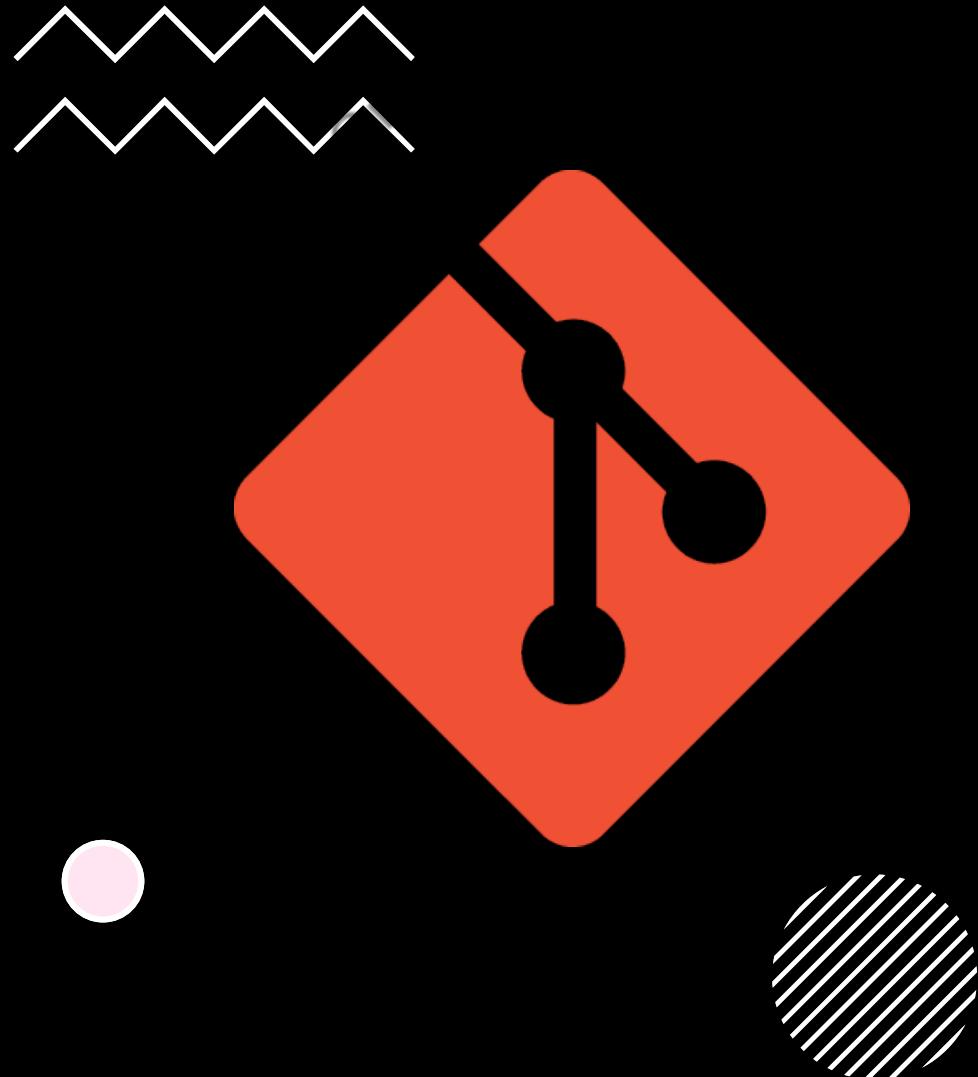
Usage at a High level

- In a project, developers have a copy of all versions of the code stored in the central server.
- Git allows developers to do the following:
 - Track the changes, who made the changes, and when
 - Rollback/restore changes
 - Allow multiple developers to coordinate and work on the same files
 - Maintain a copy of the files at the remote and local level



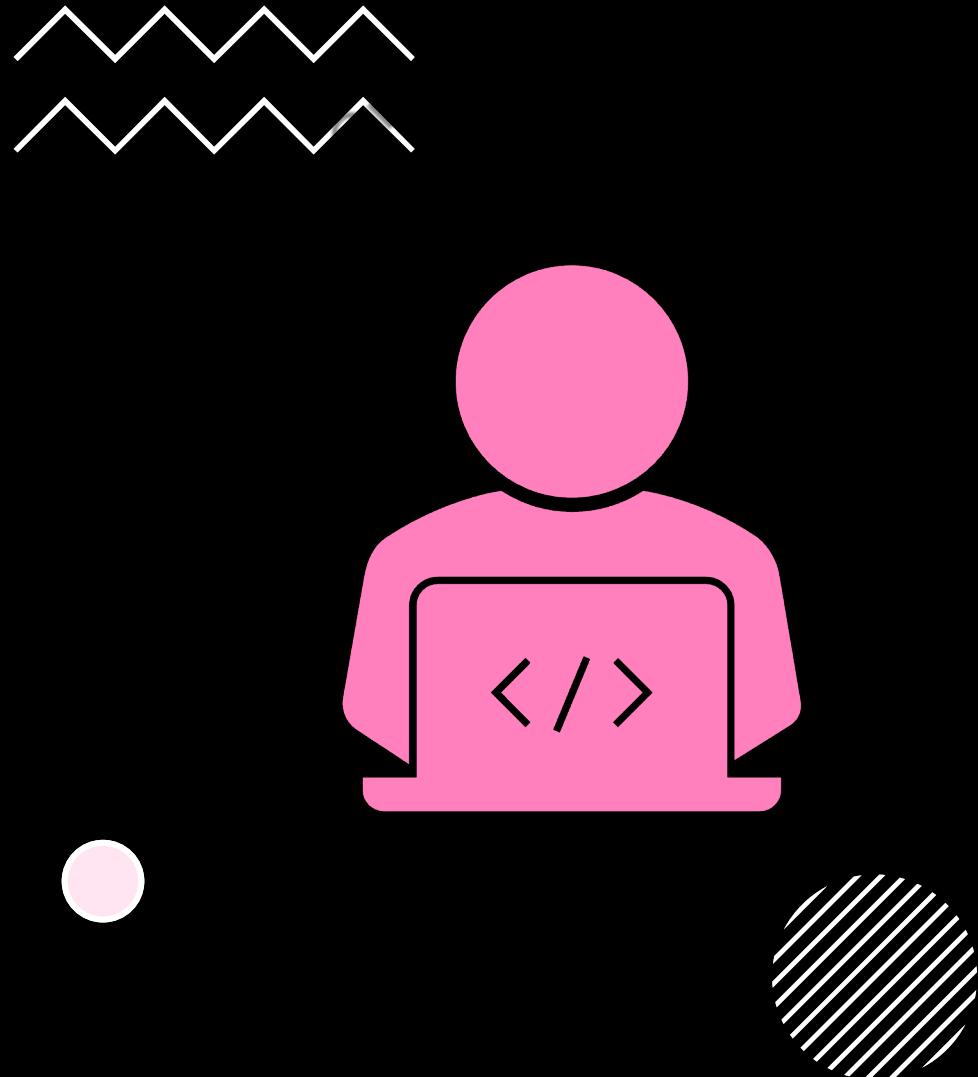
What is Git ?

- **Git is a technology designed for tracking changes in a project and facilitating collaboration among multiple contributors.**
- **A Git-controlled project at its core consists of a folder containing files, with Git monitoring changes made to these files.**
- **Git's primary function is to enable the storage of various versions of project work, earning it the title of a version control system.**



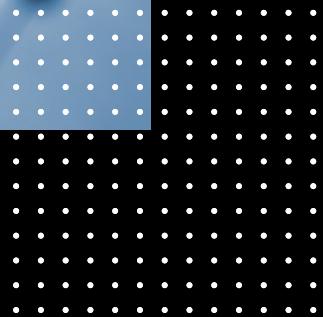
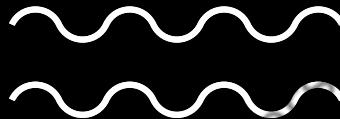
Role of Git in MLOps

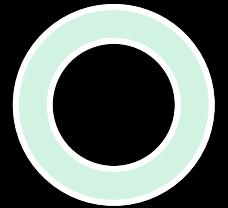
- **Version Control:**
 - **Git plays a crucial role in MLOps by providing version control for machine learning models, code, and configuration files. This ensures traceability and reproducibility of experiments and model deployments.**
- **Collaboration and Team Workflow:**
 - **Git facilitates collaboration among data scientists, engineers, and other team members, allowing them to work concurrently on different aspects of the machine learning pipeline. It enables seamless integration of changes and helps manage collaborative development efforts.**



Role of Git in MLOps

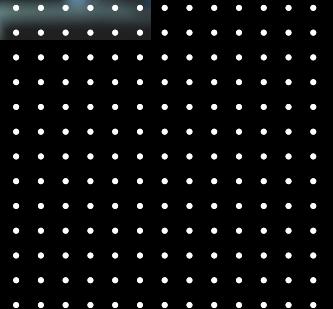
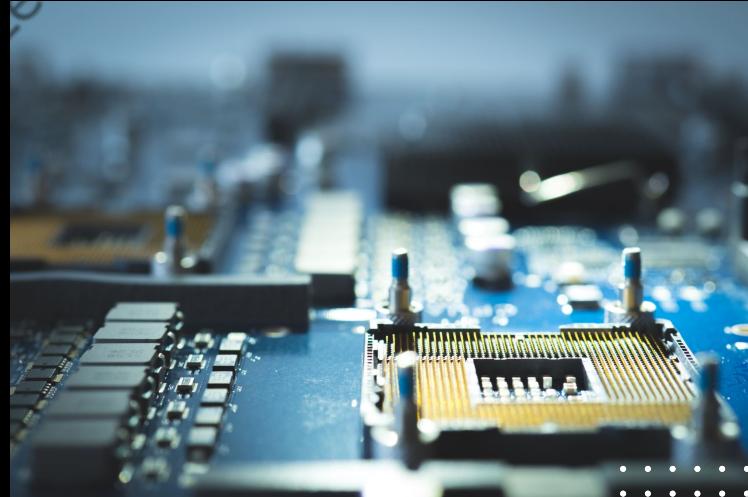
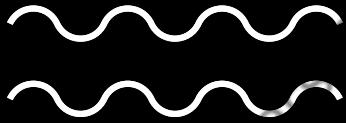
- **Experiment Tracking:**
 - **Git, in conjunction with tools like Git-based platforms or MLflow, helps track and manage machine learning experiments. This includes recording parameters, metrics, and code versions, making it easier to reproduce and compare results.**
- **Branching for Experimentation:**
 - **Git's branching feature allows for the creation of isolated branches to experiment with different model versions or parameter settings. This supports a systematic approach to testing and refining models without affecting the main codebase.**





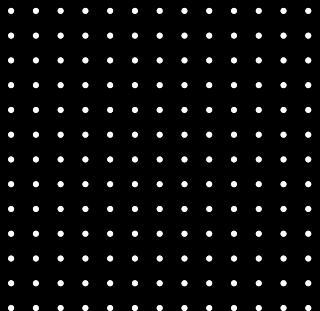
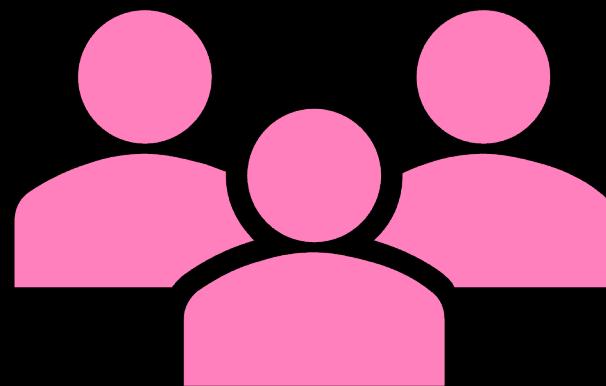
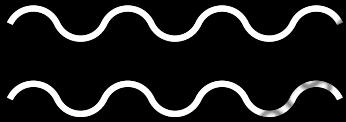
Role of Git in MLOps

- **Continuous Integration (CI) and Continuous Deployment (CD):**
 - **Git integrates with CI/CD systems to automate the testing and deployment of machine learning models. This ensures that changes to the codebase are validated, and successful builds trigger automated deployment pipelines for model updates.**
- **Infrastructure as Code (IaC):**
 - **Git is used to version control infrastructure configurations as code, ensuring consistency between development, testing, and production environments. This is particularly relevant in MLOps, where the deployment infrastructure is a critical component.**



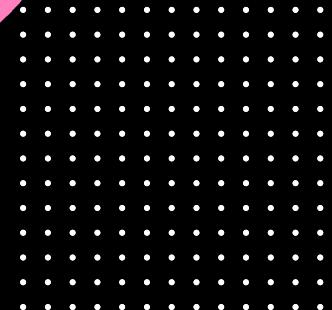
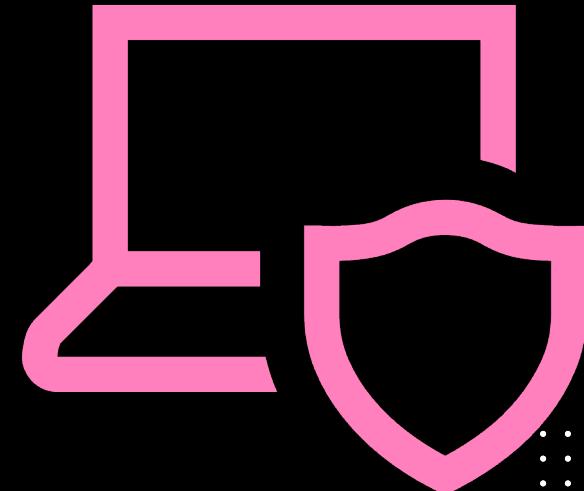
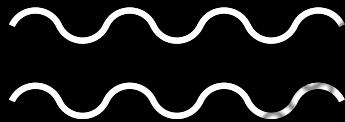
Role of Git in MLOps

- **Collaboration Across Teams:**
 - In MLOps, where teams may include data scientists, engineers, and operations professionals, Git serves as a common platform for collaboration, enabling effective communication and coordination across different stages of the machine learning lifecycle.
- **Artifact Management:**
 - Git is used in conjunction with artifact repositories to manage and version control artifacts such as trained models, datasets, and other dependencies. This ensures that all components of a machine learning project are tracked and reproducible.

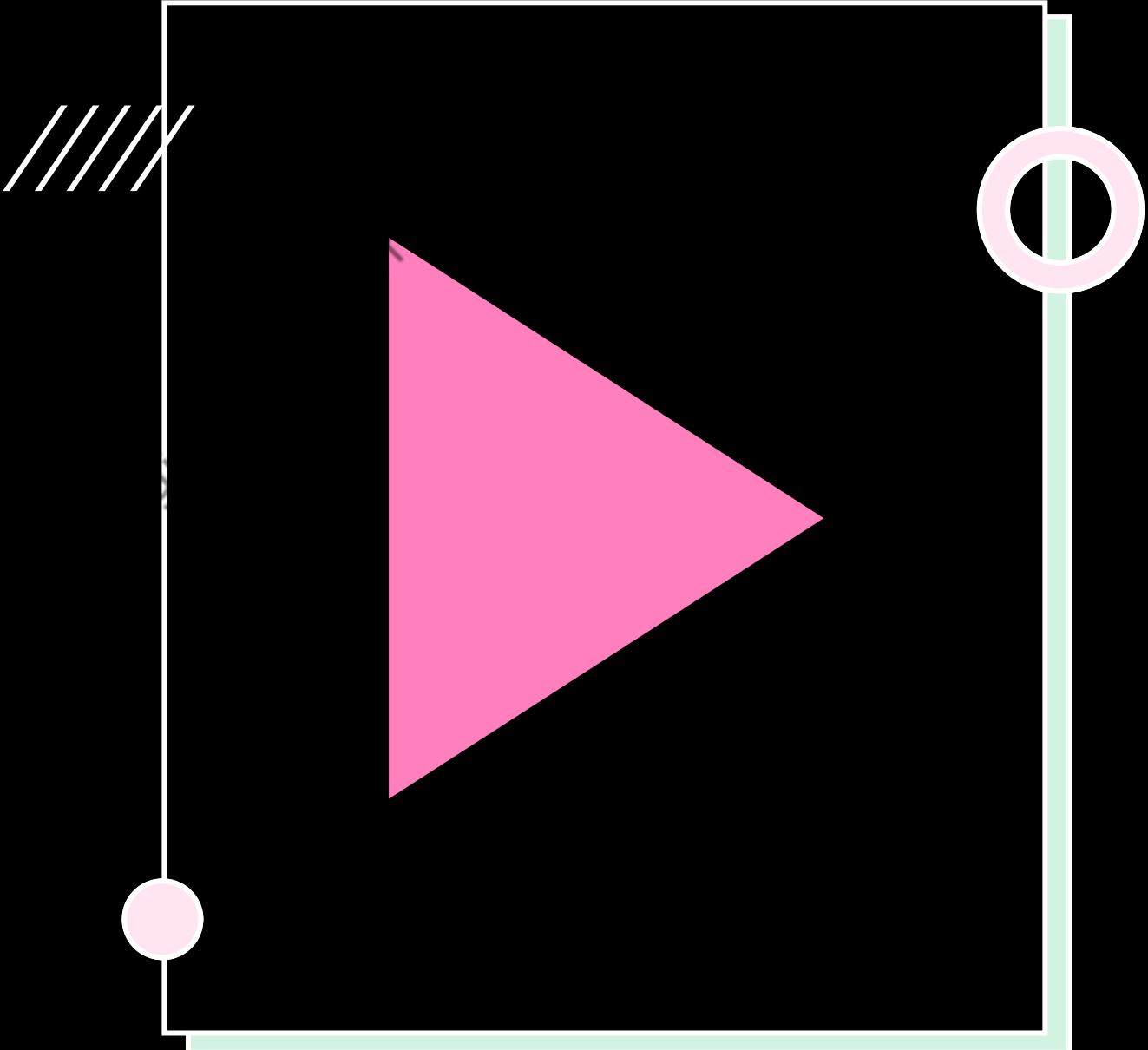


Role of Git in MLOps

- **Auditability and Compliance:**
 - **Git's commit history provides an audit trail for all changes made to the codebase and models. This is crucial for compliance purposes, allowing organizations to trace back and understand the evolution of models and associated code.**



**L E T ' S
C O N T I N U E
I N N E X T
V I D E O**



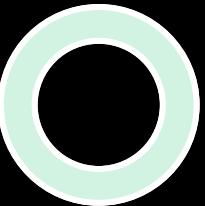
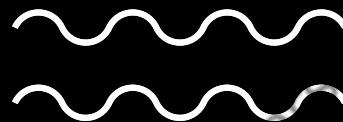
- Getting System Ready

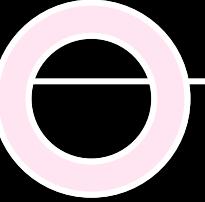
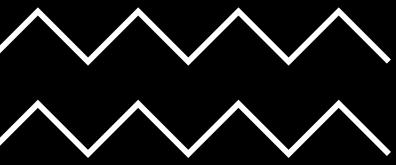
- <https://git-scm.com/download/win>



Install VS Code

- <https://code.visualstudio.com/download>





Working with Repositories

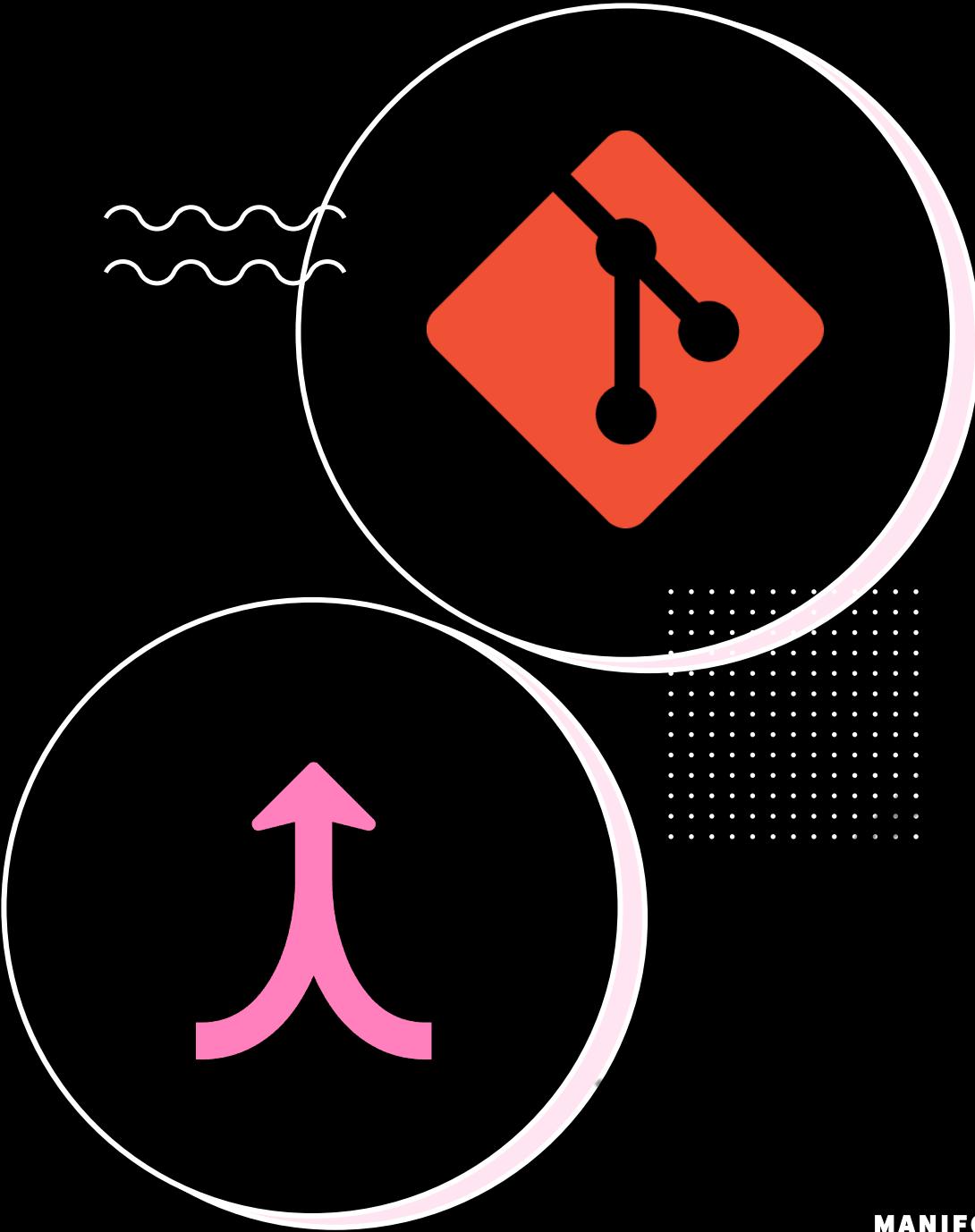




Setting Git Configuration

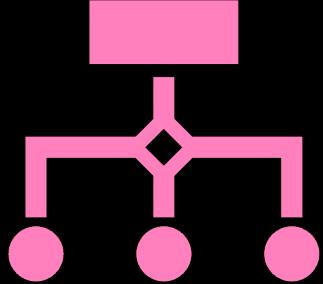
- **Git configurations are settings that allow you to customize how Git works.**
- **They consist of variables and their values, and they are stored in a couple of different files.**
- **To work with Git, you must set a few configuration variables related to user settings.**



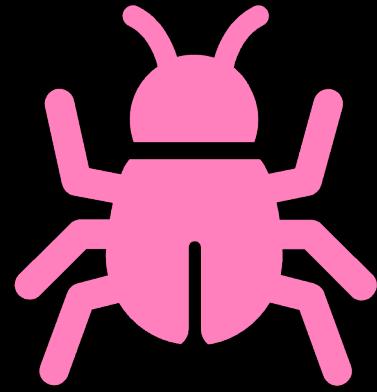


Git Merging

• What is Merging in Git ?

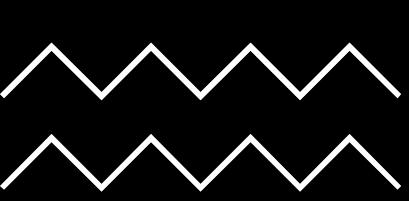


In Git, merging refers to the process of combining changes from different branches.

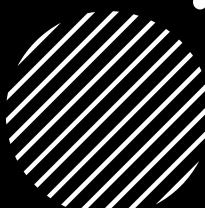


When you work on a project with multiple collaborators or when you're managing different features or bug fixes in separate branches, you may need to merge those changes back into the main branch or another target branch.





Types of Merge



MANIFOLD AI LEARNING®

Fast-forward Merge:

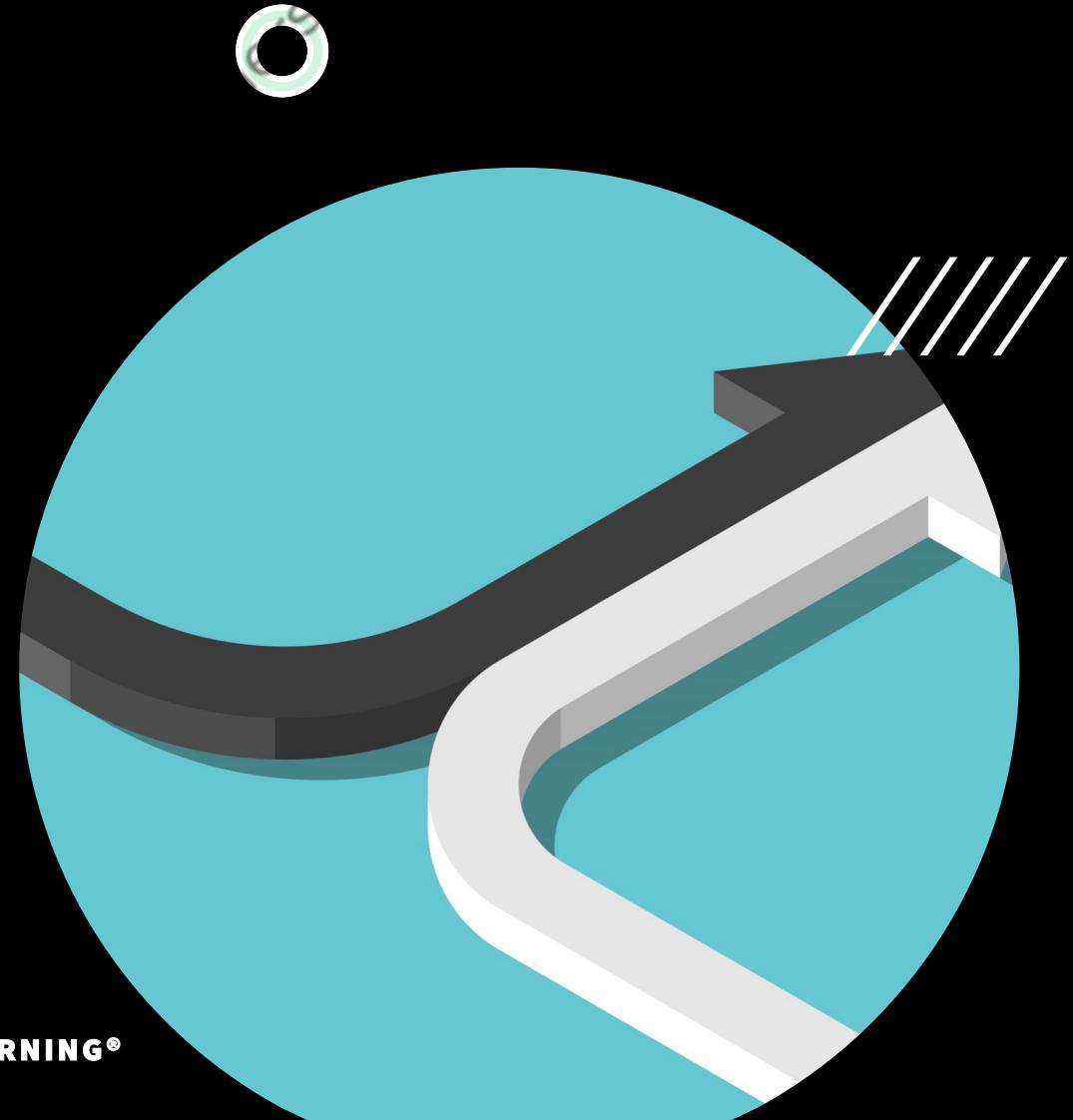
- This occurs when the branch being merged has no new commits that the branch it is merging into doesn't already have.
- In a fast-forward merge, Git simply moves the branch pointer forward, and no new merge commit is created.

Three-way Merge:

- This occurs when there are divergent changes in both the source and target branches.
- Git creates a new commit, known as a "merge commit," that has two parent commits—one from the source branch and one from the target branch.
- The merge commit represents the combination of changes from both branches.



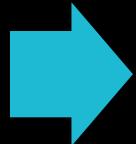
**H A N D S
O N -
F A S T
F O R W A R D
M E R G E**





Checking Out Commits

Checking out commits refers to the process of switching your codebase to a specific commit in a version control system.



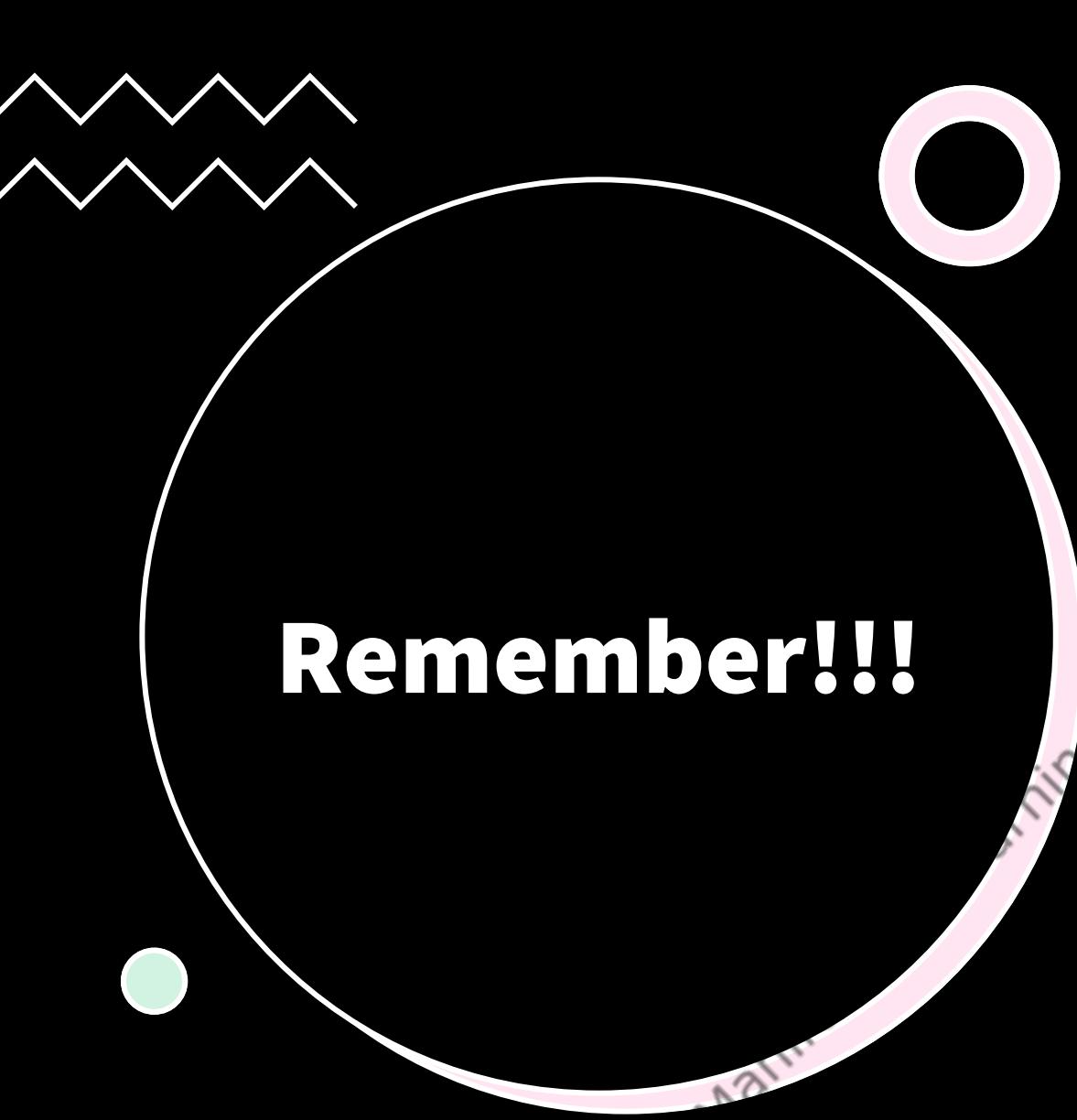
When you check out a commit, you are essentially telling the version control system to set your working directory and codebase to the state it was in at the time of that specific commit.



Uses:

- Debugging
- Branching
- Reviewing History





Remember!!!

- **Always be cautious when checking out commits, especially if you have uncommitted changes in your working directory, as Git may require you to either commit, stash, or discard those changes before proceeding with the checkout.**



WORKING WITH HOSTING SERVICES

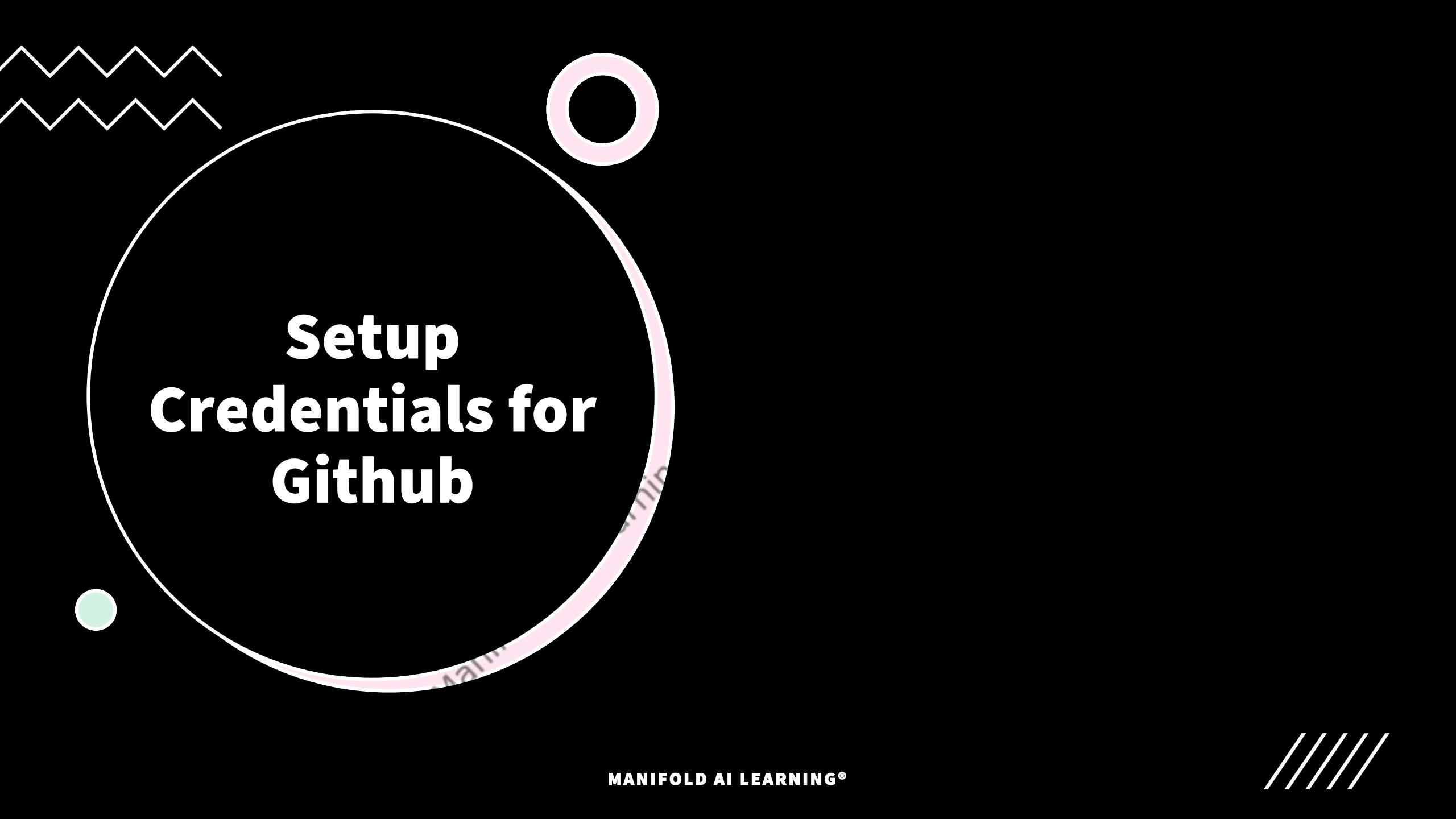




Git Hosting Services

- **Git hosting services are platforms that provide infrastructure and tools for hosting, managing, and collaborating on Git repositories.**
- **These services make it easier for individuals and teams to use Git for version control and collaborative software development.**
- **Examples : GitHub , GitLab, Bitbucket, Azure DevOps Services, SourceForge, GitKraken, AWS CodeCommit, etc.**





Setup Credentials for Github



WORKING WITH REMOTE REPOSITO RIES





Benefits of Remote Repositories

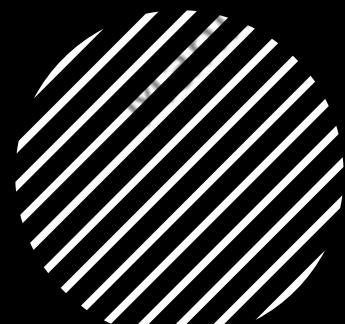
Collaboration

Backup and Redundancy

Access Control

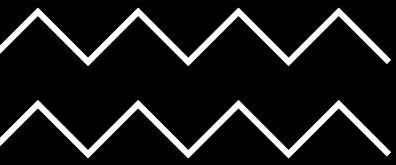
CI CD Facilitation

Versioning and History



3 WAY MERGE IN GIT

MANIFOLD AI LEARNING®

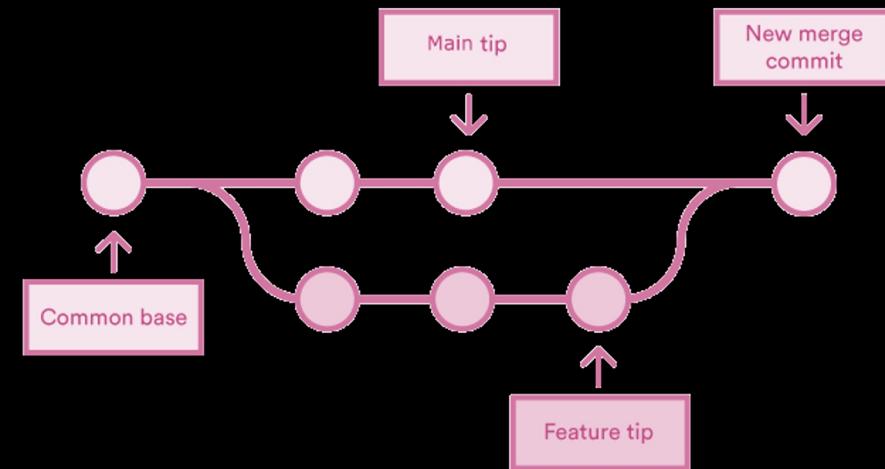
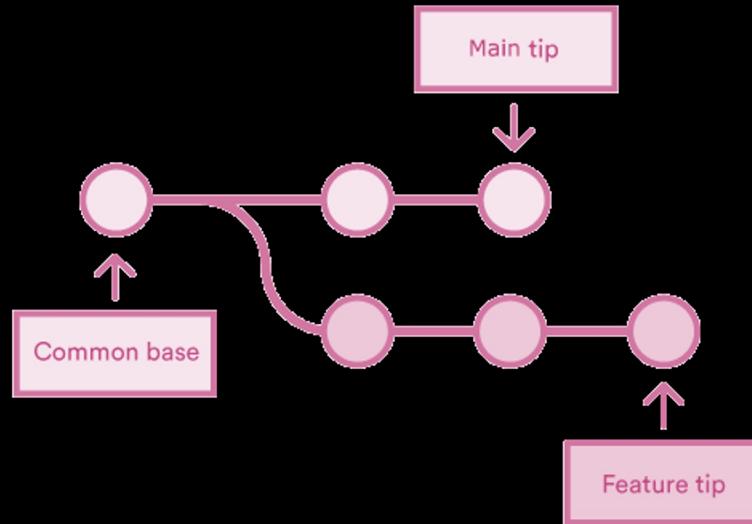


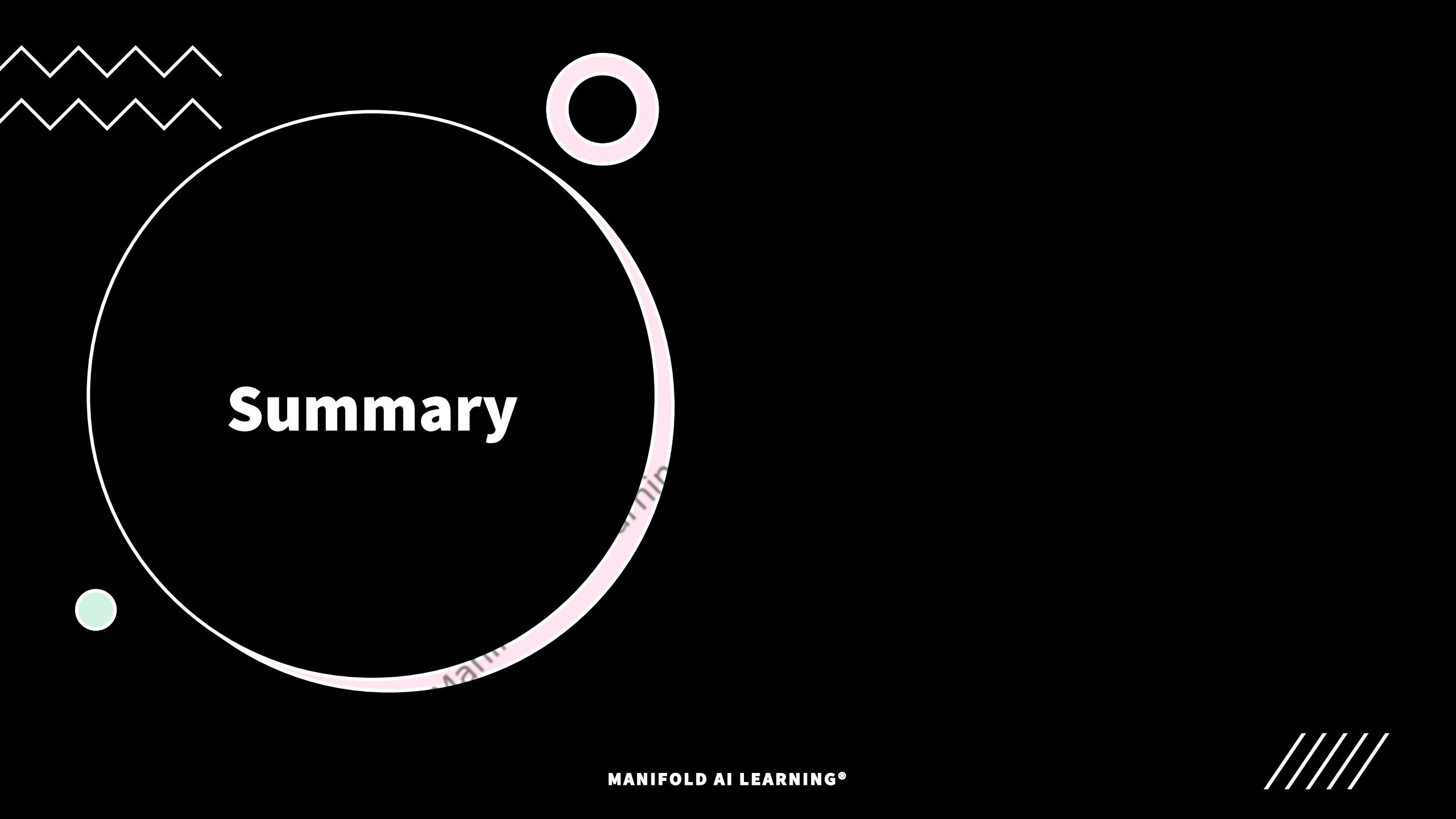
• What is 3-way Merge ?

- **Algorithm that Git uses to combine changes from two different branches.**
- **This algorithm is also known as a "three-way merge" because it involves three commits:**
 - **Common Ancestor Commit (Base Commit): The commit where the branches diverged. It represents the last commit that both branches had in common before they started to diverge.**
 - **Branch A Commit: The commit representing the tip (latest commit) of one branch.**
 - **Branch B Commit: The commit representing the tip of the other branch.**



Example





Summary

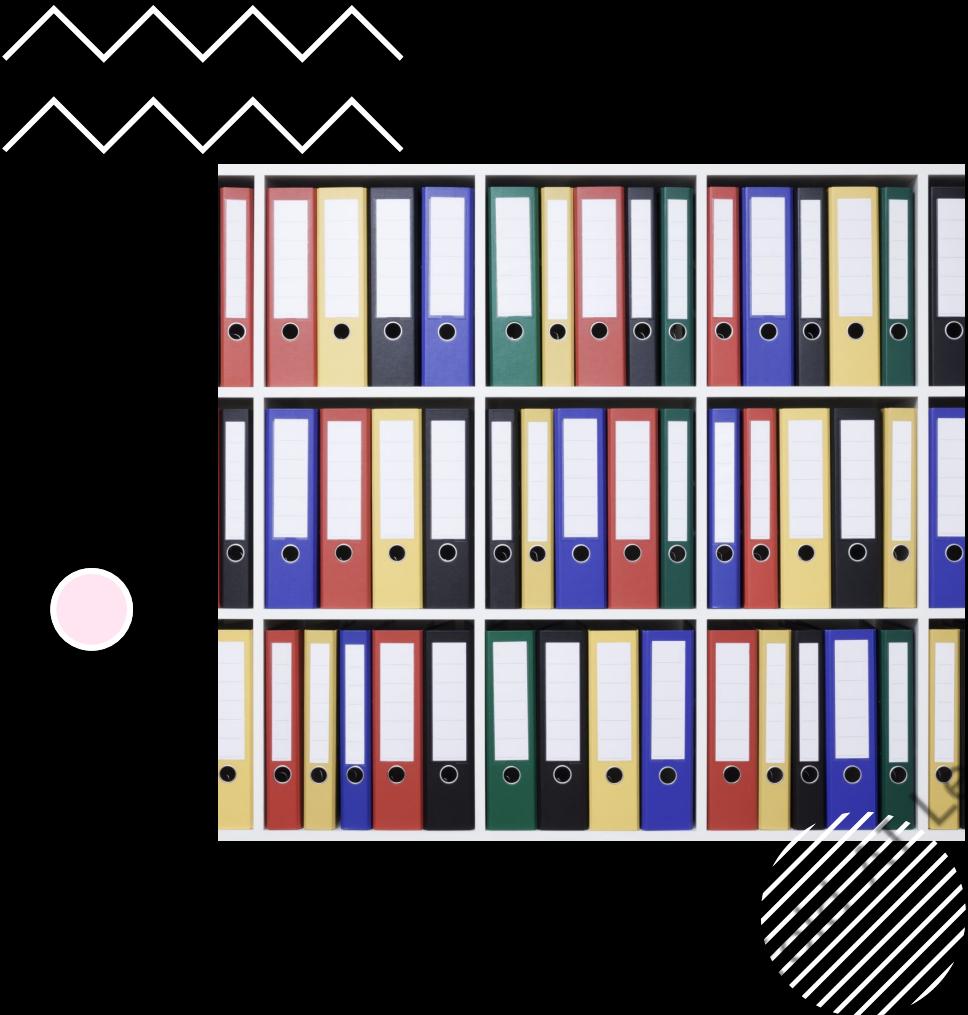


Summary

- **Introduction to Version Control Systems:** Version Control Systems (VCS) manage changes to source code, tracking modifications, and facilitating collaboration among developers. They help organize project history and enable teams to work concurrently.
- **Getting Started with Git:** Git is a distributed version control system widely used for source code management. It allows for efficient tracking of changes, collaboration, and branching. Git is known for its speed, flexibility, and decentralized architecture.
- **Local Repo vs Remote Repo:** A local repository is on a user's machine, while a remote repository resides on a server. Developers work locally, commit changes to the local repo, and then push or pull changes from the remote repo to collaborate with others.
- **Git Configurations:** Git configurations set user details, preferences, and repository settings. Users configure Git globally or per repository to customize their experience.

Summary

- **Concept of Working Directory - Staging Area - Commit:** Git operates with three main areas: the working directory (where files are edited), the staging area (where changes are prepared for commit), and commits (snapshots of changes with a commit message).
- **Git Workflow - Local Repo:** The Git workflow involves creating branches for different features or bug fixes, making changes, committing them, and merging them back into the main branch (usually 'master' or 'main').
- **Git Workflow:** The general Git workflow encompasses creating branches, making changes, staging and committing, merging branches, and resolving conflicts. This iterative process ensures organized and collaborative development.
- **Git Branch:** Branches in Git allow developers to work on isolated features or bug fixes without affecting the main codebase. They enable parallel development and easy integration of changes.





Summary

- **Switching the Branches:** Git provides commands to switch between branches, allowing developers to focus on specific features or bug fixes. This flexibility supports efficient project management.
- **Merging:** Merging combines changes from different branches into a single branch. Git's three-way merge algorithm intelligently incorporates changes from a common ancestor, preventing conflicts when possible.
- **Checking Out Commits:** Checking out commits in Git allows developers to view or work on specific snapshots of the project history. This can be useful for debugging or understanding the code at a specific point in time.
- **Git Hosting Services:** Git hosting services like GitHub, GitLab, and Bitbucket provide platforms for hosting and collaborating on Git repositories. They offer features such as issue tracking, code review, and integration with CI/CD.



Summary

- **Working with Remote Repositories:** Interacting with remote repositories involves pushing changes to share with others and pulling changes to update the local repository. Collaboration is facilitated through remote repositories hosted on platforms like GitHub.
- **Cloning and Delete Branches:** Cloning a repository creates a local copy, allowing developers to work on their machines. Deleting branches is part of Git maintenance, removing unnecessary branches after their changes are merged or discarded.
- **3-Way Merge:** The 3-way merge in Git involves combining changes from two branches and a common ancestor. This process helps automatically merge changes when possible, providing a structured approach to integration and conflict resolution.

PACKAGING ML MODELS

MLOps Bootcamp

Manifold AI Learning - Registered Leads Only

AGENDA

- Learn Modularization Approach for Code
- Virtual Environments
- Serialization and Deserialization of ML Models
- Packaging in Python
- Develop, Build and Deploy ML Packages

LET'S BEGIN!!

Experiment Phase

Manifold AI Learning - Registered Learners Only



DATASET :

LOAN

ELIGIBILITY

DATASET

- A company wants to automate the loan eligibility detection based on the customer details provided in Online Application form.
- We need classify each row as – whether a loan would be approved or not!!

EXPLORE THE DATA



LET'S START BUILDING ML MODEL

Manifold AI Learning - Registered Learners Only

MLflow

An open source platform for the
machine learning lifecycle

Machine Learning - Registered Learners Only



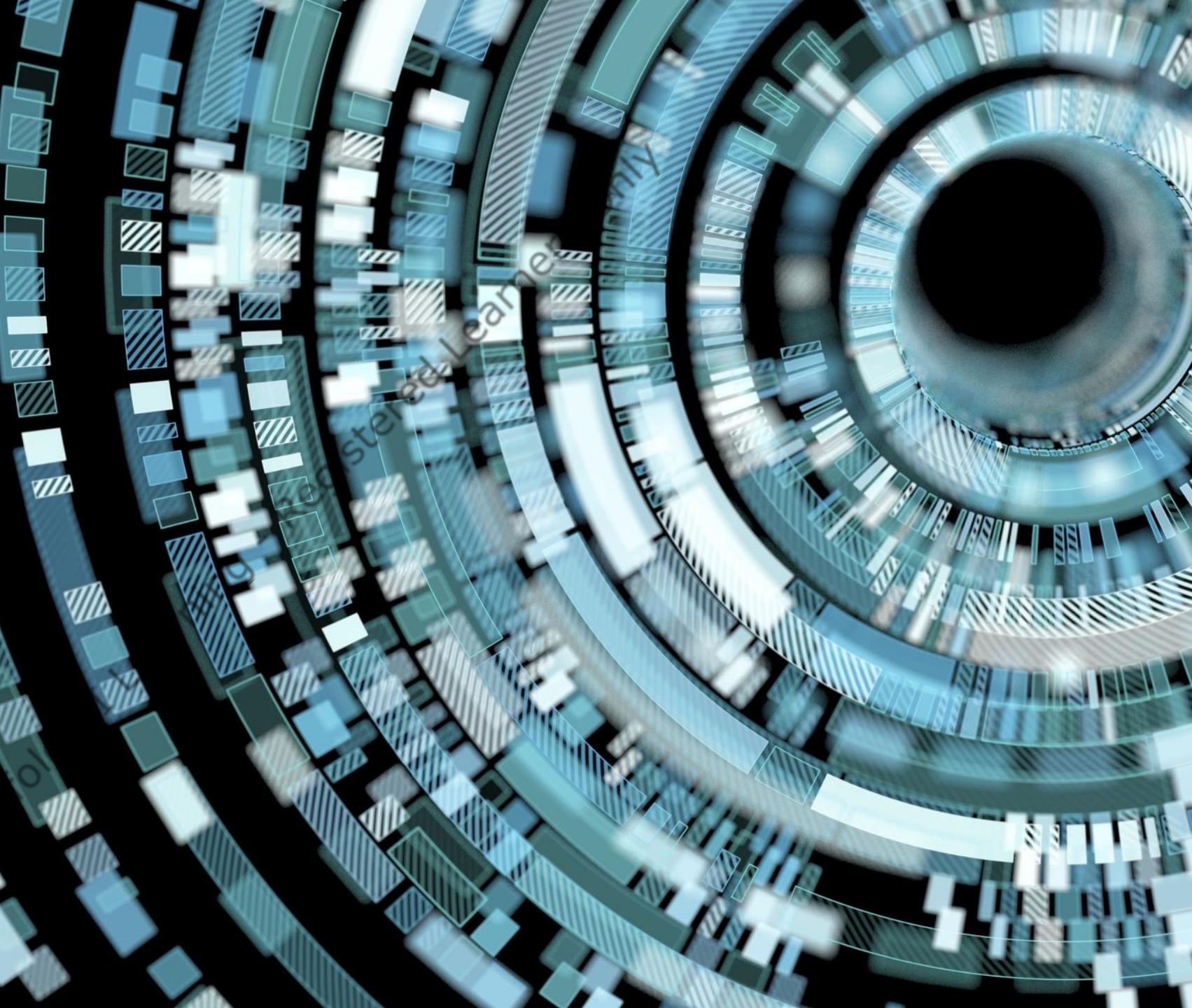
Journey so far!!

- We have modularized the code and created the package so that it can be sharable with all the dependencies!!

Manifold ML Learning - Registered Learners Only



But Machine
Learning is
not all about
the Software
Code!!





Key Things to Consider in ML :

- Its an Iterative Approach in creating the Model
- Involves Hyper parameter tuning to find the best model
- For Each change in Training data , new artifacts will be generated, and has to be tracked (Example: Model File, Preprocessor File, Validation file, etc.)
- Easy to lose the settings of Best Performing model if its not closely monitored



Solution : MLflow

A Platform to manage the complete ML Lifecycle

Manifold AI Learning™ Registered Learners Only

What is MLflow ?

- MLflow is an open source platform to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry.

Manifold AI Learning

Manifold AI Learning - Registered Learners Only



Components of MLflow

- **MLflow Tracking**
 - Record and query experiments: code, data, config, and results
- **MLflow Projects**
 - Package data science code in a format to reproduce runs on any platform
- **MLflow Models**
 - Deploy machine learning models in diverse serving environments
- **Model Registry**
 - Store, annotate, discover, and manage models in a central repository



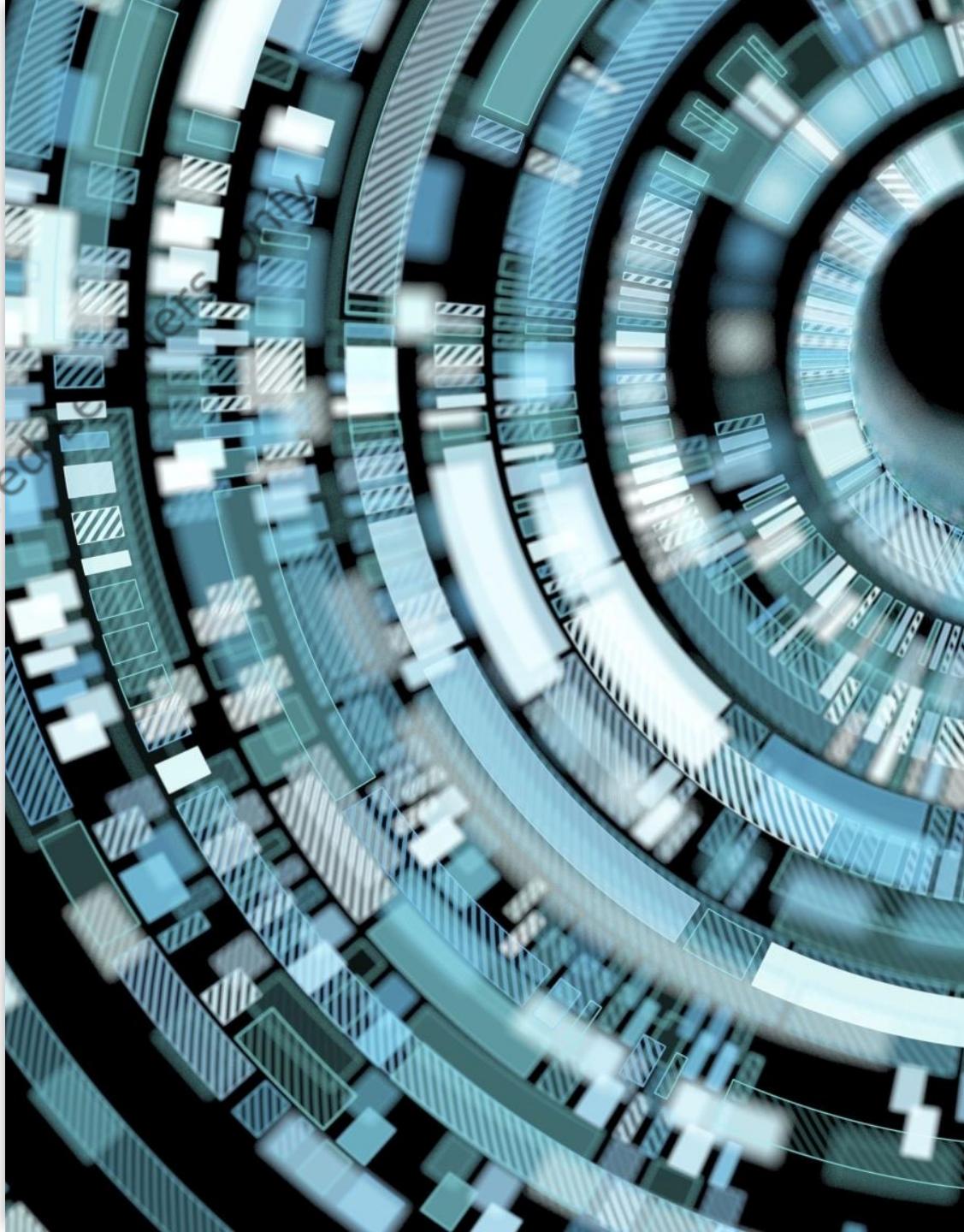
Features of MLflow

- MLflow offers data scientists the flexibility to conduct numerous experiments before moving a model to production.
- It diligently records crucial model evaluation metrics like RMSE and AUC, while also maintaining a log of the hyperparameters employed during model development.
- It facilitates the storage of the trained model in conjunction with its optimal hyperparameters.
- empowers users to seamlessly deploy machine learning models to production servers or cloud environments
- it allows for the monitoring of models in both staging and production, ensuring that all team members stay informed.



Is it compatible with my current workflow ?

- MLflow's versatility extends to its library-agnostic nature, making it compatible with various popular machine learning libraries
- MLflow supports multiple programming languages, accessible via REST API and a Command Line Interface (CLI).
- Integrating MLflow with your existing code is a straightforward process.



Use Case of MLflow



Comparing different models :

Using the mlflow UI we can compare multiple ML models side by side, along with their metric and parameter settings.



Cyclic Model deployment:

To push the models reliably to production environment with the changes in Data, Requirements , Model's performance. Mlflow helps in tracking the models effectively with its metadata



Multiple Dependencies:

Maintaining the dependencies in a large project with model



Working with Large Data Science Team:

To Track the Model metadata by extracting the work from other team members by creating the queries

Getting Started with mlflow



Docker for Machine Learning Projects



Why Learn Docker Containers ?

- Containers are an abstraction at the application layer which packages code and dependencies together
- Multiple containers can be run on same machine, and they share OS kernel



Long Setup

- When we wanted to Run a Package/Software code : Environment had to setup using – **requirements.txt**
- Its a manual activity , requires it to be a pre-requisite to run the application
- Need to repeat this activity for every environment/stages of the project, such as development, staging and production.
- Its difficult to manage the deployment scripts

Solution : Docker



- With Docker,

Put all the required files in a directory

Write down configuration

OS Version

Commands to be executed sequentially

*Can setup to communicate between 2 containers
in the same network*

Source Code :

The screenshot shows a GitHub repository named "Complete-MLOps-BootCamp". The repository is public and has one branch ("main") and no tags. The commit history for the "main" branch is displayed, showing 9 commits. A red box highlights the first commit, which corresponds to the "Docker-for-ML" folder. The commit details are as follows:

File	Commit Message	Time
Docker-for-ML	docker-for-ml	now
MLFlow-Manage-ML-Experiments	docker-for-ml	now
Packaging-ML-Model	mlflow-manage	yesterday
Python-for-mlops	mlflow-manage	yesterday
.DS_Store	docker-for-ml	now
.gitignore	mlflow-manage	yesterday
README.md	packaging-ml-model	last week

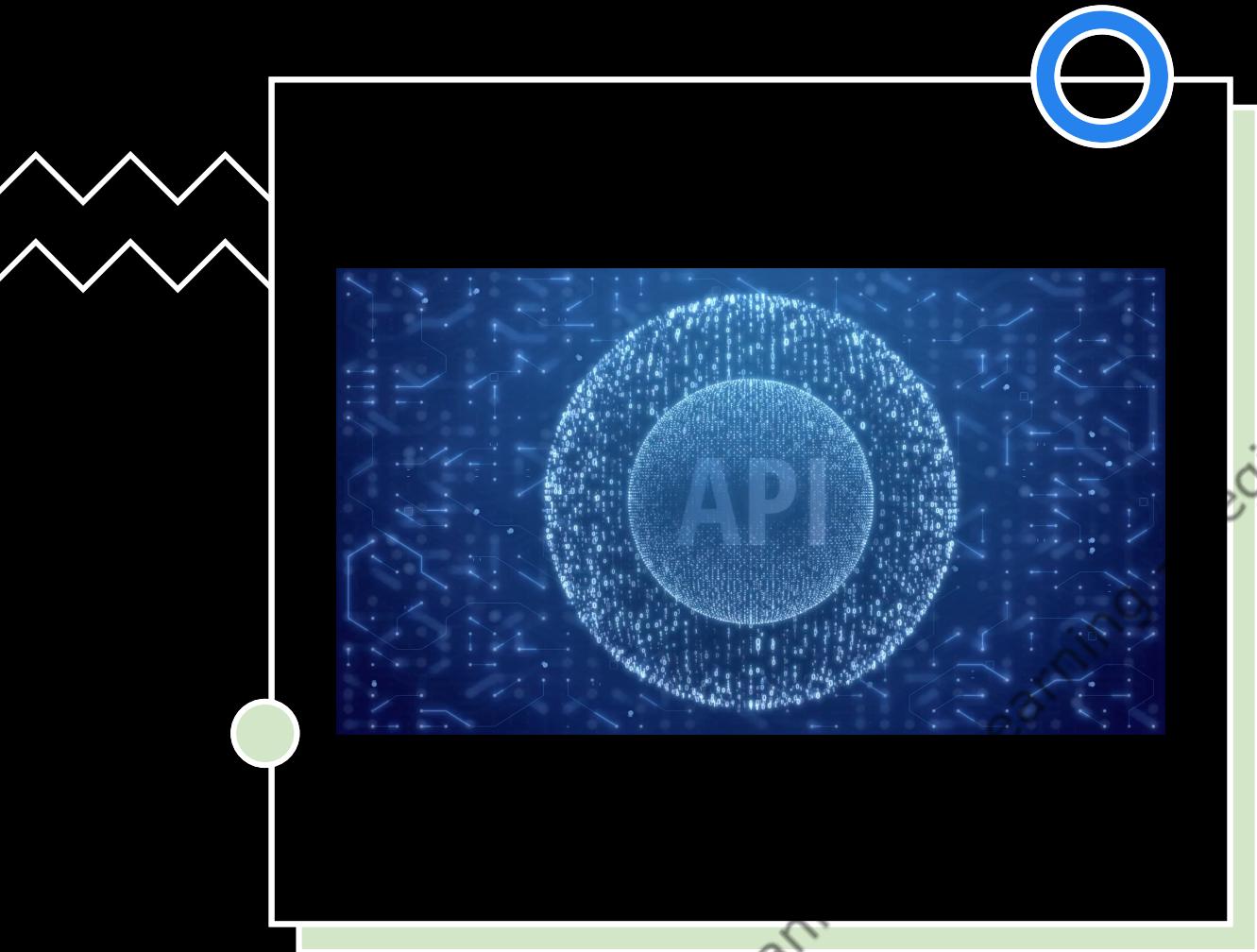
- Next Video : Understanding the Docker



Dockerize and Deploy the Machine Learning Model

Build Machine Learning Web Apps using API





What is API ?



What is API ?



An **application programming interface (API)** is a software gateway that allows different software components to communicate with each other.



APIs help expose the capabilities of an application to the outer world, allowing for programmatic access to their data.

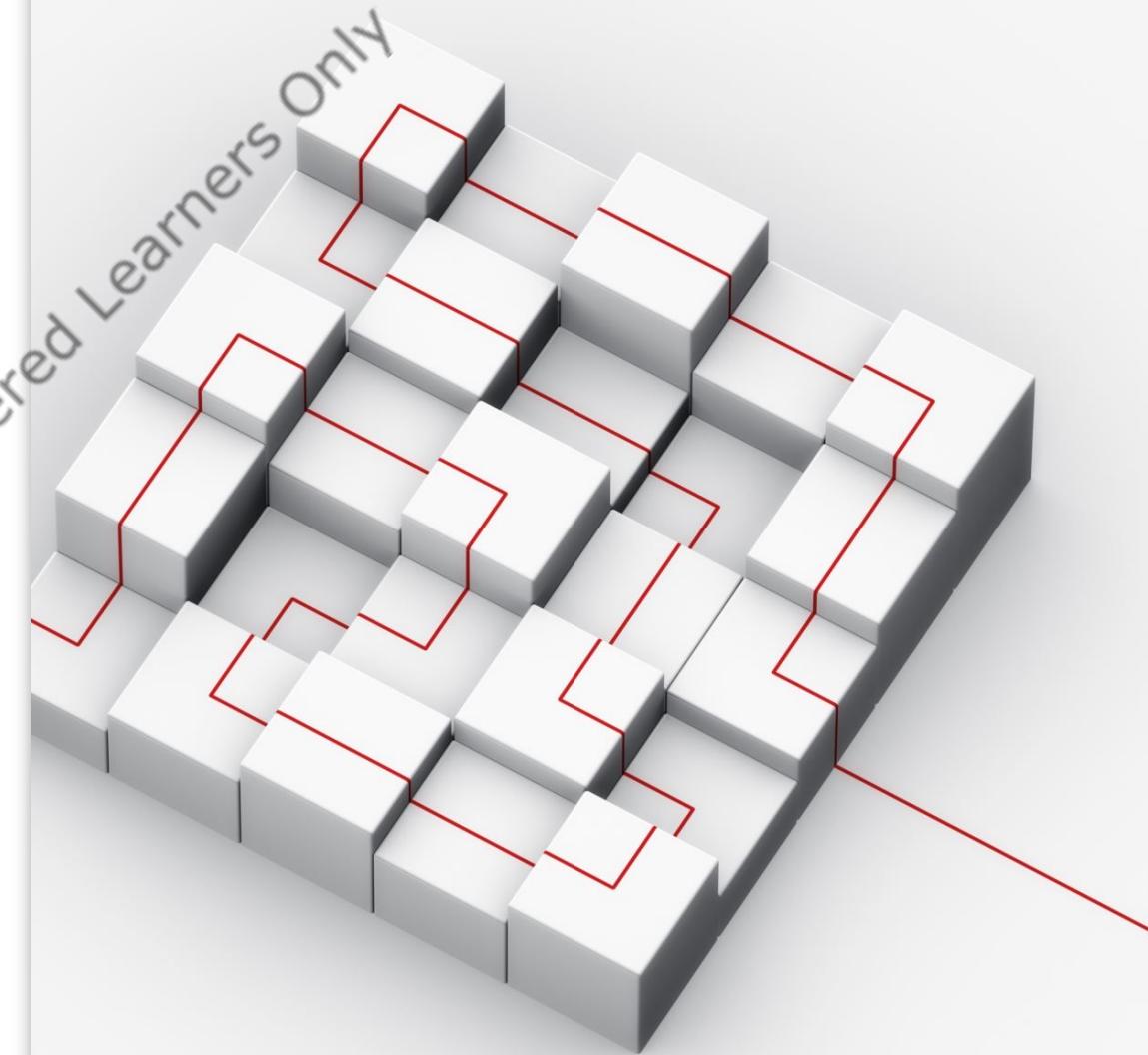


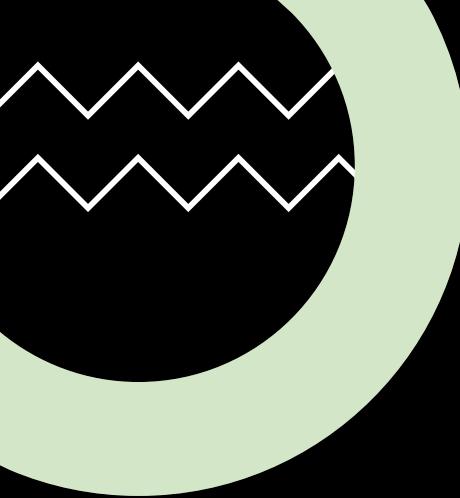
Example: The **Google Maps API** is a widely used API that allows developers to integrate Google Maps functionality into their own applications. It provides programmatic access to various mapping features, such as Mapping, directions, etc.



What is REST ?

- REST, which stands for **Representational State Transfer**, is an architectural style for designing networked applications.
- It's not a technology or a protocol but rather a set of constraints and principles that define how web services should be structured and how they should behave.
- REST is commonly used in the context of building web services and APIs (Application Programming Interfaces).
- It leverages the HTTP protocol making it simple and widely adopted approach for creation of distributed and scalable systems on the web.





So what is REST API ?

- Representational State Transfer (**REST**) is an architectural style that defines a set of constraints to be used for creating web services.
- **REST API** is a way of accessing web services in a simple and flexible way without having any processing.





Why REST technology is preferred ?

- REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because:
 - REST uses less bandwidth
 - Simple to use
 - Flexible & suitable for Internet usage

All Communication done via REST API uses HTTP request only

How does REST API Work?

Manifold AI Learning®
Registered Learners Only

How does REST API work ?



Resources:

Everything is considered as a Resource in the RESTful system.

Resources can be data objects, services, or even abstract concepts. Each resource is identified by a unique URI (Uniform Resource Identifier), similar to a web URL.



HTTP Methods:

REST APIs use standard HTTP methods to perform actions on resources

HTTP Methods



GET: Retrieve data from a resource. When a client sends a GET request to a resource's URI, the server responds with the resource's representation (usually in JSON or XML).



POST: Create a new resource. Clients use POST to send data to the server, typically to add new entries or perform actions that create new resources. The server responds with information about the created resource.



PUT: Update an existing resource. Clients use PUT to send updated data to the server to modify an existing resource identified by its URI.



DELETE: Remove a resource. A DELETE request instructs the server to delete the resource specified in the URI.

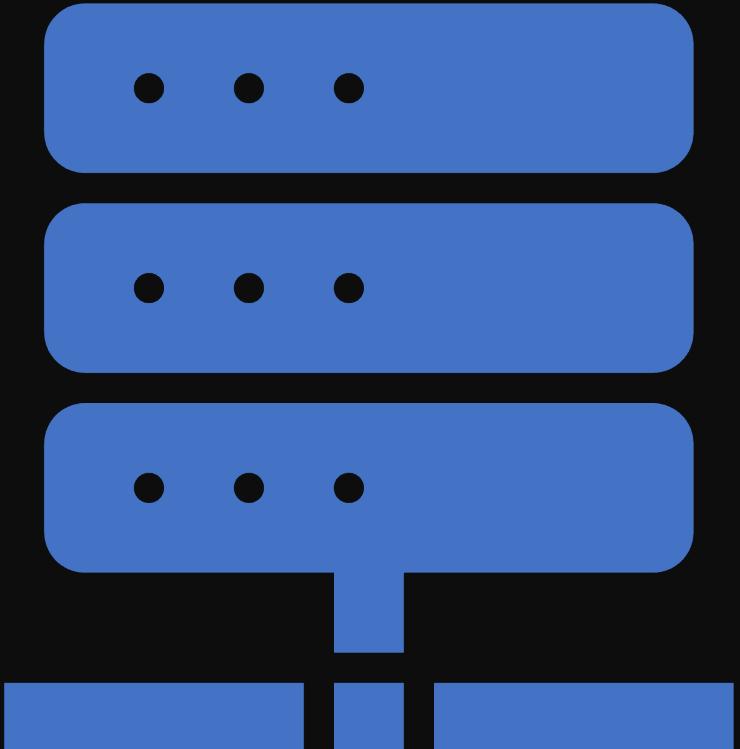
Stateless Communication

- RESTful communication is stateless, meaning each request from the client to the server should contain all the information necessary to understand and process the request.
- The server should not rely on any previous requests or store client-specific information between requests.
- This statelessness makes RESTful systems scalable and easy to maintain.

Manifold AI Learning - Registered Partners Only

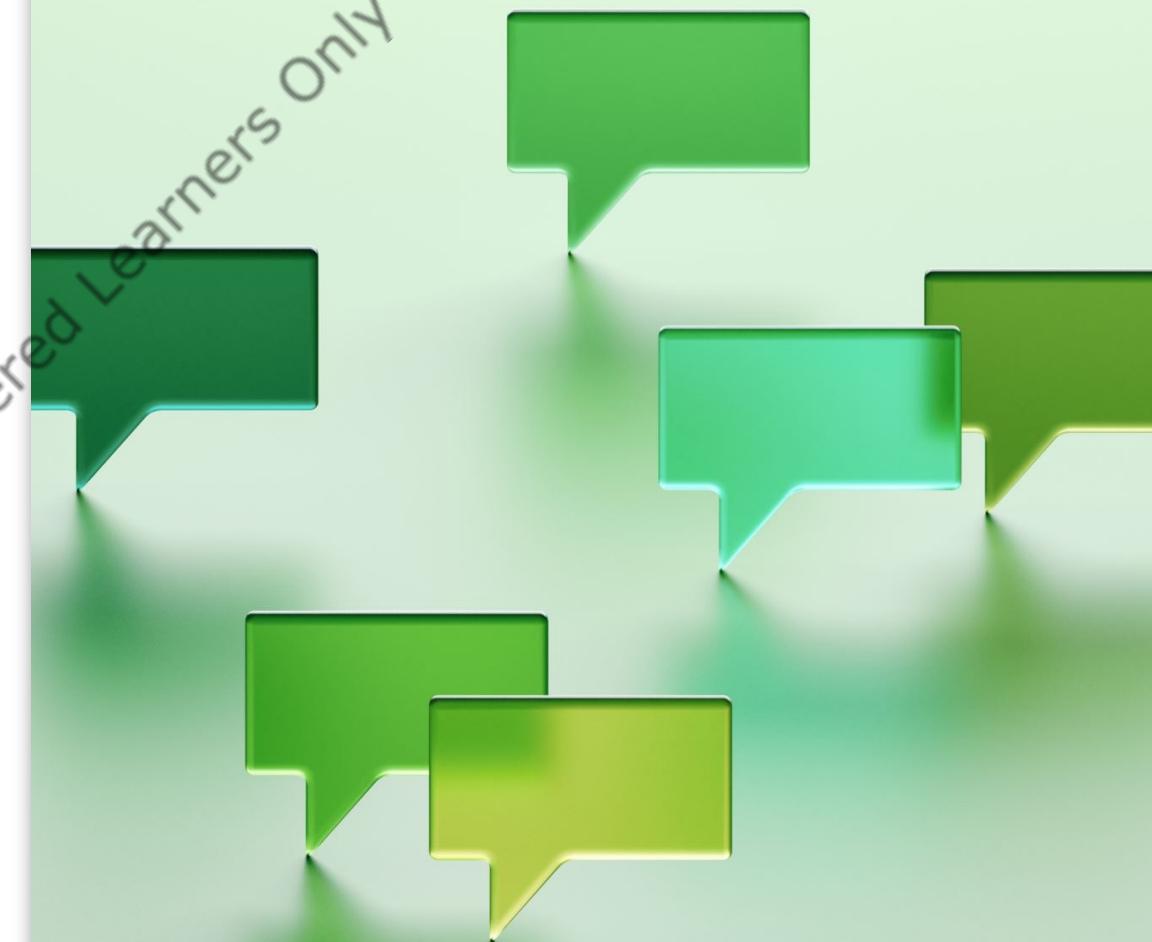
Responses

- When the server receives a request, it processes the request based on the HTTP method and the resource's URI. It then sends back an HTTP response to the client. The response typically includes:
 - A status code (e.g., 200 OK, 201 Created, 404 Not Found) indicating the result of the request.
 - Headers with metadata about the response.
 - The resource's representation in a format like JSON or XML.

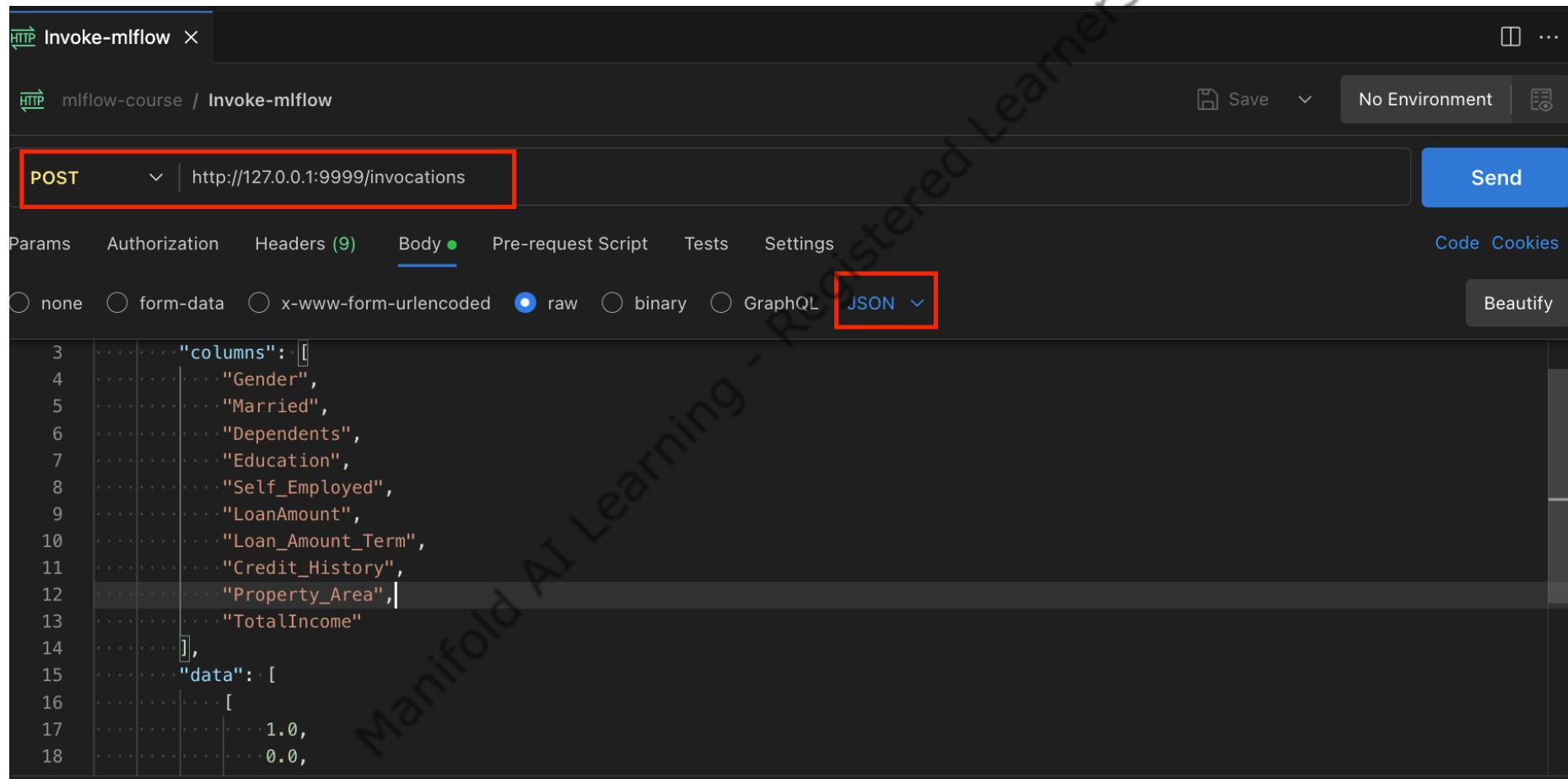


Overall Story!!

- Clients interact with a REST API by sending HTTP requests to the appropriate resource URIs, using the appropriate HTTP methods and including any required data or parameters.
- The server processes these requests and sends back HTTP responses, which can contain data, error messages, or status information.
- This exchange of requests and responses enables clients to interact with the server and access or manipulate resources in the system.



Remember the Screenshot from Previous Invocation



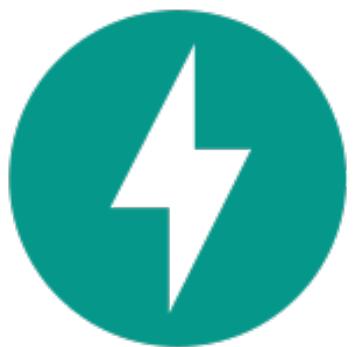
Next Video : FastAPI



FastAPI

Manifold AI Learning® Registered Learners

Fast API



FastAPI

Manifold AI Learning® Registered Learners Only



FastAPI

Manifold AI Learning - Registered Learners Only

What is FastAPI ?

- FastAPI is a modern, fast (high-performance), web framework for building APIs with Python.
- It's designed to make it easy to create RESTful APIs with minimal code while ensuring type safety, automatic documentation, and high performance.
- FastAPI has gained significant popularity in the Python web development community due to its simplicity and efficiency.

Features of FastAPI

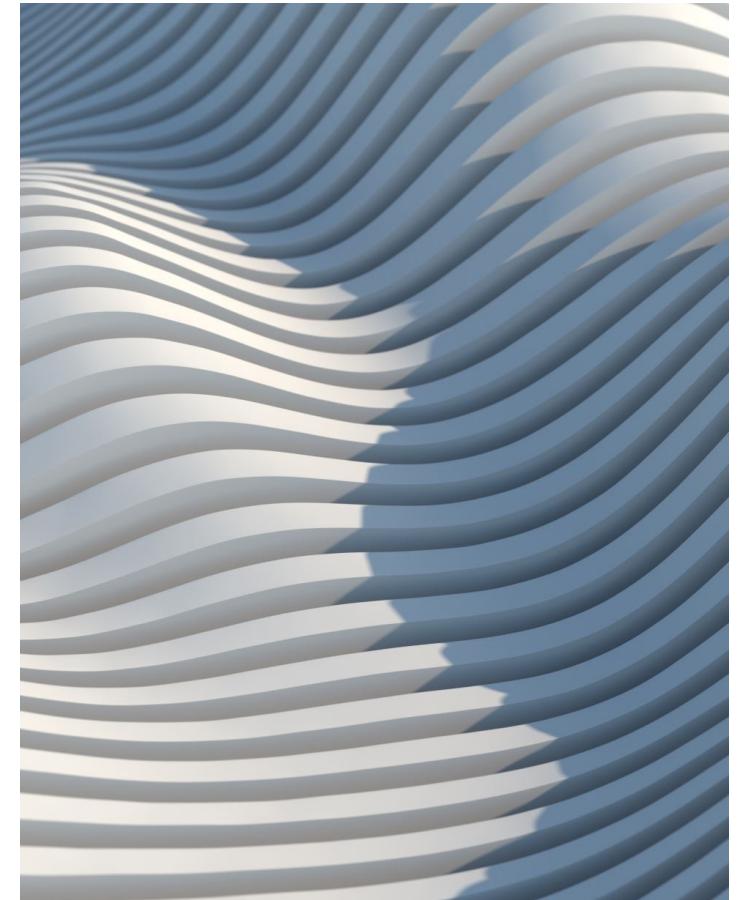
Python Type Hints: FastAPI leverages Python's type hinting system to provide automatic validation of request and response data. This means you can declare the expected data types, and FastAPI will handle input validation and serialization automatically.

Asynchronous Support: FastAPI fully supports asynchronous programming using Python's `async/await` syntax. This makes it suitable for handling concurrent requests efficiently, which is especially important for I/O-bound operations like database queries or external API calls.

Automatic API Documentation: FastAPI generates interactive API documentation automatically based on your code and type hints. It uses the OpenAPI standard, and you can access the documentation by visiting a specific endpoint in your application.

Interactive Development: With FastAPI's automatic documentation and request validation, developers can interactively test and explore their API endpoints using a web-based interface, making the development and debugging process smoother.

Dependency Injection: FastAPI supports dependency injection, allowing you to organize and reuse code components (e.g., database connections, authentication) easily within your API routes.



Features of FastAPI (Contd.)

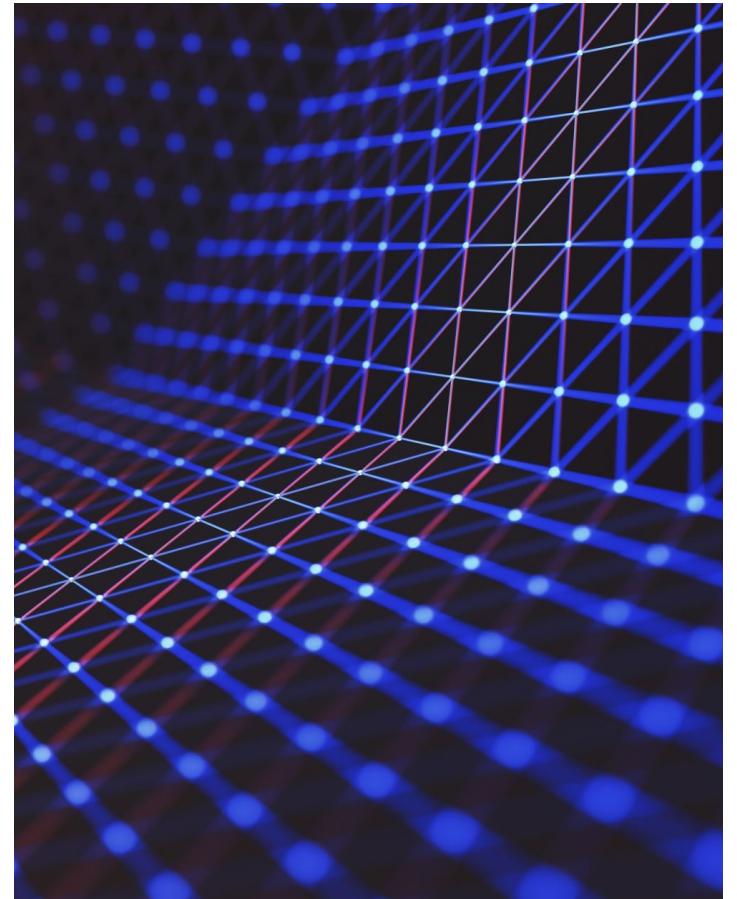
WebSockets: While FastAPI is primarily designed for building RESTful APIs, it also provides support for handling WebSocket connections, enabling real-time communication between clients and the server.

Integration with Popular Data Stores: FastAPI can be integrated with various databases and ORMs (Object-Relational Mapping) like SQLAlchemy and Tortoise-ORM for data persistence.

Security Features: FastAPI includes built-in security features to help protect your APIs, including authentication, authorization, and input validation.

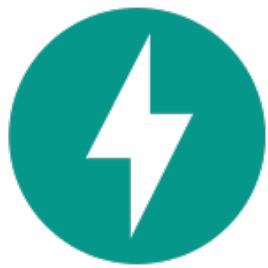
Middleware Support: You can use middleware functions to add custom logic to your API application, such as logging, CORS (Cross-Origin Resource Sharing) handling, and more.

Extensible: FastAPI is highly extensible, allowing you to add custom components, authentication methods, and plugins to tailor the framework to your project's needs.



Next Video:

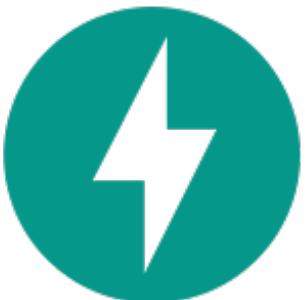
FastAPI Crash Course



FastAPI

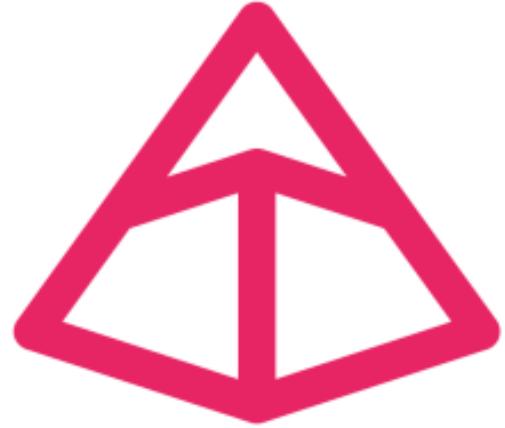
Manifold AI Learning - Registered Learners Only

Crash Course on FastAPI



FastAPI

Manifold AI Learning® Registered Learners Only



Pydantic

Bringing schema and sanity to your data

Pydantic Library

Manifold ML Learning - Registered Learners Only

What is Pydantic ?

It's a parsing library.

Guarantees the types and constraints of the output model

Can also use it for custom validation as well

Why Use Pydantic ?



It enhances code readability by clearly defining the expected or required data structure and types, which proves especially valuable when collaborating on code or returning to it after an extended period.



Pydantic validates data passed to functions, preventing unwanted consequences resulting from incorrect data types. In situations where you cannot guarantee the nature of the data supplied to your program, this protective measure becomes essential.

In Relationship to ML Model Deployment

Data you receive from
Other clients might not
always be in the
format that my model
expects

Pydantic ensures that
– data will be in right
format before we
perform prediction

Basic Hands on – Learn Pydantic

Manifold AI Learning - Registered Learners Only

Deploy ML Model as Web Apps with Streamlit

Manifold AI Learning - Registered Learners Only



Streamlit



Streamlit

- Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps.
- It is a Python-based library specifically designed for machine learning engineers/ Data Scientists.



Why Use Streamlit ?

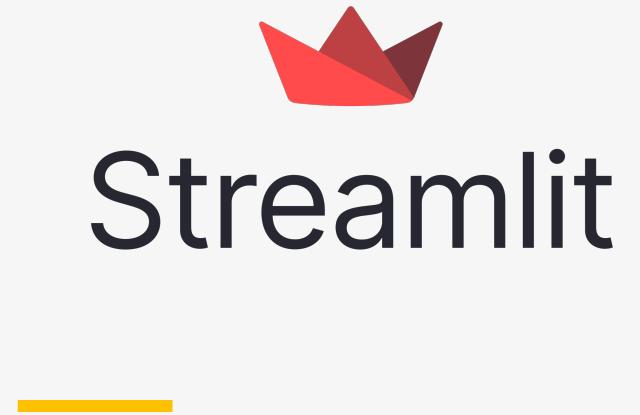
1. Streamlit is an open-source Python library for rapidly creating attractive user interfaces.
2. Building an effective and intuitive user interface is crucial for application success.
3. Streamlit allows users with no web development experience to create web applications effortlessly.
4. It is ideal for data scientists looking to deploy models quickly and easily.
5. Many data-heavy apps struggle with UI development, but Streamlit simplifies the process.

Benefits of Streamlit

- No need for front-end (HTML, JS, CSS) knowledge.
- Quick development, enabling beautiful machine learning and data science apps in hours or minutes.
- Compatibility with popular Python libraries like pandas, matplotlib, seaborn, plotly, Keras, PyTorch, and SymPy (latex).
- Requires less code to create impressive web apps.
- Offers data caching to simplify and speed up computation pipelines.

Next Video : Hands On
Tutorial on Streamlit

Manifold AI Learning - Registered Learners Only



Build Machine
Learning Web
apps using
Flask

Manifold AI Learning - Registered Learners Only



Flask

web development,
one drop at a time

What is Flask ?

- Flask is a web framework.
- Flask provides you with tools, libraries and technologies that allow you to build a web application.
- This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

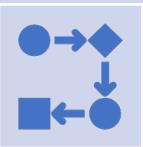
Manifold AI Learning ® Registered Learners Only



Benefits of Flask



Lightweight and Minimalistic: Flask follows the "micro" framework philosophy. It provides the essentials for building web applications without imposing a lot of structure or dependencies.

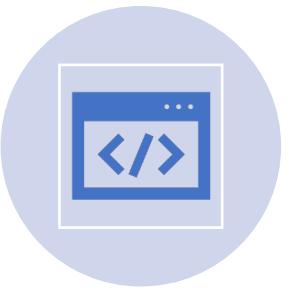


Easy to Learn and Use: Its straightforward API and well-documented documentation make it accessible to beginners.



Extensible: Flask is highly extensible, allowing developers to choose and add the specific components and libraries they need for their project. This means you can add extensions for things like authentication, databases, and more, tailoring your application to your requirements.

Other Benefits :



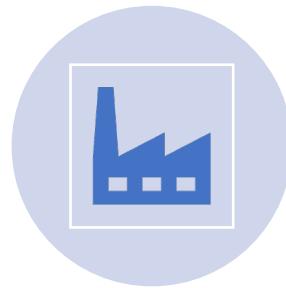
Build-in Development Server makes it easier to test the application



Supports URL Routing



Well suited for RESTful APIs and web services



Its Production-Ready framework

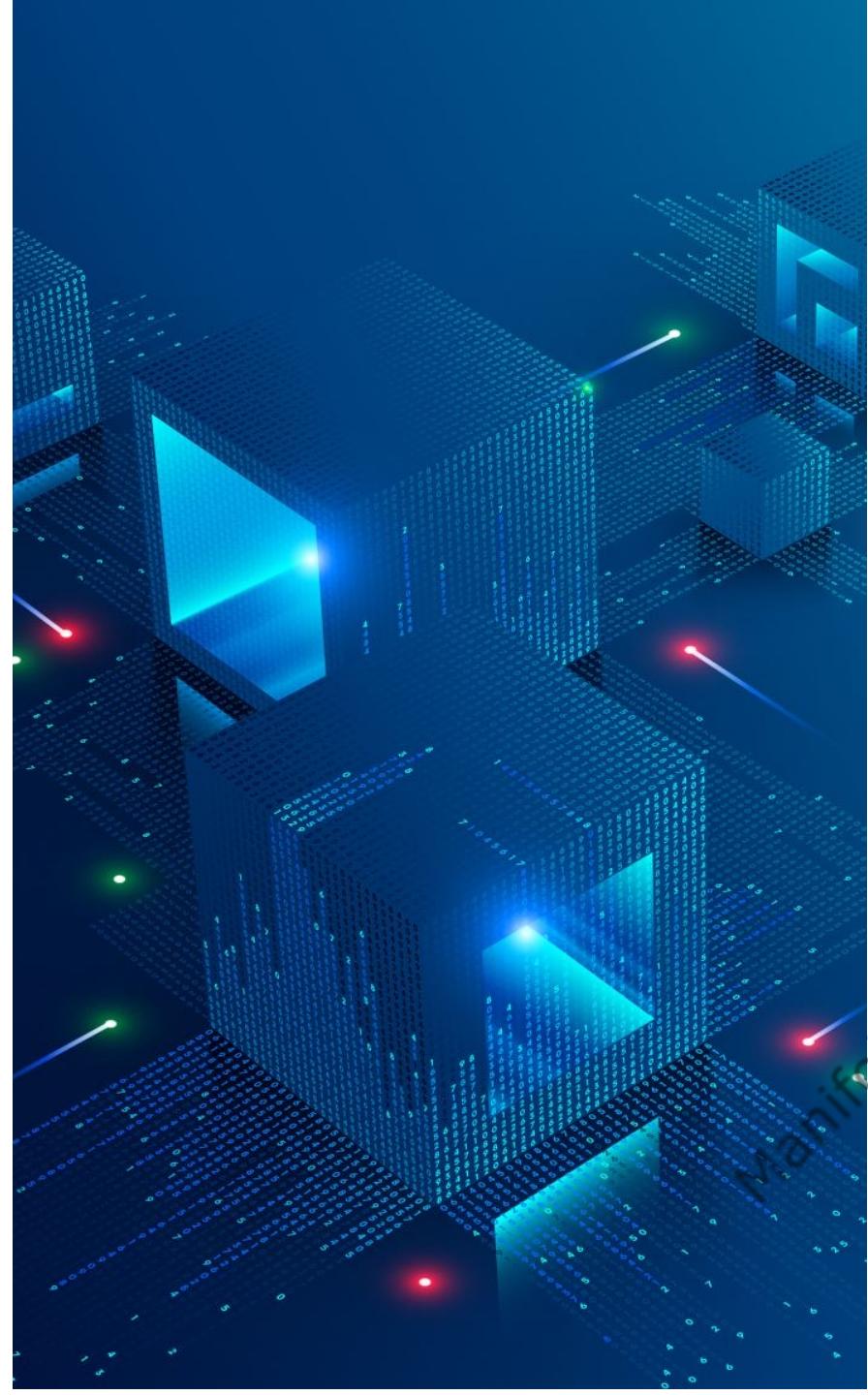
Manifold AI Learning - Registered Learners Only

Next Video:
Getting
Started with
flask



Flask

web development,
one drop at a time



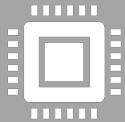
What are Frameworks in Python ?

- A framework is a pre-established set of libraries, modules, and guidelines that provide a structured way to develop applications or software.
- Frameworks are designed to simplify the process of building complex software by offering reusable code, predefined structures, and conventions that help developers create applications more efficiently.

Library Vs Framework



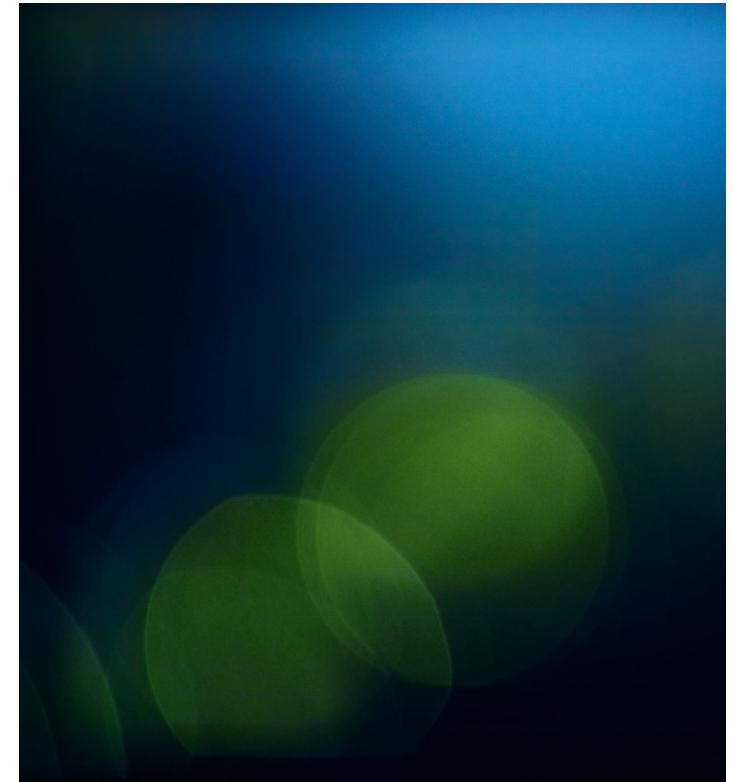
A library comprises packages that carry out specialized tasks, whereas a framework encompasses the fundamental structure and design of an application.



The primary distinction between the two lies in their intricacy. Libraries encompass numerous methods that developers can readily invoke when coding. Conversely, frameworks necessitate the creation of additional functionalities.



Frameworks offer a significant advantage: adaptability. They furnish essential tools and capabilities that developers can expand upon and further develop.



Linux Operating System

DevOps & Data Scientists

Manifold AI Learning - Registered Learners Only

Agenda of this Section:

This section is intended to help the practitioners to become aware of the Linux Operating System which is a basic for working on Command Line Interface on Servers



Infrastructure used in Hands-On Videos

EC2 Instance with Amazon Linux 2 AMI

Manifold AI Learning - Registered Learners Only

If you're new to AWS - Refer to Getting Started with AWS Section

Manifold AI Learning - Registered Learners Only

Next Video:

Basics of Linux

Manifold AI Learning - Registered Learners Only

Basics of Linux

DevOps & Data Scientists

Manifold AI Learning - Registered Learners Only

Agenda

What is Linux

Components of Linux

Why Linux ?

Linux Distro

Bash in Linux

Package Manager in Linux

Manifold AI Learning - Registered Learners Only

What is Linux ?



Linux is an operating system similar to Windows, iOS, and Mac OS.

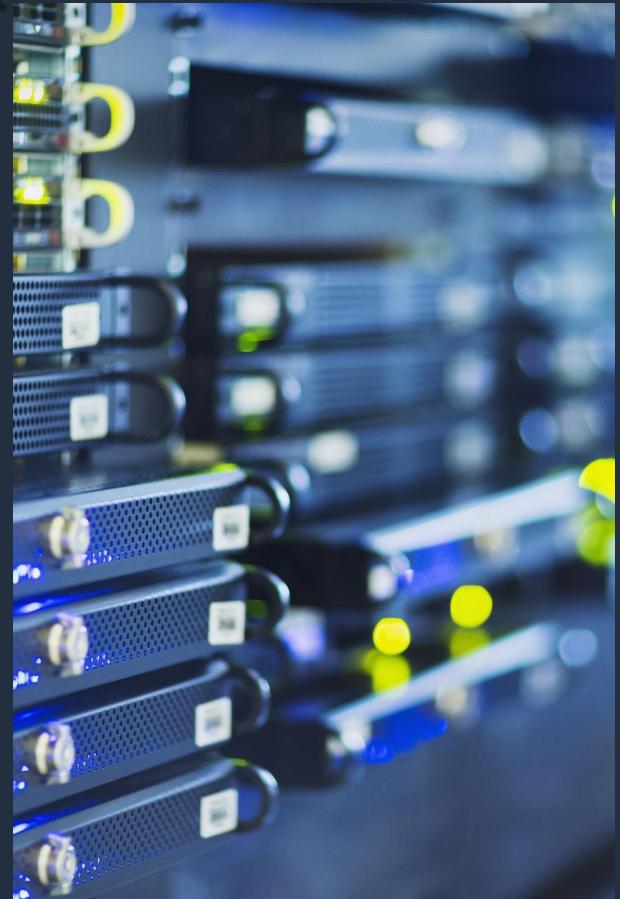
It is worth noting that Android, a highly popular platform worldwide, operates on the Linux operating system.

An operating system is a software that oversees the hardware resources of your computer. In essence, it facilitates the communication between your software and hardware components.

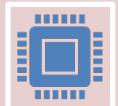
Without the operating system (OS), the software would be unable to operate.

Components of Linux Operating System:

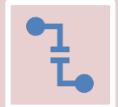
- **Bootloader** - This software manages the computer's boot process
- **Kernel** - The kernel, known as "Linux," is the core component of the system
- **Init system** - This subsystem initiates the user space and controls daemons
- **Daemons** - These are background services like printing, sound, and scheduling that either start during boot or after logging into the desktop
- **Graphical server** - Also referred to as the X server or X, this subsystem handles graphics display on the monitor
- **Desktop environment** - This is the part of the operating system that users directly interact with
- **Applications** - While desktop environments have limited applications, Linux, like Windows and macOS, provides a vast selection of high-quality software titles



Why use Linux?



Open-source nature: Linux is an open-source operating system, which means its source code is freely available to the public.



Stability and reliability: Linux is known for its stability and reliability.



Security: Linux has a strong focus on security. Due to its open-source nature, security vulnerabilities can be quickly identified and fixed by the community.

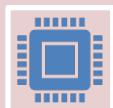
Why use Linux? (Contd.)



Flexibility and customization: Users have the freedom to choose from various desktop environments, customize their system's appearance, and configure it to their preferences.



Software and package management: Linux distributions provide centralized software repositories and package management systems.



Community and support: Linux has a large and vibrant community of users and developers worldwide.

Why use Linux? (Contd)

Cost-effective: Linux is often considered a cost-effective option compared to proprietary operating systems like Windows or macOS.

What is a “distribution?”

A Linux distribution, also known as a distro, is a complete operating system package that consists of the Linux kernel, various software applications, libraries, and configuration files. Different distributions may have different default software, desktop environments, package management systems, and overall design philosophies. They are created and maintained by different organizations or communities, each with its own goals and target audience.

Popular Linux Distributions

Ubuntu: One of the most widely used and beginner-friendly distributions. It focuses on ease of use, stability, and a large user community.

Debian: Known for its stability, reliability, and adherence to open-source principles. Debian serves as the base for several other distributions, including Ubuntu.

Fedora: Developed by the Fedora Project and sponsored by Red Hat. It emphasizes the use of cutting-edge technologies and serves as a testing ground for new features that may eventually make their way into Red Hat Enterprise Linux (RHEL).

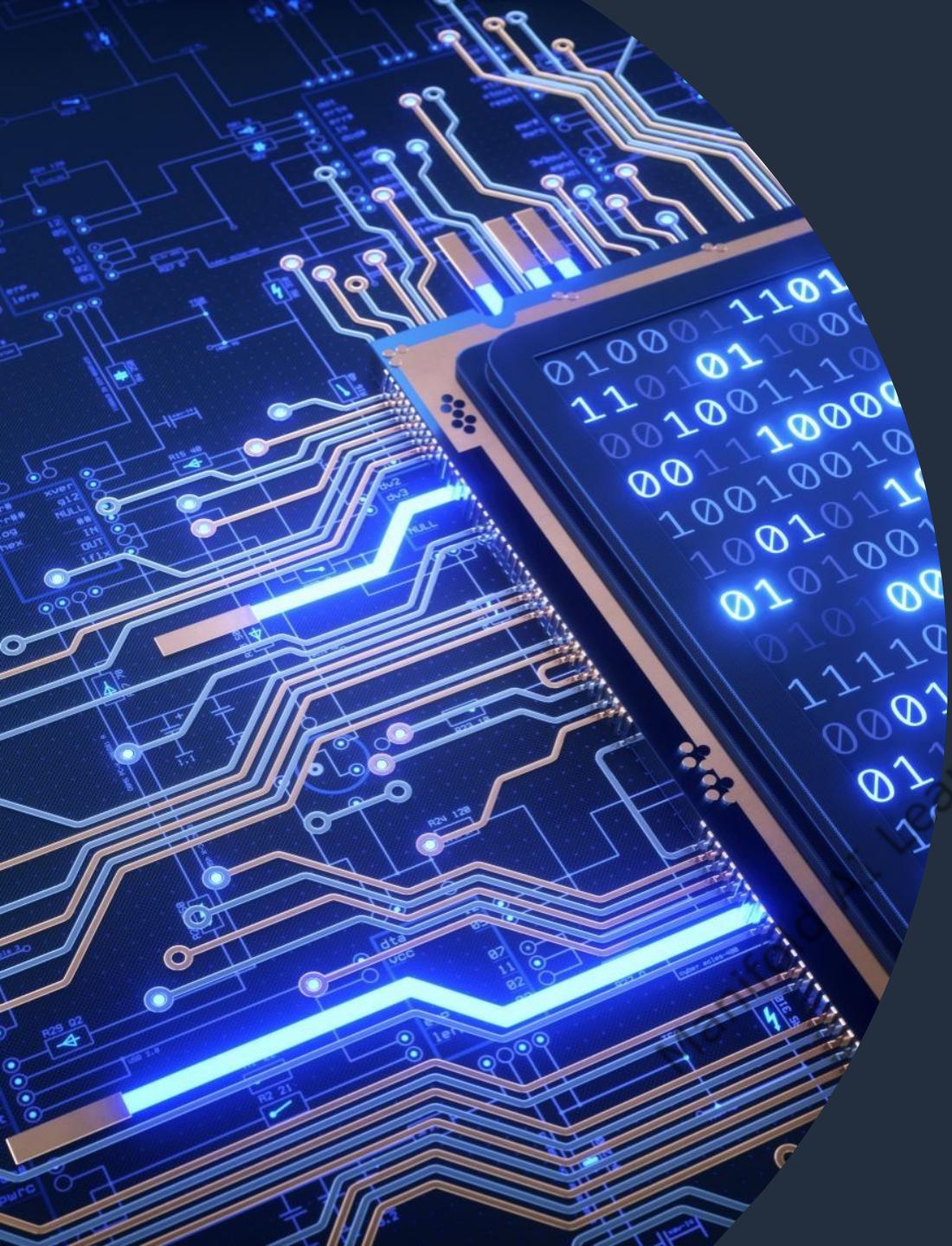
CentOS: Based on the source code of Red Hat Enterprise Linux, CentOS aims to provide a free, community-supported alternative to RHEL. It is popular for server deployments.

Arch Linux: Designed for advanced users who prefer a DIY (do-it-yourself) approach. Arch Linux focuses on simplicity, minimalism, and user-centric customization.

openSUSE: A community-driven distribution known for its stability, user-friendly configuration tools, and professional-grade features.

Linux Mint: Built on top of Ubuntu, Linux Mint provides a polished and user-friendly desktop environment, making it an attractive choice for beginners transitioning from Windows.

Manjaro: Based on Arch Linux, Manjaro is known for its user-friendly approach and pre-installed software packages. It offers a balance between cutting-edge software and stability.

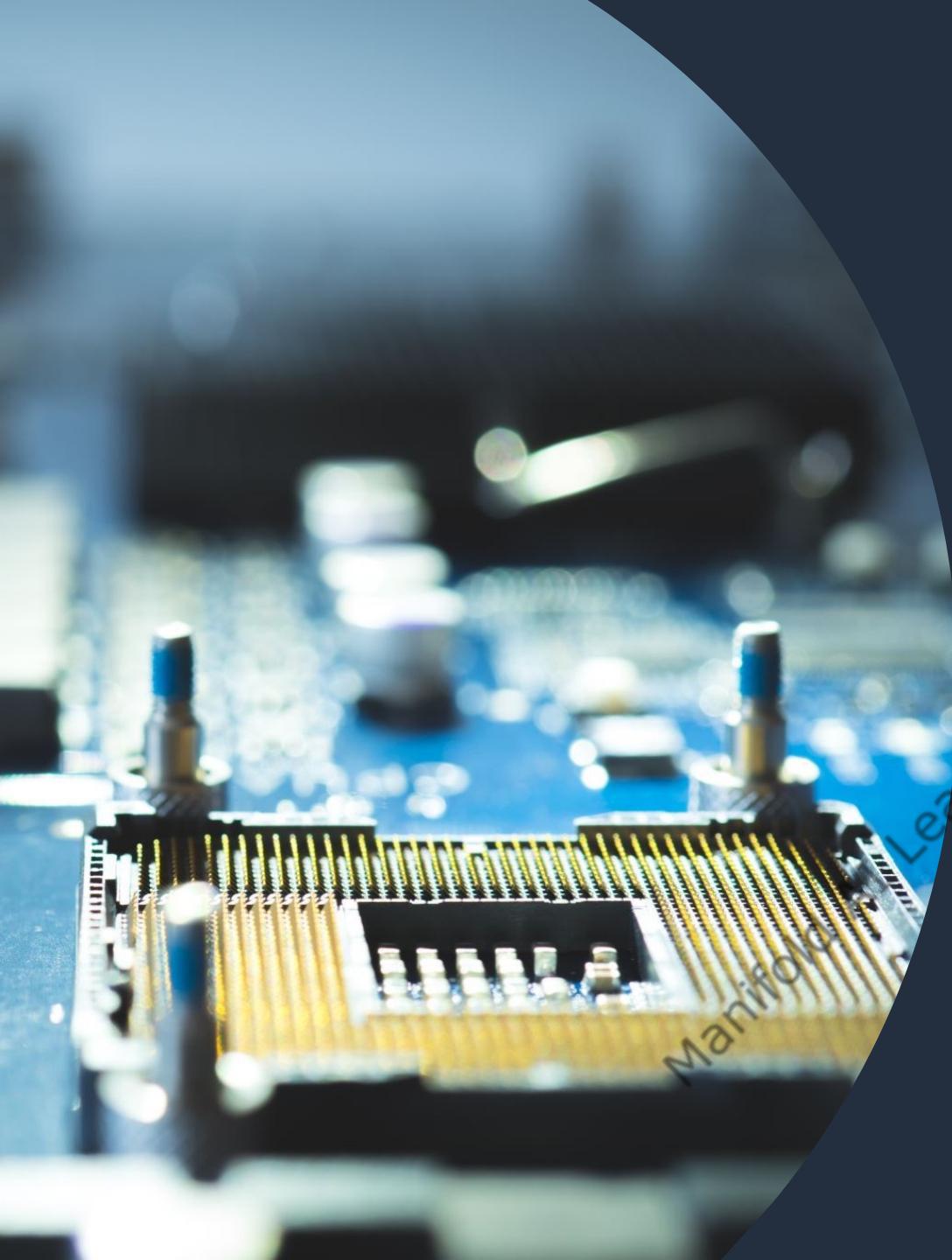


Bash in Linux

Bash, short for "Bourne Again Shell," is a command-line interpreter and scripting language that is widely used in Linux and Unix-based operating systems

Bash provides a textual interface for users to interact with the operating system by executing commands

Bash offers powerful features and capabilities, including command-line completion, command history, variables, conditionals, loops, functions, and input/output redirection



Package Managers in Linux

Package managers in Linux are software tools that handle the installation, management, and removal of software packages on a Linux distribution

Package managers simplify the process of installing and updating software, making it more efficient and convenient for users

Package managers simplify the software management process in Linux, making it convenient for users to keep their systems up to date and install new software with ease

Common Package Managers in Linux

APT

YUM

DNF

Pacman

Zypper

Manifold AI Learning - Registered Learners Only



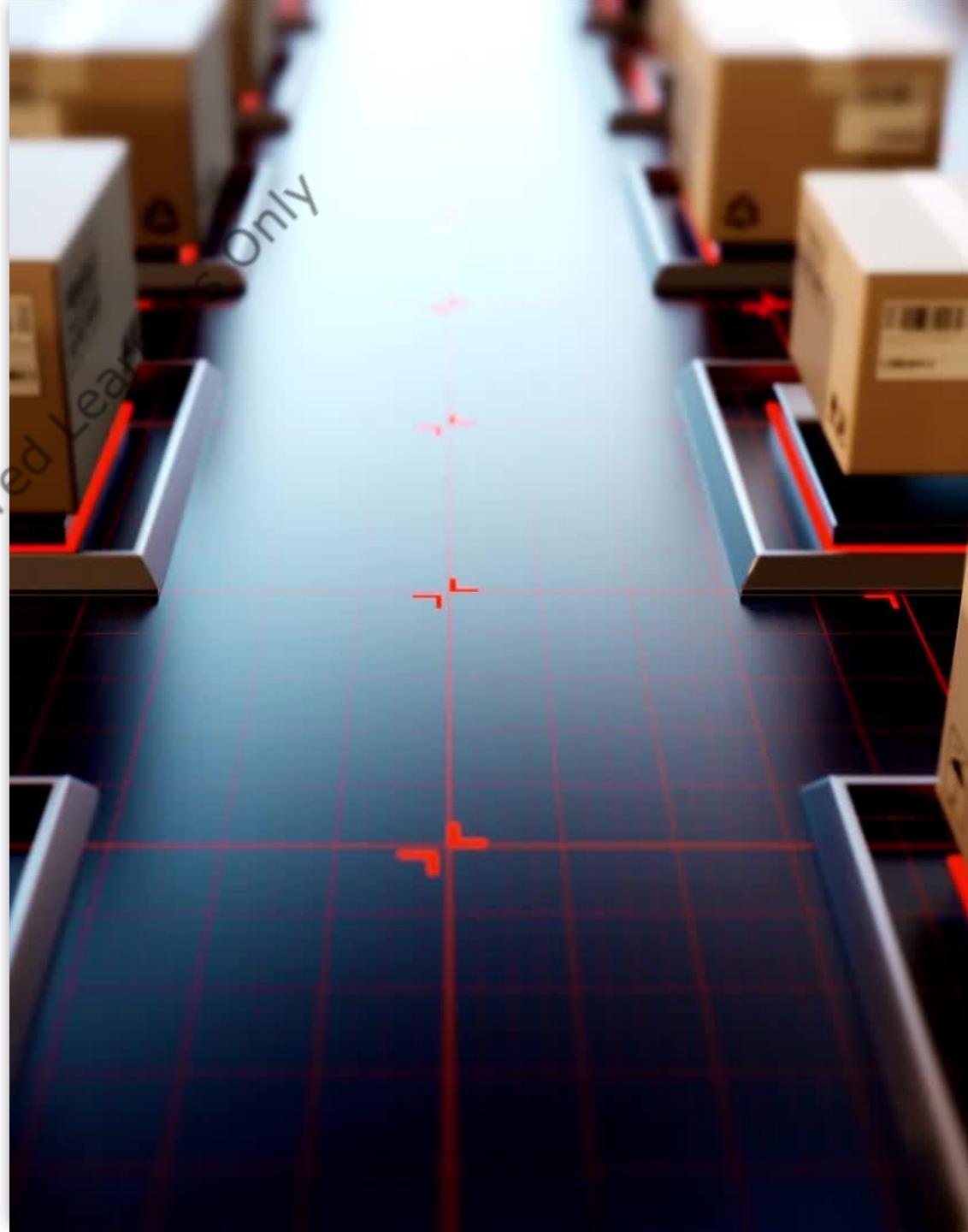
Continuous Integration
& Continuous
Deployment with
Jenkins

Automation Tool

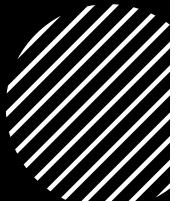
Manifold AI Learning ®
Registered Learners Only

Our Journey in Implementing the MLOps :

- We have learnt about Packaging of the ML Code
- Developers write the Code and Commit to Git based repository
- To ensure the environment is replicable – We have learnt about Docker Containers and ensure it runs smoothly in all the systems
- We have also seen as how to perform test of ML Code using **pydantic**
- ***Now, to build and test the Software/ML projects continuously and integrate the changes to Project – We use the CICD tool Jenkins.***



Best Practices for MLOps Success



Split the Entire Code into Modules (Segments)



Integrate the Modules and Package it



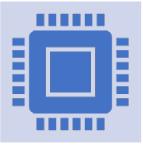
Adopt a Continuous Integration methodology to coordinate with the team

Case Study

Let's consider the problem faced by the Manifold AI Learning, and how they adopted Continuous Integration



The Problem



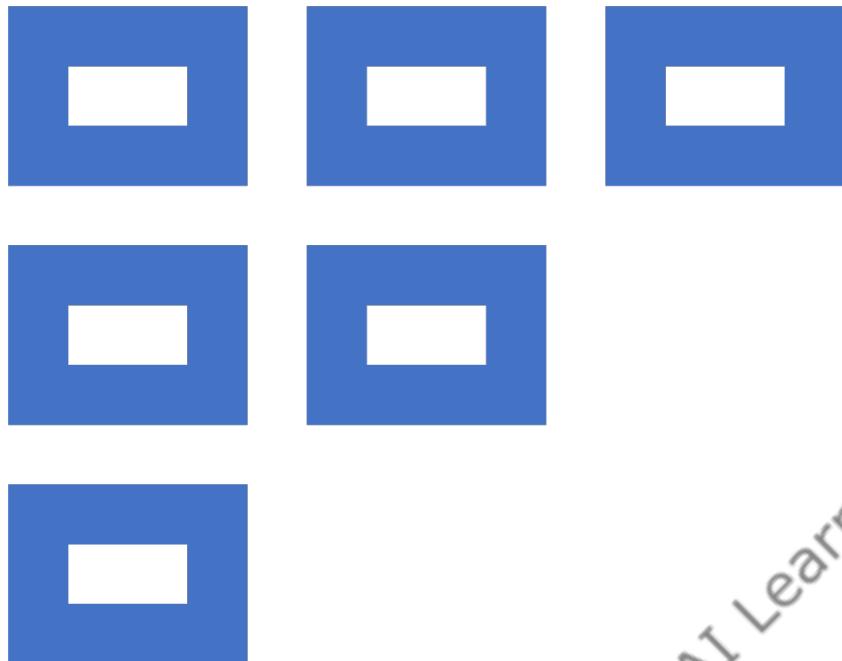
In a software product development project at Manifold AI Learning, there was a process called Nightly builds



Every night an automated system pulls the code added to the shared repository throughout the day, and performs the build on that code



This idea is quite similar to Continuous Integration, but since the code was built at night, and was quite large, locating and fixing of bugs was very difficult



Solution

- With Continuous Integration:
 - Every Commit made to Source code in the repository was built
 - If the Built result was failed, then the developers will receive the notification, and they need to work on that commit only.
 - This significantly reduced the time required to release new software

Manifold AI Learning - Registered Learners Only

What is Continuous Integration ?



Manifold AI Learning - Registered Learners Only

- It's a process of automating the building and testing of code, each time a team member commits the changes to version control system.



Benefits

- 
- 

Improves the Quality by running multiple unit tests and analyze various static code
- 

Increases productivity automating build of code saves a lot of time, thereby increasing productivity
- 

Eliminate the Risk of potential human errors by automating the test
- Manifold AI Learning ®

Popular CI Tools in Market

- Gitlab CI
- Jenkins
- CodeShip
- Bamboo
- Travis CI
- Teamcity

Manifold AI Learning - Register & Learn

Manifold AI Learning



Introduction to Jenkins

A continuous integration server which manages and control processes such as plan, code, build, test, deploy, operate and monitor in DevOps Environment.



Jenkins

Manifold AI Learning - Registered Learners Only

Why Jenkins is popular ?

- Open Source
- Extensibility
- Automation
- Large Community
- Compatibility
- Ease of Use
- Scalability
- Security
- Integration
- Active Development

Manifold AI Learning - Registered Learners Only



Features of Jenkins

**Easy
Installation and
Configuration**

**Pipeline as
Code**

**Monitoring and
Reporting**

Notification

**Artifact
Management**

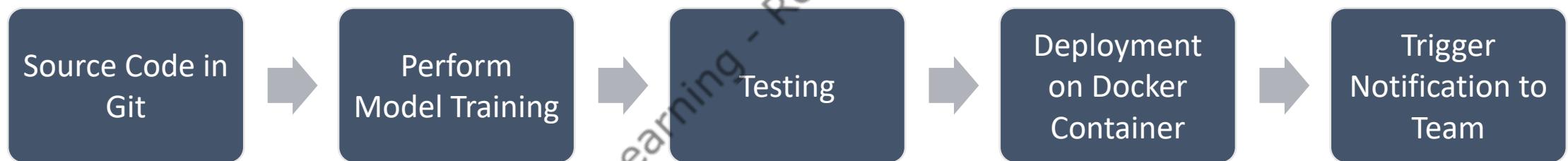
**Environment
Provisioning**

Manifold AI Learning - Registered Learners Only

Next Video : How do we use Jenkins on MLOps Project

Manifold AI Learning® Registered Learners Only

Jenkins Project:



Pre-Requisites for Jenkins Project

- Prepare and Package ML Model
- Create Fast API App
- Dockerization of ML App
- Test Locally

Manifold AI Learning - Registered Learners Only



Jenkins

Next Video : Installation of Jenkins on Ubuntu

Manifold AI Learning® Registered Learners Only

Installations to be Done

- Launch Ubuntu Server with Port 8080 & 8005 availability
- Install Jenkins
- Install Docker Service
- Validation of Installation

Manifold AI Learning - Registered Learners Only

Setting Up Ubuntu Server



Provide Permissions for Jenkins to Start and create containers



Provider the user permission for Docker



Validate the Permission

Setup Github Repository for Tracking Changes

- Create Repo
- Push the data to Github Repository
- Generate Personal Access Token & Store in Jenkins
- Setting Up Webhooks

Test Github Webhooks in Jenkins



Jenkins

Manifold AI Learning ®
Registered Learners Only



Create FreeStyle Project in Jenkins

Manifold AI Learning - Registered Learners Only

Freestyle Project

- A freestyle project in Jenkins is a type of project that allows you to build, test, and deploy software using a variety of different options and configurations.
- This can be done with GUI
- Suitable for Less Complex scenarios (Ansible is used for maintenance of config files)
- Good for Beginners

Pipeline Jobs

- Use code (Groovy language - Java like language) for giving instructions
- Since, everything in one script you can keep that in the source control and have ability to revert back to an earlier version at any time or keep track of changes made to the script
- Entire pipeline consists of steps (ex: build, test, deploy etc..)
- Created using a Jenkinsfile

Test Github Webhooks in Jenkins



Jenkins

Manifold AI Learning ®
Registered Learners Only



Setting Up Plugins for Jenkins

- Docker Plugin Installation
- Setting Up Credentials for Dockerhub
- Validation with Simple Project

Manifold AI Learning - Registered Learners Only

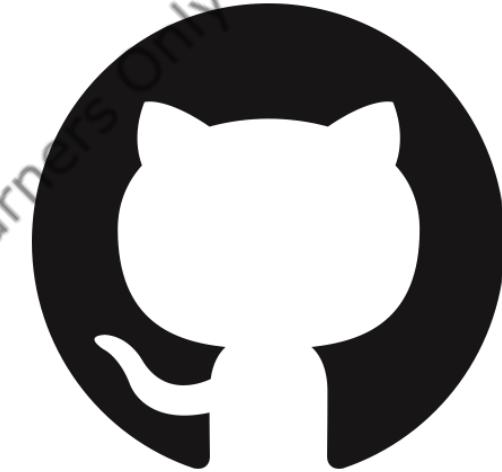
Setting Up Email Notification Trigger

- Setting Up 2FA on Gmail
- Generate Application Password
- Email Notification Plugin Jenkins
- Test Email Trigger

Manifold AI Learning - Registered Learners Only



Jenkins



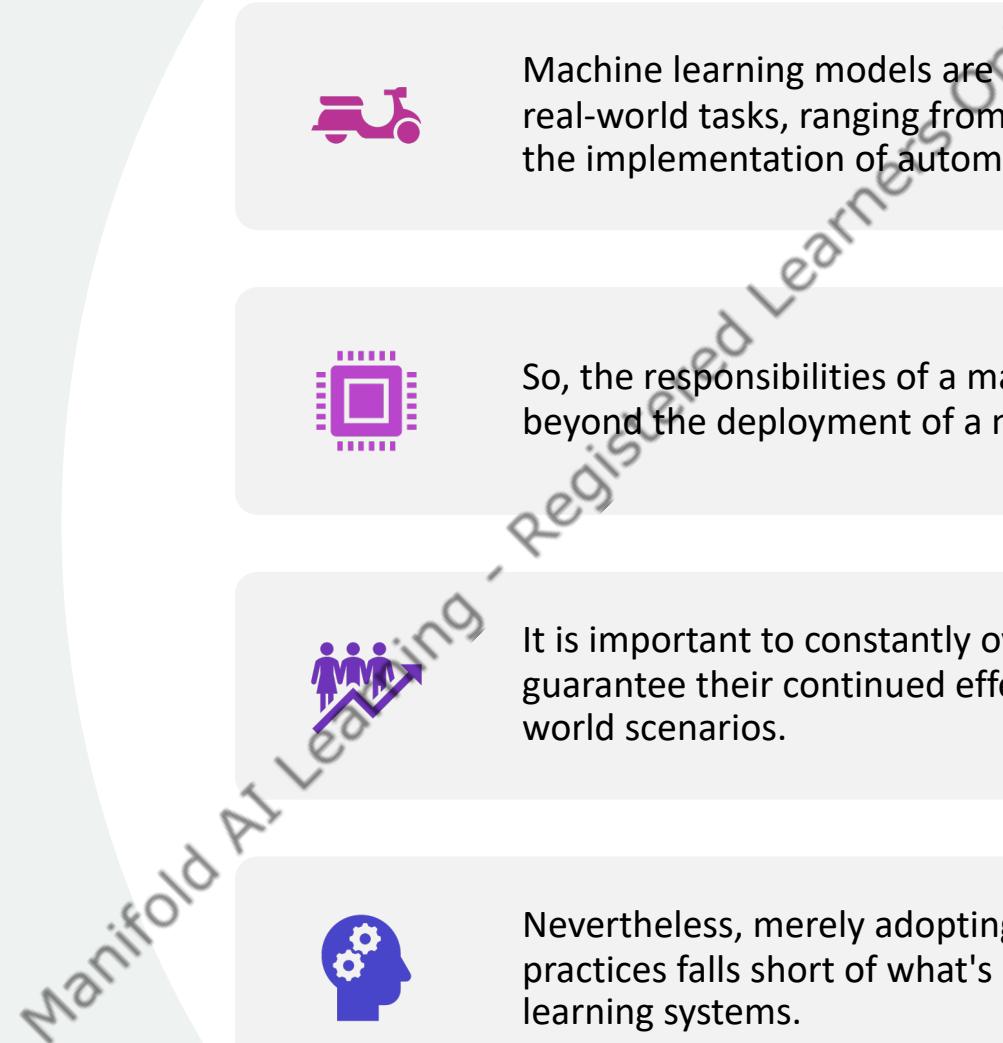
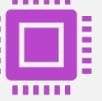
Create CI CT CD Pipeline in Jenkins

Manifold AI Learning - Registered Learners Only

Monitoring Machine Learning Models in Production

Manifold AI Learning - Registered Lead

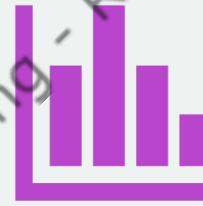
Importance of Monitoring

- 
-  Machine learning models are now more frequently employed for critical real-world tasks, ranging from the detection of fraudulent activities to the implementation of automated braking systems in vehicles.
 -  So, the responsibilities of a machine learning professionals extend well beyond the deployment of a model into a production environment.
 -  It is important to constantly oversee the performance of these models to guarantee their continued effectiveness when confronted with real-world scenarios.
 -  Nevertheless, merely adopting conventional software monitoring practices falls short of what's necessary in the context of machine learning systems.

Questions to think !!



How do we monitor
effectively ?



What metrics to
choose ?



What tools are
available ?

Manifold AI Learning ® Registered Learners Only

Monitoring of ML Models

61.6 %: 99.19

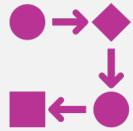
Monitoring of Machine Learning Models

It's important to note that monitoring isn't a one-time task that can be completed and then forgotten.

Monitoring entails the ongoing process of observing a deployed model's behavior to assess its performance.

Post-deployment monitoring is crucial because machine learning models can deteriorate and malfunction when in active use.

When to Update the Model in Production?



To determine the appropriate moment for updating a model in a production environment, it's essential to maintain a real-time perspective that enables stakeholders to continuously evaluate the model's performance within the live setting.

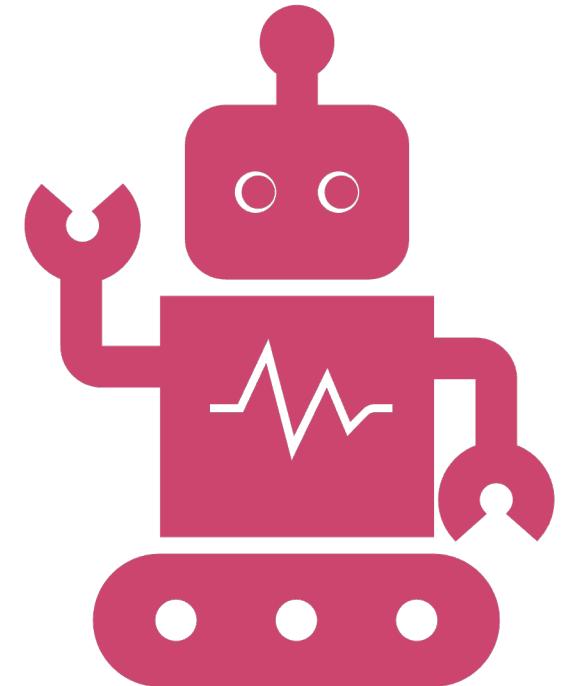


This continuous assessment ensures that the model is operating as anticipated.

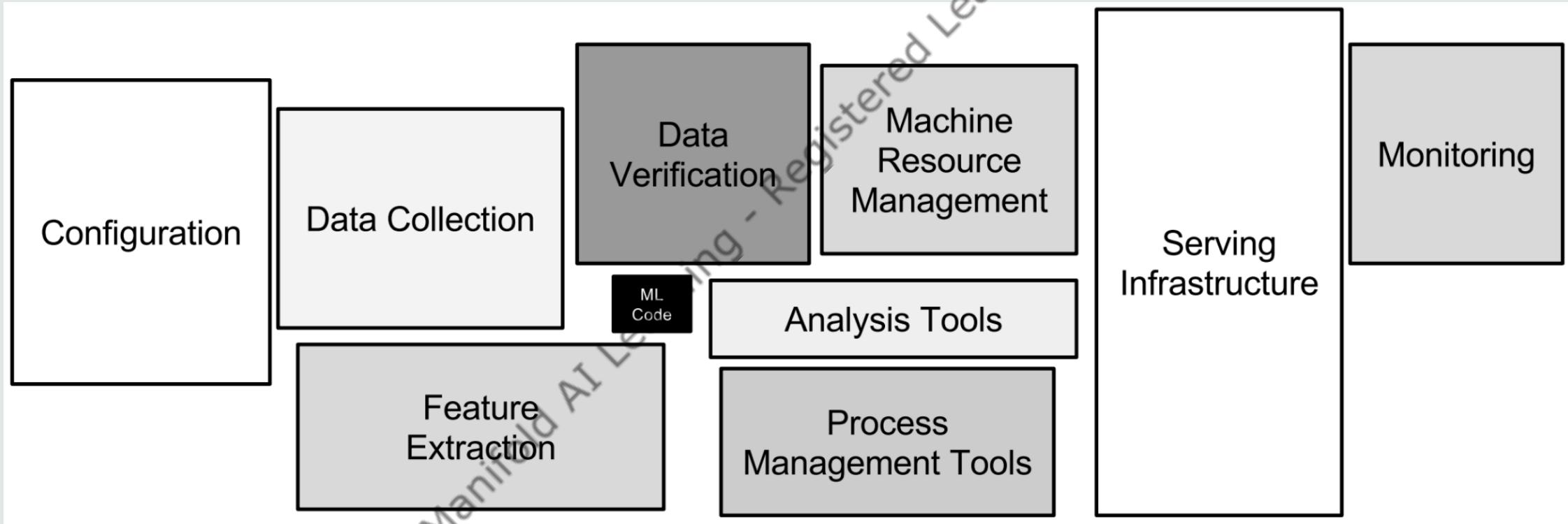
Achieving maximum visibility into your deployed model is a necessity for identifying potential issues and their origins before they have a detrimental impact on the business.

Why Monitoring Machine Learning System is Difficult ?

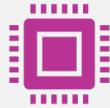
Manifold AI Learning - Registered Learners Only



Hidden Technical Debt of Machine Learning



Summary of Hidden Technical Debt of Machine Learning



The hidden technical debt in machine learning refers to the unforeseen and often overlooked challenges and complexities that arise as machine learning models are deployed and maintained in real-world applications.



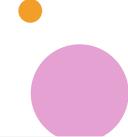
This debt can accumulate due to various factors, such as data quality issues, model performance degradation, changing environments, and evolving business requirements.



It highlights the need for ongoing monitoring, maintenance, and adaptation of machine learning systems to ensure they continue to perform effectively and meet their intended objectives.



Addressing this hidden technical debt is crucial to avoid unexpected issues and maintain the reliability of machine learning solutions over time.



Machine Learning System Behavior



Data (Machine Learning Context): The performance of a machine learning system is heavily influenced by both the dataset used for training the model and the ongoing data input it receives during production.



Model (Machine Learning Context): In machine learning, the model is the result of applying a machine learning algorithm to a dataset. It encapsulates the knowledge gained by the algorithm. It's beneficial to perceive the model as a comprehensive pipeline, comprising all the necessary steps for managing data flow into and out of the model.



Code: To construct the machine learning pipeline and specify the model configurations for training, testing, and evaluating models, code is an essential component.

Challenges in Machine Learning Systems



It's not as simple as saying – We have 3 additional dimensions on dataset when building ML Model.



Code and Configuration – introduces further complexity and sensitivity into machine learning system due to
Entanglements & Configurations



Entanglements: Changes in input data distributions can significantly affect a model's accuracy and lead to shifts in its predictions, emphasizing the importance of thorough testing for feature engineering and selection code to account for these effects.



Configurations: Flaws in a model's configuration, including hyperparameters, versions, and features, can dramatically alter the system's behavior. Importantly, such issues may go unnoticed in traditional software testing, allowing a machine learning system to generate valid yet incorrect outputs without raising exceptions.

Challenge – Who Owns What ?

- Since ML Projects involves multiple stakeholders, and each stakeholder will have completely different perspective about “monitoring” based on the business areas and responsibilities.

Manifold AI Learning - Registered Learners Only



From Data Scientists Perspective!!

Data scientists focus on functional objectives, including changes in input data, the model, and model predictions.

Monitoring functional objectives entails visibility into data input, model metrics, and understanding the model's predictions.

Model accuracy in a production environment is a primary concern for data scientists.

Real-time access to true labels is ideal for insight but is not always available.

Data scientists often use proxy values to gain visibility into their models in the absence of real-time true labels.

DevOps Engineer's Perspective

Manifold AI Learning ® Registered Learners Only



Engineers are tasked with achieving operational objectives to maintain the health of machine learning system resources.



This involves monitoring standard software application metrics, a common practice in traditional software development.



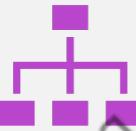
Examples metrics :

- Latency
- IO/memory/disk usage
- System reliability (uptime)
- Auditability

Best Practice!!



Effective monitoring of machine learning systems incorporates both data scientist and DevOps engineer perspectives.



A comprehensive understanding of the system is essential for success.



Collaboration among all stakeholders is crucial to establish clear and consistent definitions of terms, fostering effective communication within the team.

Manifold AI Learning® Registered Learner Only

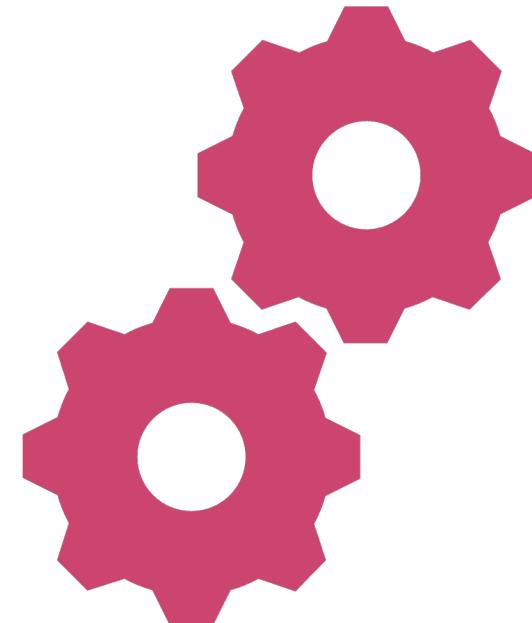


What Needs to be monitored in Production ?

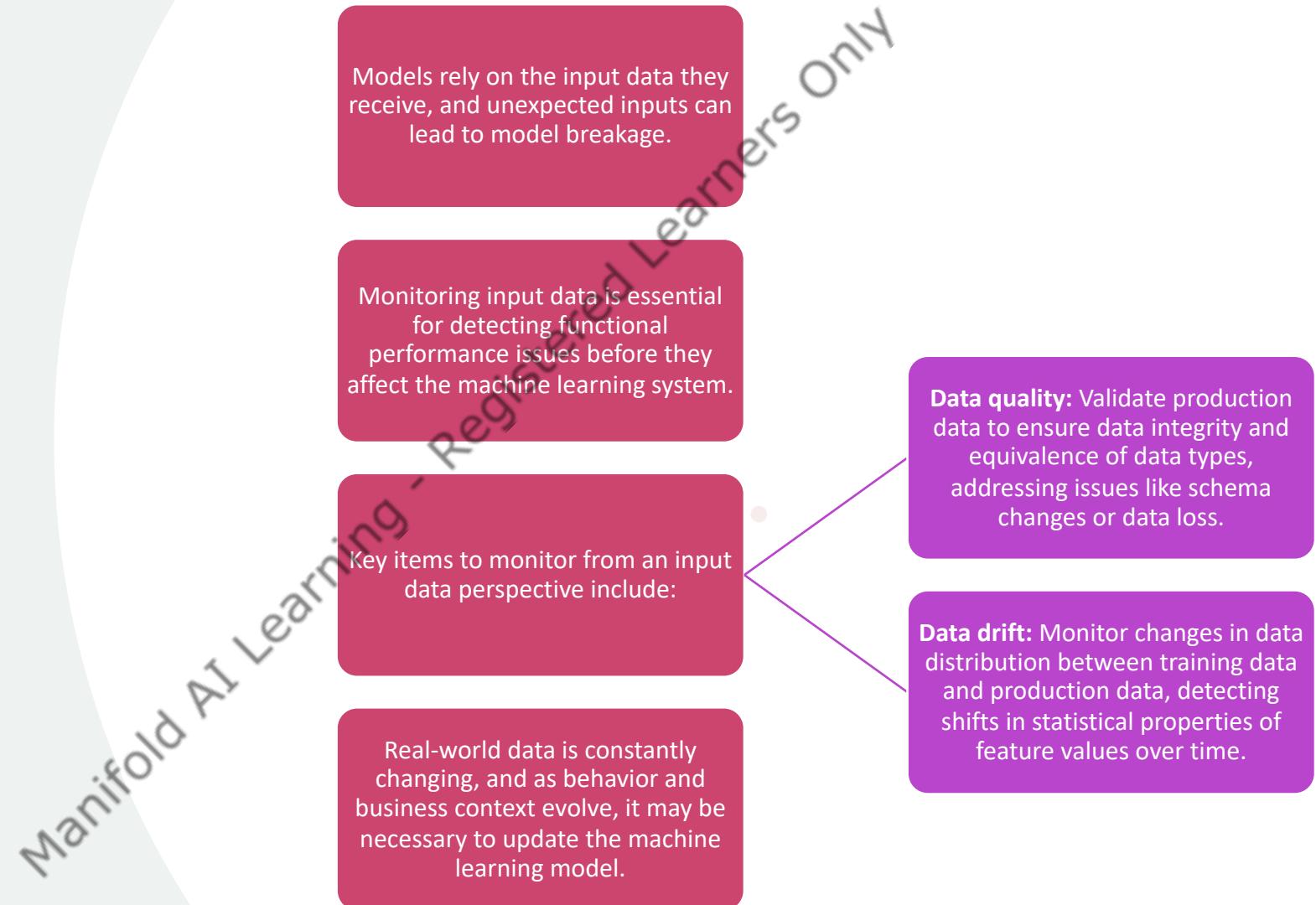
What needs to be monitored in production?

- Two Levels of Monitoring:
 - Functional : The Input, the Model and the output predictions
 - Operational : System Performance, Pipeline and Costs

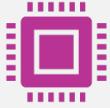
Manifold AI Learning - Registered Learners Only



Functional Level Monitoring – Input Data



Functional Level Monitoring – The Model



The core of a machine learning system is the machine learning model, which must consistently perform above a certain threshold to deliver business value.



Ongoing **monitoring** is required to address factors that can impact the model's performance, including model drift and version management.



Model drift is the decline in a model's predictive accuracy caused by changes in the real-world environment, and it should be detected using statistical tests and monitored for predictive performance over time.

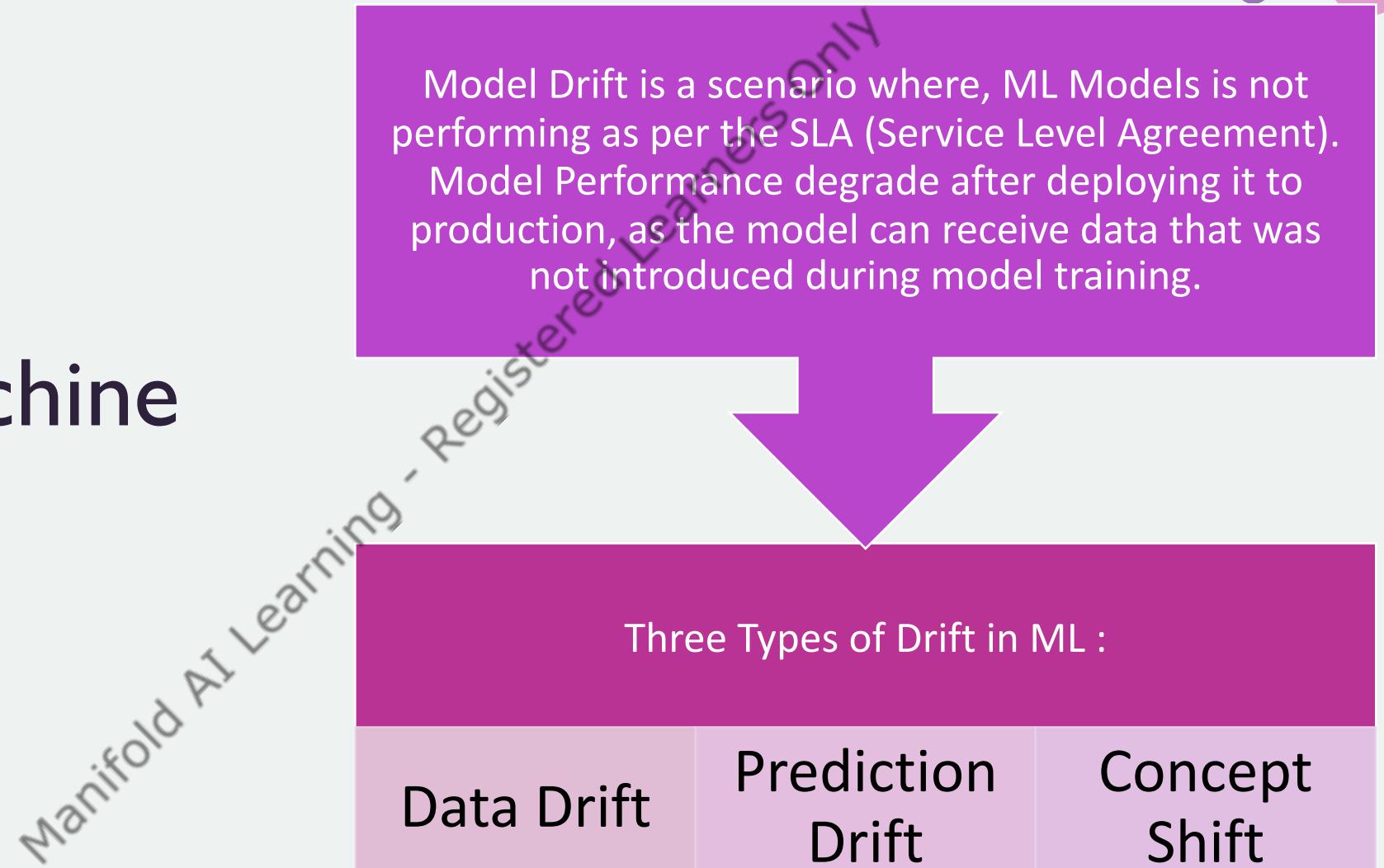


Tracking versions is crucial to ensure the correct model is in production, involving the management of version history and prediction records.

Functional Level Monitoring – The Output

- Understanding the performance of a machine learning model in production entails monitoring its output, ensuring it aligns with key performance metrics.
- **Ground truth:** In cases where ground truth labels are available, such as in ad click prediction, model predictions can be compared directly to the actual outcomes to assess performance.
- However, evaluating model predictions against ground truth is challenging in most machine learning scenarios, necessitating alternative methods.
- **Prediction drift:** When ground truth labels are unavailable, monitoring predictions is crucial. A significant shift in prediction distribution can indicate potential issues, like changes in input data structure, system misbehavior, or shifts in the real-world environment, as seen in fraud detection with a sudden increase in flagged transactions.

Drift in Machine Learning



Data Drift



Data drift, also known as feature drift, population drift, or covariate shift, occurs when the distribution or characteristics of input features change in relation to the training data.



Such changes can affect model predictions because the model isn't prepared for the new data distribution.



For example, if a new category is introduced as a feature post-deployment, it can lead to prediction errors since the model wasn't trained with this category.



Changes in critical features, like Credit Rating, can substantially impact the model output. For instance, switching from S&P to Moody's credit rating alters the input data distribution, affecting model performance.



In summary, data drift highlights the importance of ensuring that the distribution of training data matches the distribution of reference (production) data to maintain model accuracy.

Prediction Shift



Data drift can lead to changes in the target or prediction variable over time, resulting in prediction drift.



Prediction drift is also known as prior probability shift, label drift, or unconditional class shift.



It can occur due to the removal or addition of new classes in the data.



Retraining the model is a strategy to mitigate the impact of prediction drift on model performance.



In summary, prediction drift signifies changes in predictions as input data evolves, and it can be addressed through model retraining to maintain accuracy.

Concept Shift



Concept shift, also called posterior class shift, conditional change, or real concept drift, occurs when the relationship between independent variables and dependent variables changes.



It involves alterations in the connection between input and target variables.



Significant concept shifts can lead to unreliable model predictions.



The concept in concept shift refers to the relationship between independent and dependent variables.

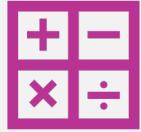
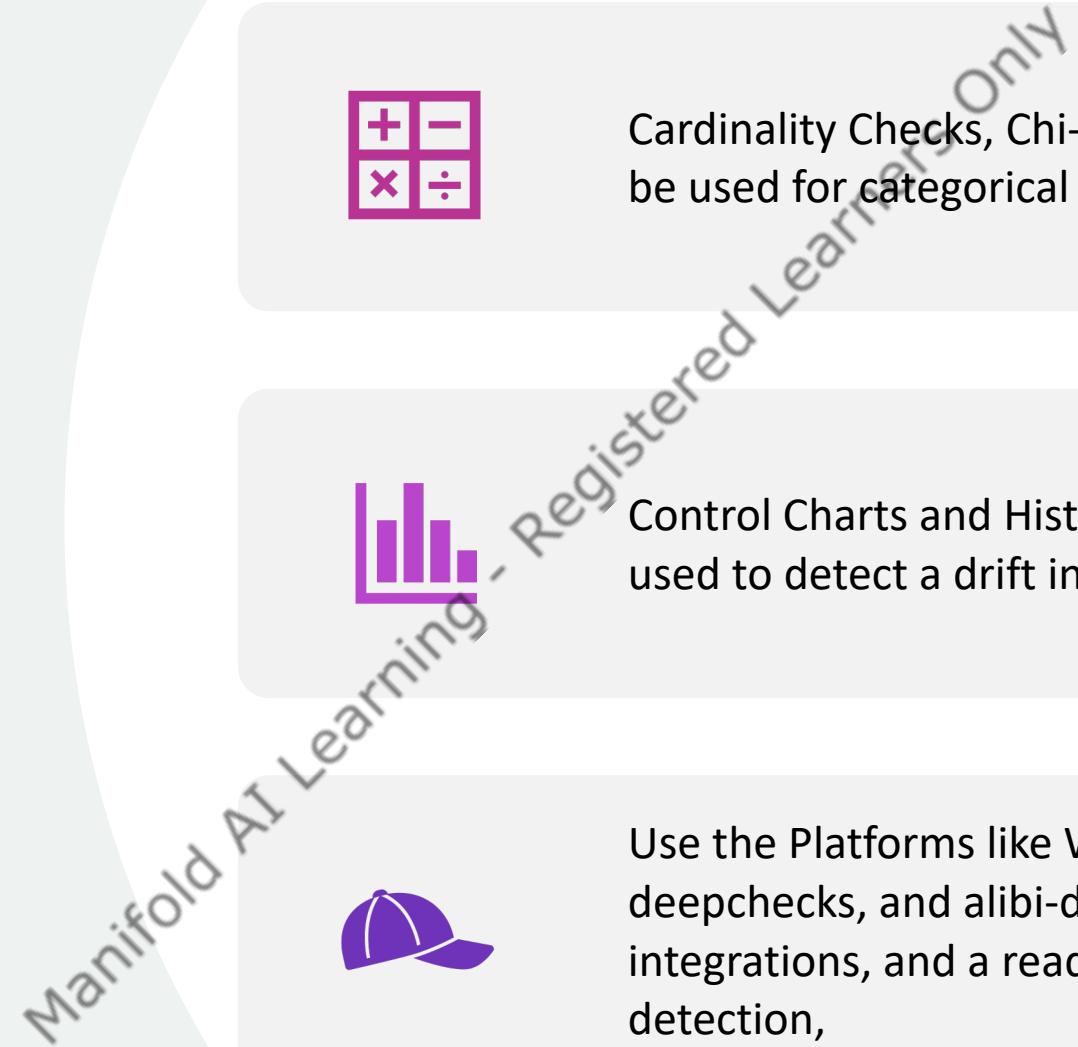


For example, changes in a car insurance company's claim policy can result in a concept shift, where the mapping between input features and target features changes, even if the input data distribution remains the same.

Techniques to detect drift in ML

-  Statistical distance metrics are valuable for detecting drift in machine learning.
-  For datasets with numerous independent variables, dimensionality reduction techniques like PCA can be employed.
-  Monitoring many features can strain the system, making it challenging to address drift by focusing on specific features.
-  Basic statistical metrics such as mean, standard deviation, correlation, and comparisons of minimum and maximum values can gauge drift between training and current independent variables.
-  Distance measures like Population Stability Index (PSI), Characteristic Stability Index (CSI), Kullback–Leibler divergence (KL-Divergence), Jensen–Shannon divergence (JS-Divergence), and Kolmogorov-Smirnov (KS) statistics are suitable for continuous features when assessing drift.

Techniques to detect drift in ML (Contd.)



Cardinality Checks, Chi-Squared Test, and entropy can be used for categorical variables



Control Charts and Histogram intersections can be used to detect a drift in data



Use the Platforms like WhyLabs, and libraries like deepchecks, and alibi-detect, which comes with easy integrations, and a ready framework for drift detection,

Addressing the Drift Issues

Manifold AI Learning - Registered Learners Only

Addressing the Drift Issues



Data quality issues can be resolved if there are problems with input data. For example, transitioning from high-resolution training images to low-resolution deployment images can be addressed.



Retraining the model is a strategy to enhance its performance after detecting data or concept shifts.



When production data is insufficient for training, you can combine historical data with recent production data, giving more weight to recent information.

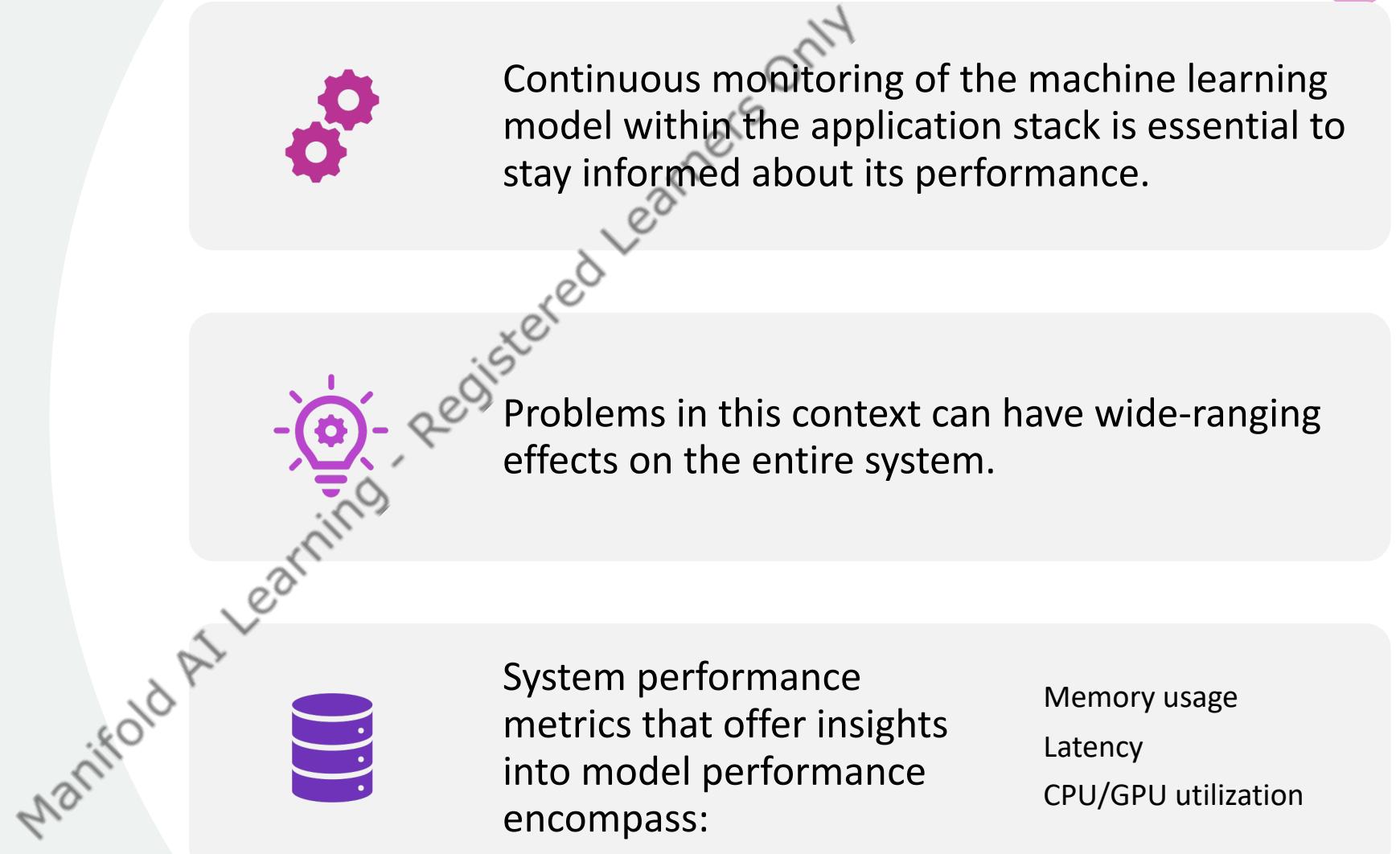


Four strategies for retraining the model include **periodic retraining** at scheduled times, **event-driven retraining** when new data becomes available, **model or metric-driven retraining** based on accuracy or SLA thresholds, and **online learning for continuous real-time** or near real-time model improvement.



If retraining doesn't yield satisfactory results, **rebuilding or tuning the model** on recent data may be necessary, and this process can be automated using a pipeline.

Operational Level Monitoring – The System Performance



Operational Level Monitoring – The Pipelines

Essential pipelines for monitoring in machine learning are the data pipeline and the model pipeline.

Neglecting data pipeline monitoring can result in data quality problems that disrupt the system.

For the model pipeline, it's vital to track and monitor factors that could lead to model failure in production, including dependencies and associated issues.

Operational Level Monitoring – The Costs

Machine learning involves financial costs spanning from data storage to model training.

While machine learning can deliver substantial value, it can also become expensive, making continuous cost monitoring crucial.

Setting budgets through cloud vendors like AWS or GCP, with bill and spending tracking, allows for timely alerts when budgets are exceeded.

For on-premises machine learning applications, monitoring system usage and cost provides insight into cost breakdown and potential cost-cutting measures.

Tools for Monitoring Machine learning Models

Manifold AI Learning - Registered Learners Only



Prometheus



Grafana

WHYLABS
whylogs

Best Practices for Machine Learning Model Monitoring

Monitoring starts during model development and is not limited to the deployment phase.

Tracking and monitoring metrics and logs are essential throughout the iterative process of building a machine learning model.

Sudden major performance degradation in a model is a red flag, requiring immediate investigation.

Teams should establish a troubleshooting framework to guide them from alert to resolution during model maintenance.

A well-defined plan of action is crucial for responding to system failures, ensuring a prompt transition from alert to problem-solving and effective model maintenance.



Grafana

Next Section : Continuous Monitoring
with Prometheus & Grafana

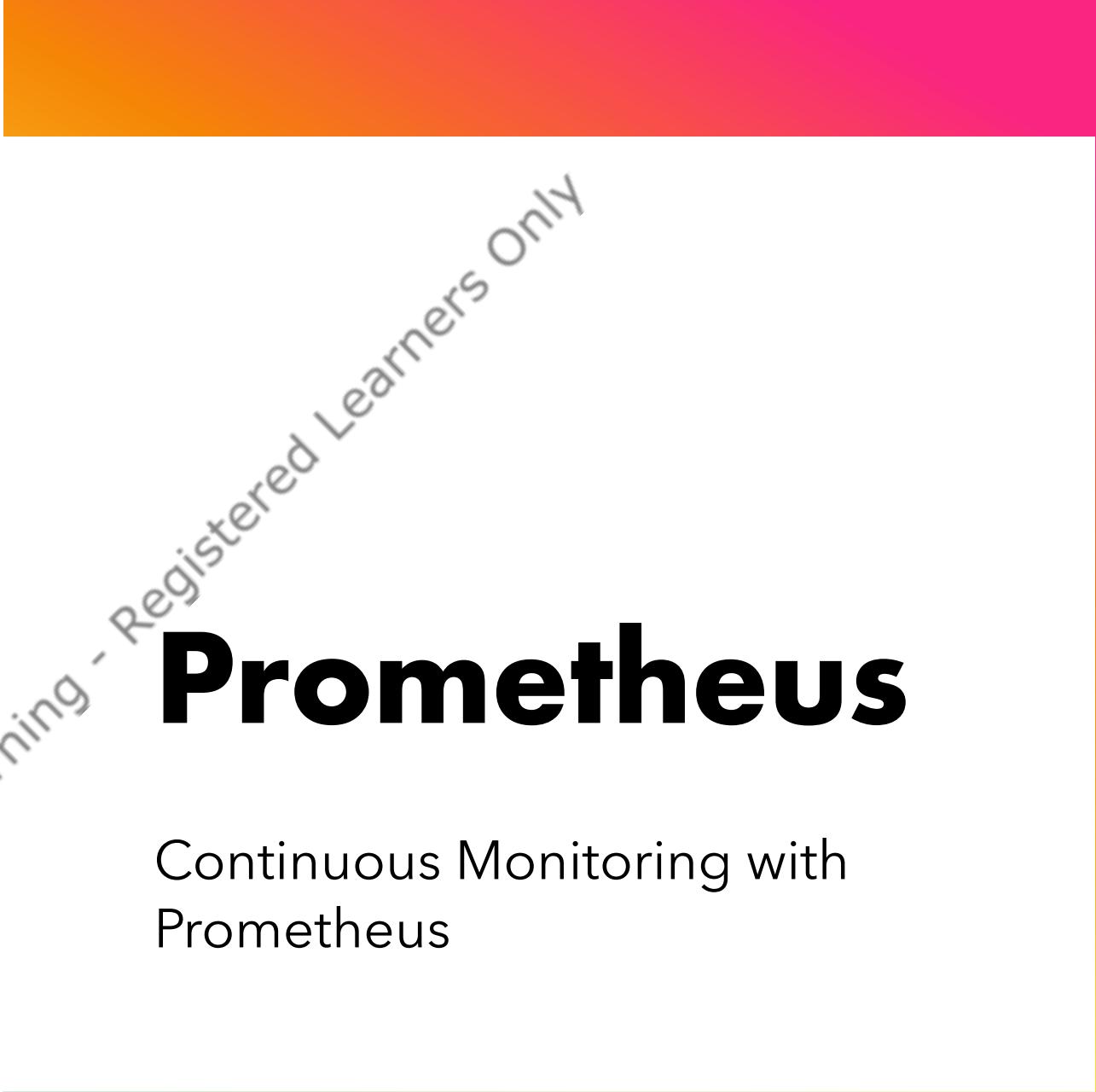
Manifold AI Learning - Registered Learners Only



Prometheus



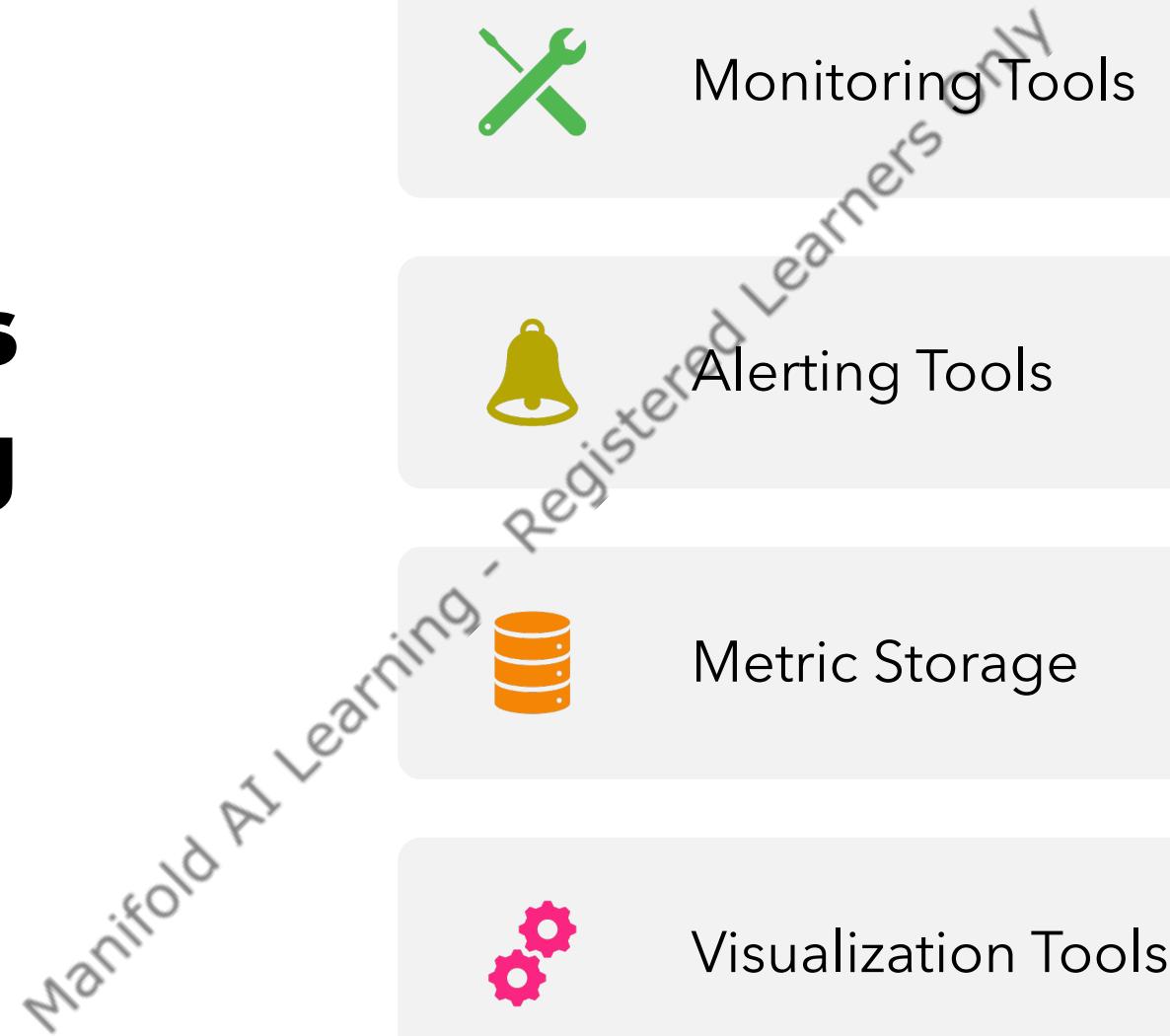
Manifold AI Learning ®



What is Continuous Monitoring ?

- + An automated process by which one can observe and detect compliance issues and security threats during the each phase of DevOps/MLOps pipeline

Continuous Monitoring Tools in DevOps





Sensu

by sumo logic



Nagios

Monitoring Tools

Manifold AI Learning - Registered Learners Only



Alerting Tools

Manifold AI Learning - Registered Learners Only

Metric Storage

Manifold AI Learning - Registered Learners Only

splunk®

aws

 influxdb

Visualization Tools



Monitoring issues at Manifold AI Learning

Manifold AI Learning

-  Manifold AI Learning is a EdTech company that creates a Edtech products for Employee Upskilling
-  The Company was founded in 2020 and currently has more than 50000 learners worldwide
-  Such a massive operations is handled using the combination of Physical data centers and the cloud

Manifold AI Learning : Monitoring Issues

- + The Old Infrastructure required an extremely labour intensive, difficult to scale the storage system.
- + This system was very unreliable in getting queries and providing the metrics on time
- + This meant that the system required a lot of hand-holding on a regular basis

Solution: **Prometheus** **and Grafana**



Prometheus provided the company an easy way to collect all the required metrics for day-to-day operations



Prometheus also provided the ability to monitor the modern system as well as legacy system



Alerting mechanism in the new monitoring system was easy to embed with the dashboards



Grafana made the dashboard visualization more accessible for various teams in the organization

Manifold AI Learning - Registered Learner.com



Manifold AI Learning - Registered Learners Only

Prometheus

Prometheus

- An **open source monitoring** solution
- Started at SoundCloud around 2012-13, and was made public in early 2015
- Prometheus provides **Metrics & Alerting**
- It is inspired by Google's **Borgmon**, which uses time-series data as a data source, to then send alerts based on this data.

Prometheus

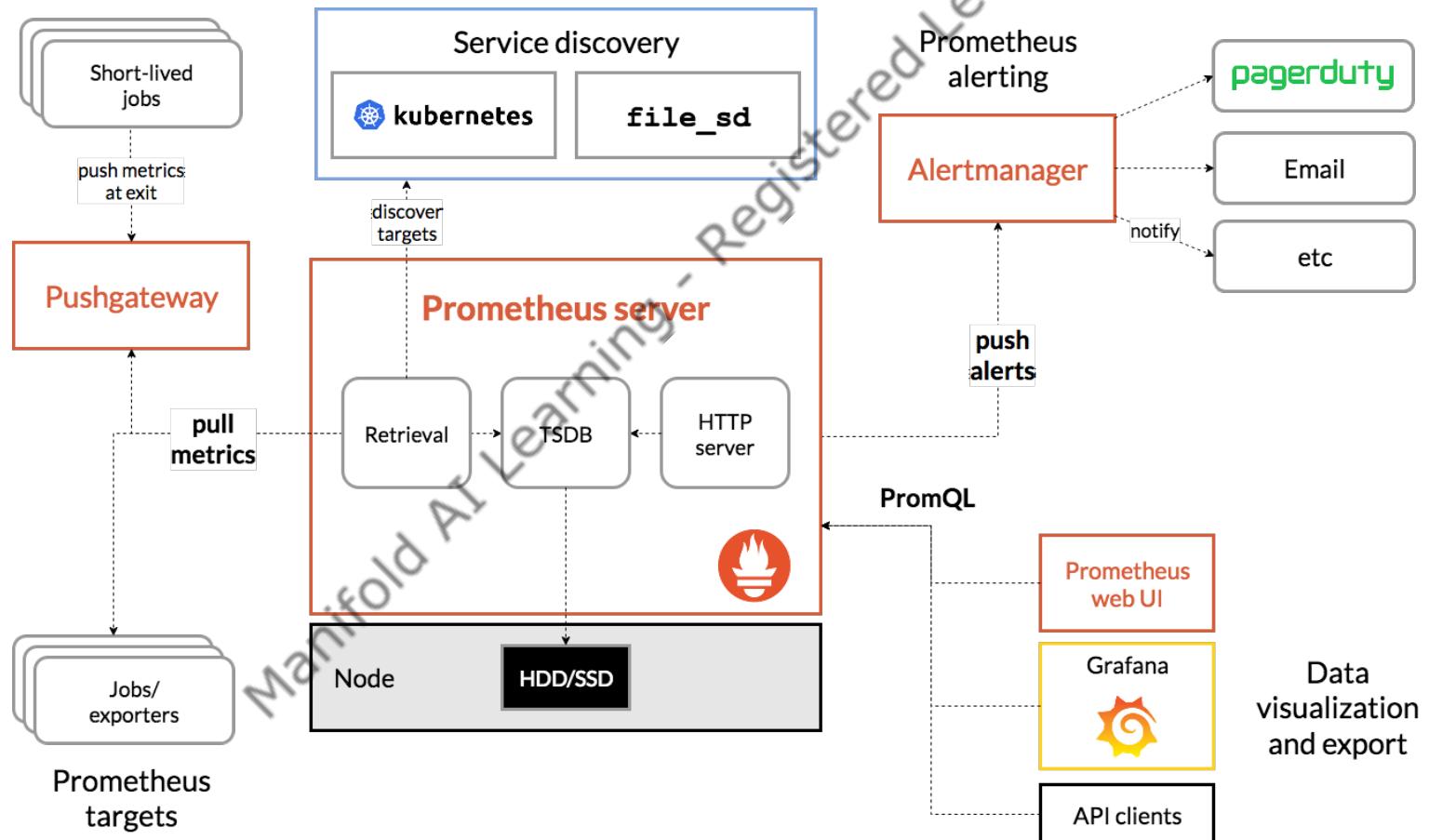
- + In Prometheus we work with **Dimensional data**: Time series are identified by metric name and a set of key/value pairs.

Metric Name	Label	Sample
CPU Usage	System=1	65

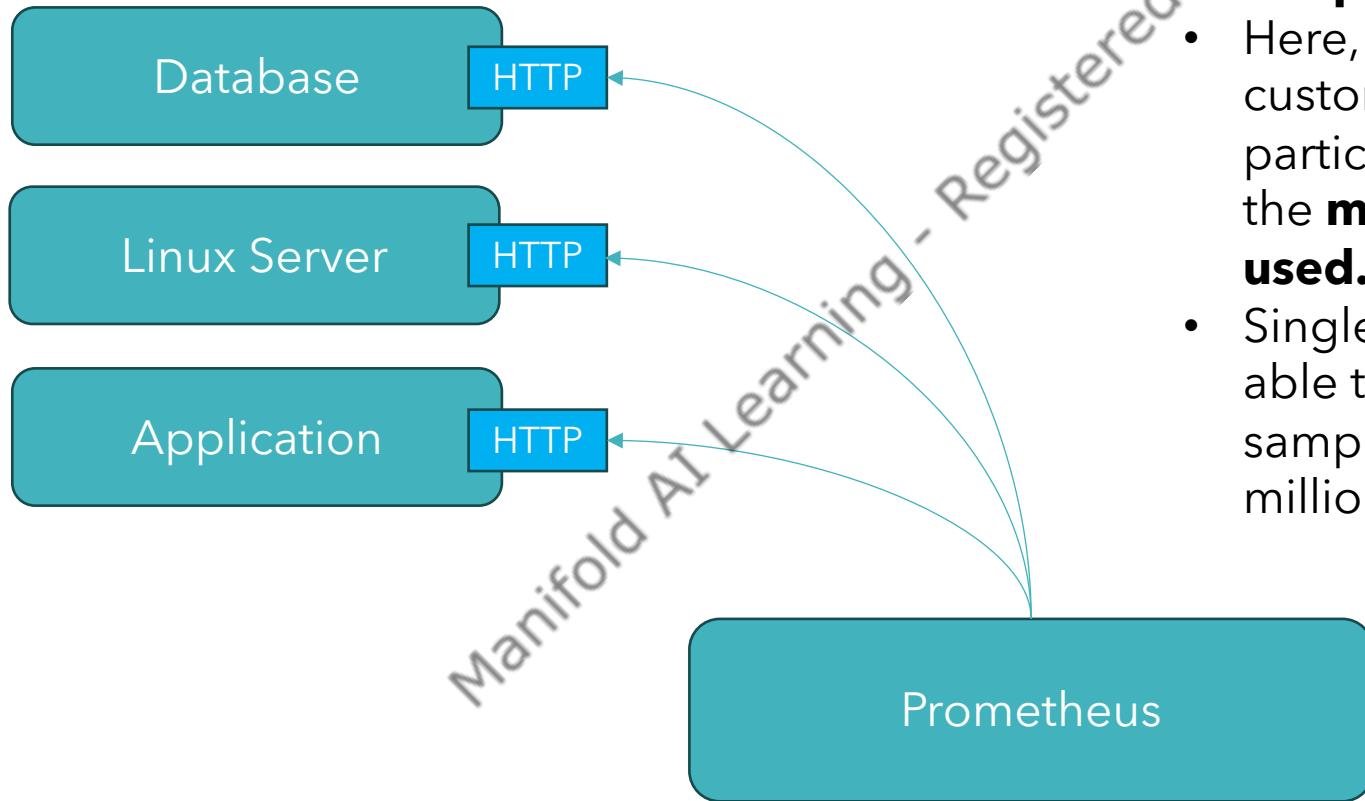
Prometheus

- + Prometheus includes a Flexible Query Language - **PromQL**
- + Can generate visualizations using Built-in expression browser, or can be integrated with Grafana
- + It Stores metrics in **memory** and **Local Disk** in an own **custom, efficient format**
- + Written in **Go**
- + Supports multiple **client libraries** and **integrations available**

Architecture of Prometheus



Prometheus Server



- Prometheus collects the metrics from monitored targets by **scraping the metrics HTTP endpoints**
- Here, instead of running the custom scripts that check on particular services & systems, the **monitoring data itself is used**.
- Single Prometheus server is able to ingest upto one million samples per second as several million time series

Prometheus Push Gateway

- + The Prometheus Pushgateway exists to allow ephemeral and batch jobs to expose their metrics to Prometheus.
- + Since these kinds of jobs may not exist long enough to be scraped, they can instead push their metrics to a Pushgateway.
- + The Pushgateway then exposes these metrics to Prometheus.

Exporters and Integrations



There are a number of libraries and servers which help in exporting existing metrics from third-party systems as Prometheus metrics.



This is useful for cases where it is not feasible to instrument a given system with Prometheus metrics directly (for example, HAProxy or Linux system stats).

<https://prometheus.io/docs/instrumenting/exporters/>

Prometheus Alertmanager

- + The Alertmanager handles alerts sent by client applications such as the Prometheus server.
- + It takes care of deduplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie.
- + It also takes care of silencing and inhibition of alerts.

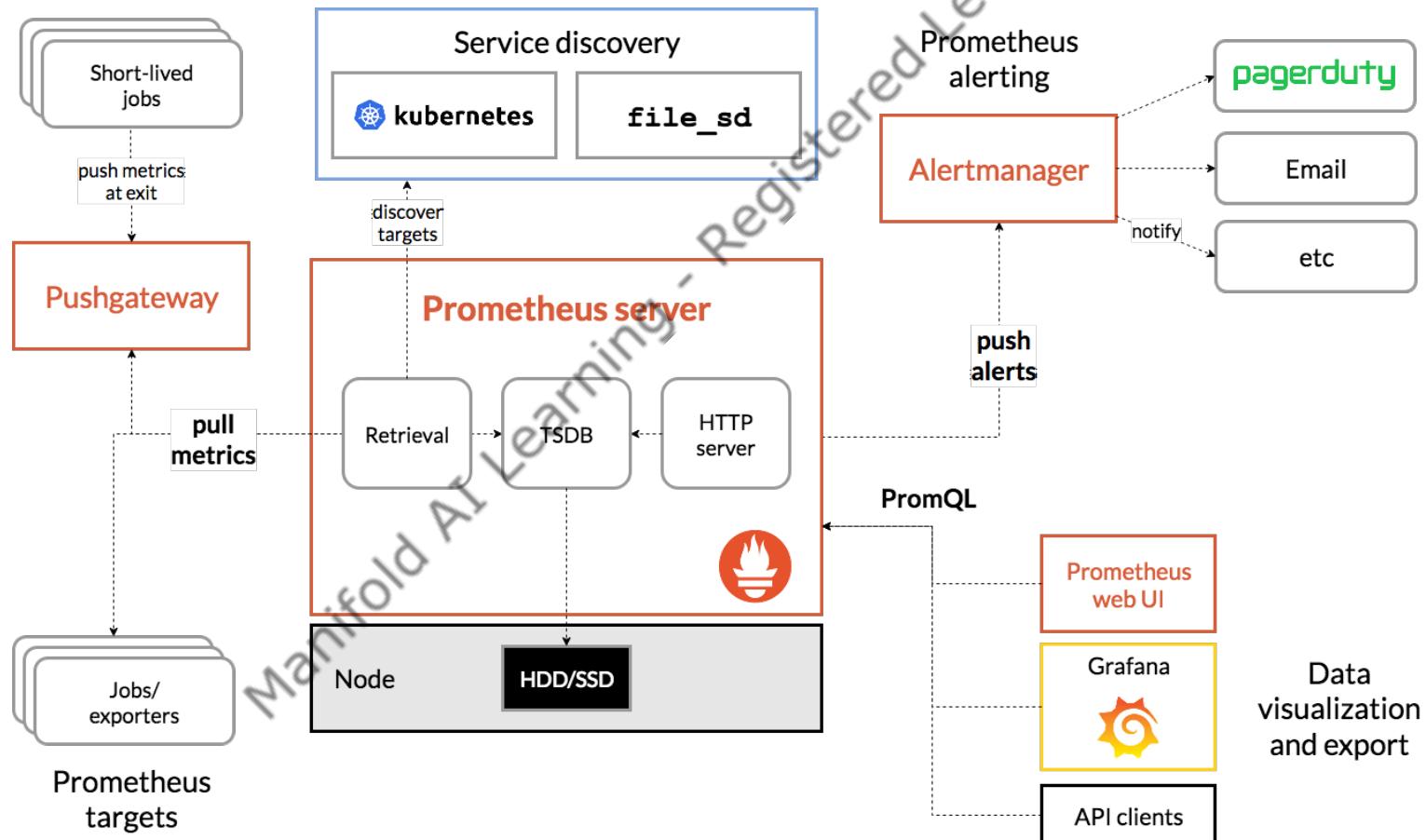
Service Discovery

- + Prometheus service discovery is a standard method of finding endpoints to scrape for metrics
- + We need to configure **prometheus.yaml** and custom jobs to prepare for scraping endpoints in the same way we do for native Prometheus.

PromQL

- + PromQL, short for Prometheus Querying Language, is the main way to query metrics within Prometheus.
- + You can display an expression's return either as a graph or export it using the HTTP API.

Architecture of Prometheus



Prometheus Metrics and Its Types

- + The Prometheus client libraries offer four core metric types.

Counter

Gauge

Histogram

Summary

Manifold AI Learning - Registered Learners Only



Counter



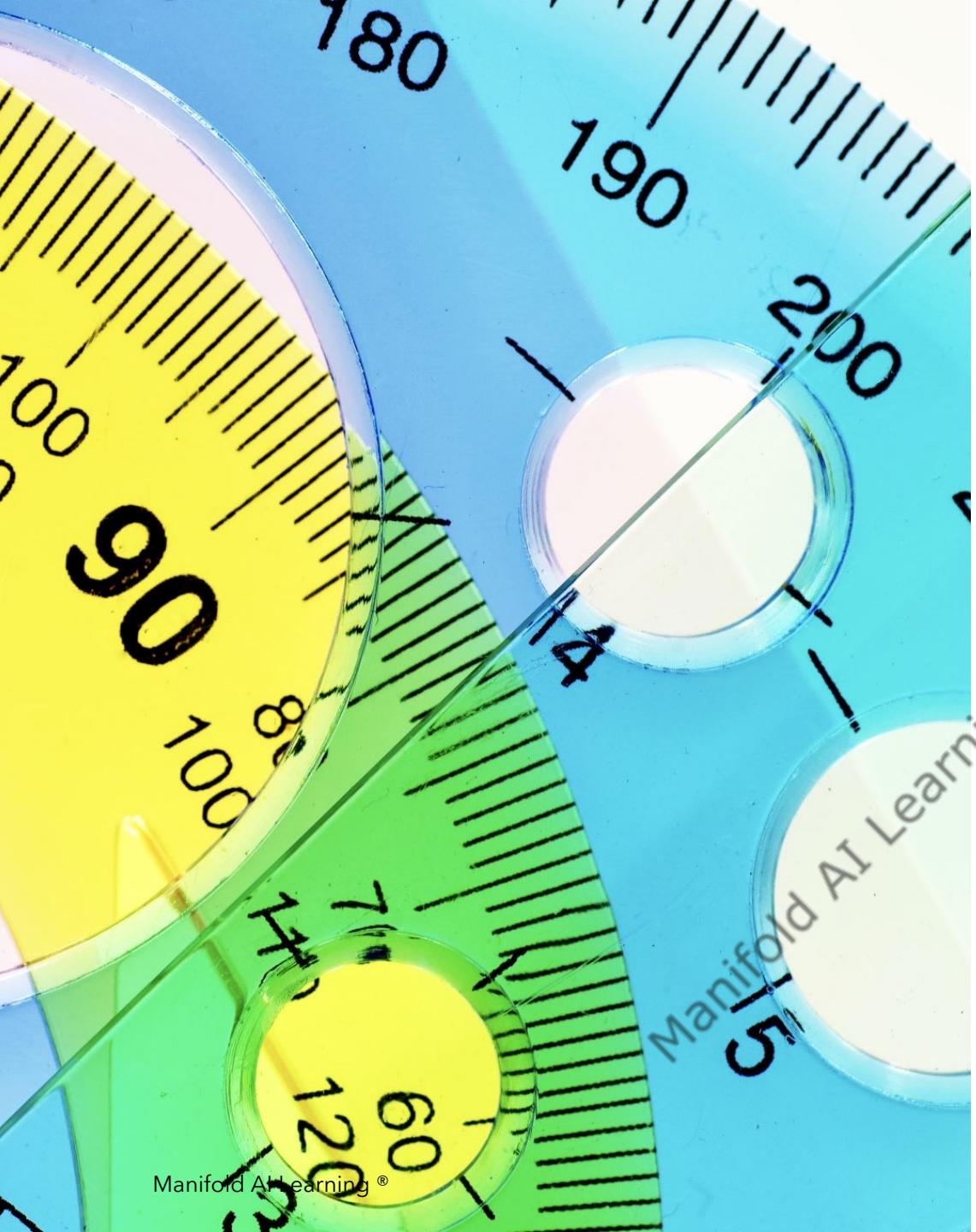
A counter is a cumulative metric that represents a single monotonically increasing counter whose value can only increase or be reset to zero on restart.



Example: you can use a counter to represent the number of requests served, tasks completed, or errors.



Do not use counter to expose a value that can decrease. (like - Number of currently running process)



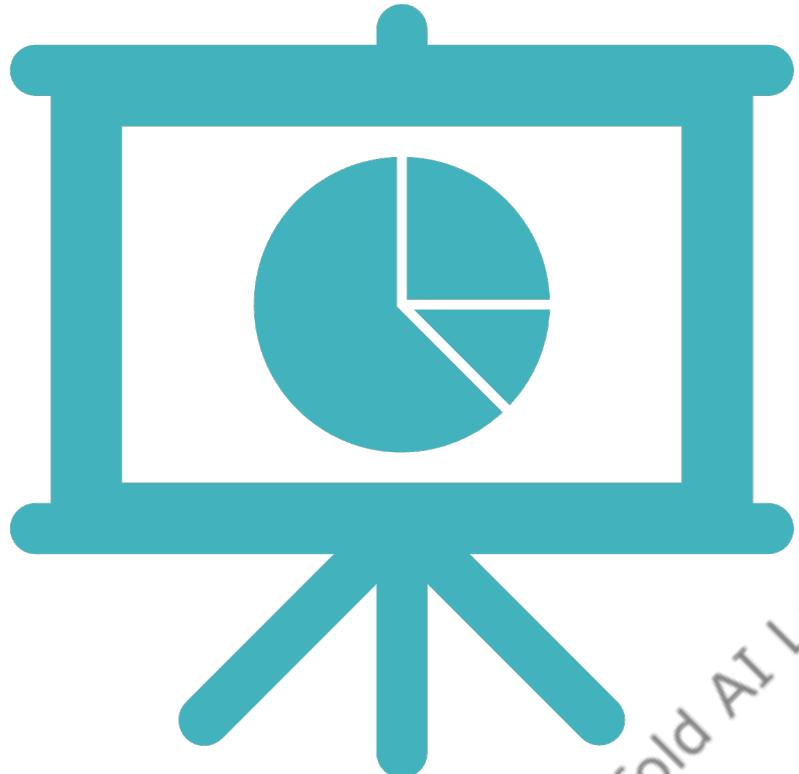
Gauge

- + A gauge is a metric that represents a single numerical value that can arbitrarily go up and down.
- + Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests.



Histogram

- + A histogram samples observations (usually things like request durations or response sizes) and counts them in configurable buckets. It also provides a sum of all observed values.



Summary

- + Similar to a histogram, a summary samples observations (usually things like request durations and response sizes).
- + It also provides a total count of observations and a sum of all observed values, it calculates configurable quantiles over a sliding time window.



Manifold AI Learning - Registered Learners Only

Installation of Prometheus

Grafana



Grafana

Manifold AI Learning - Registered Learners Only



Grafana

- + Its a Multi-platform open source analytics and interactive visualization web application.
- Grafana provides:

Charts

Graphs

Alerts

Features of Grafana



Grafana has many visualization options to help you understand your data, beautifully



Seamlessly define alerts when it make sense - while you're in the data

Other Features

- Manifold AI Learning - Registered Learner Only
-  Discover hundreds of Dashboards and plugins in the official library
 -  Share data & Dashboards across the teams
 -  Supports multiple databases, natively.

Installation of Grafana



Prometheus Configuration



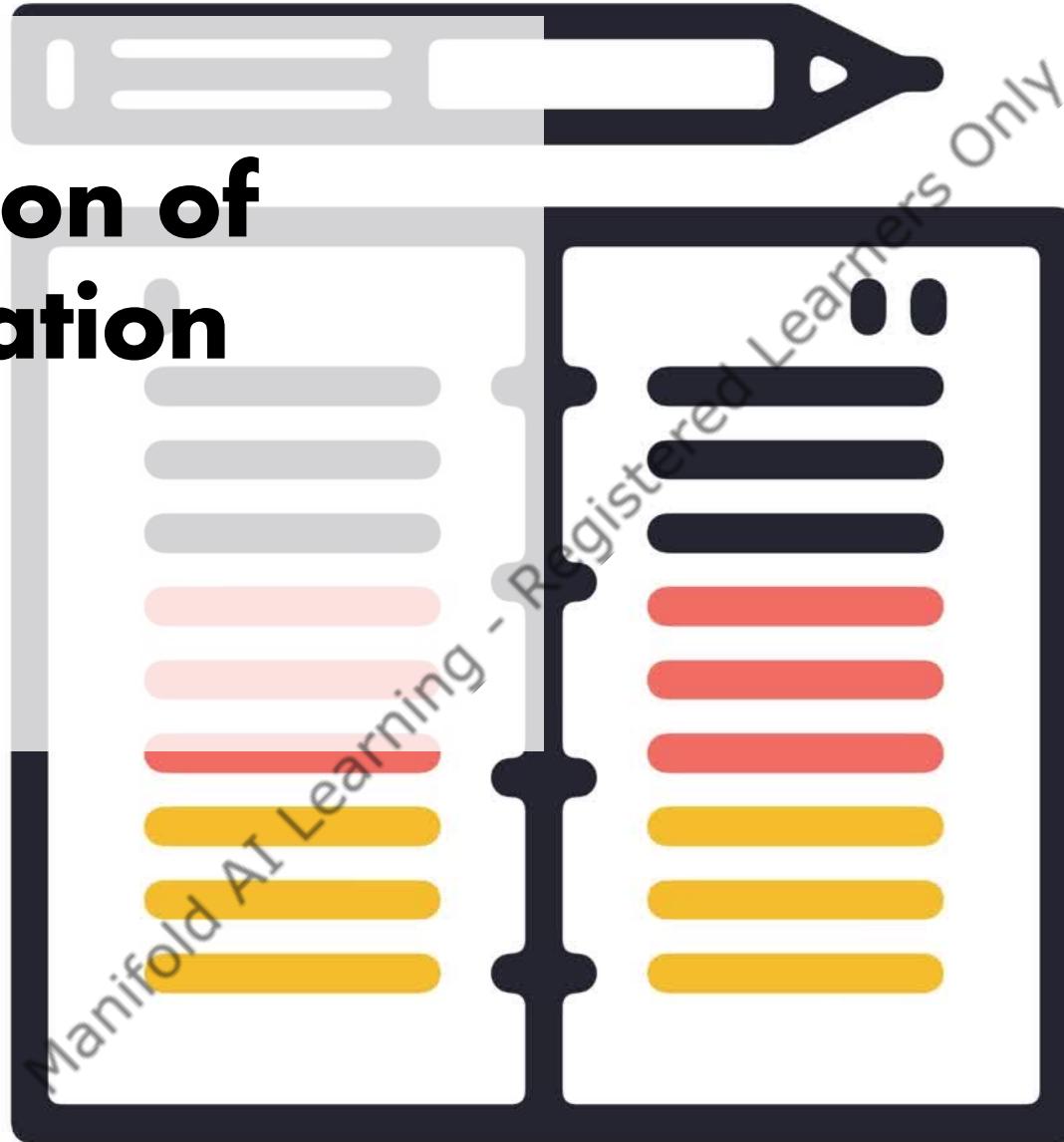
Prometheus Configuration

- + The Configuration is stored in Prometheus configuration file, in **yaml** format
- + Configuration can be **changed and applied**, without having to restart Prometheus
A reload can be done by executing `kill -SIGHUP <pid>`
- + You can also pass parameters (flags) at **startup time** to `./prometheus`

Prometheus Configuration

- + To scrape metrics, you need to add configuration to the Prometheus config file.
- + When installed - Prometheus scrapes metrics from itself.

Exploration of Configuration File





Manifold AI Learning
Registered Learners Only

Exploring the Basic Querying Prometheus

Monitor the Infrastructure using Prometheus



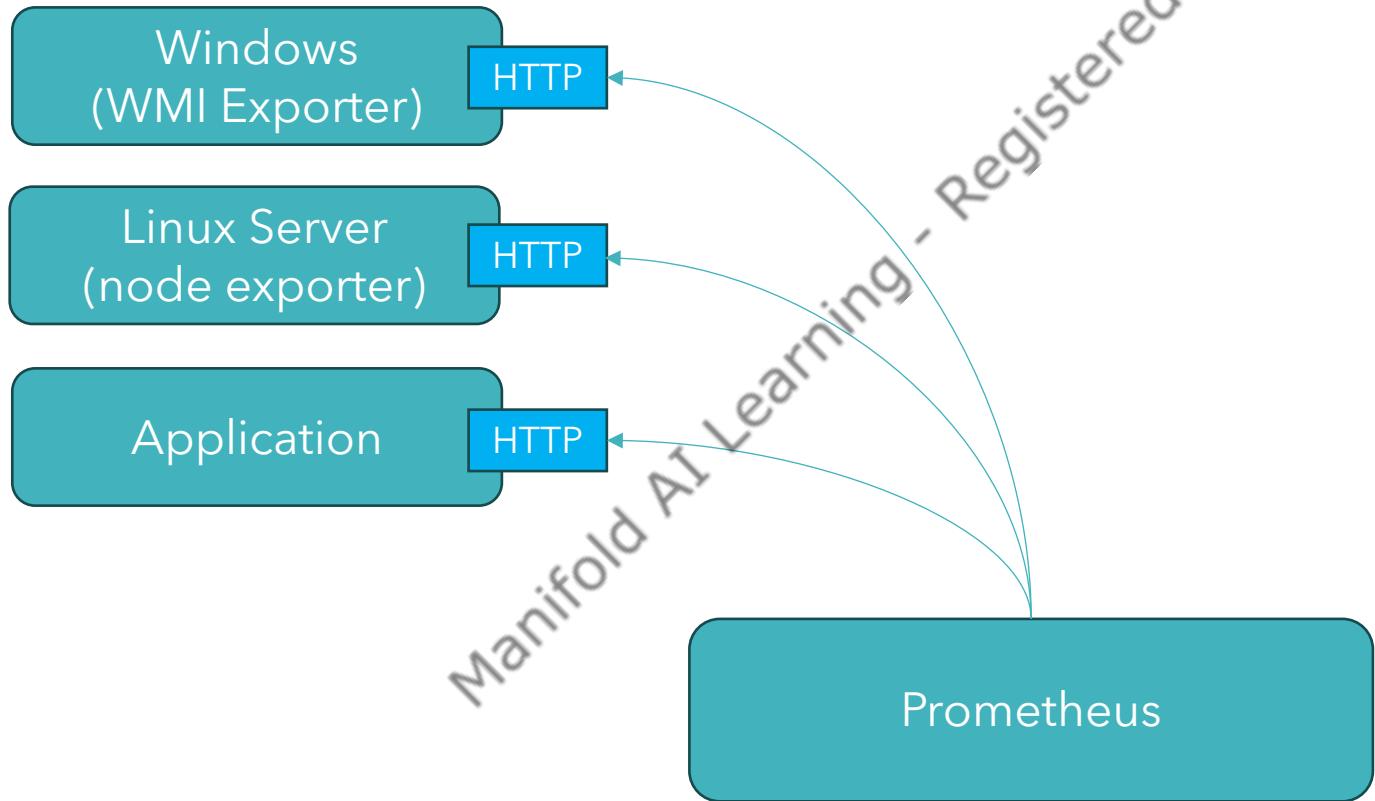
Manifold AI Learning - Registered Learners Only



Monitor the Nodes (Servers)

- + To Monitor the Nodes , you need to install node-exporter
- + The node exporter will expose machine metrics of Linux/*Nix machines
 - Example: CPU Usage, Memory Usage, etc.
- + Node Exporter can be used to monitor machines, and an alert can be configured based on these ingested metrics

Monitor Nodes



Working with Node Exporters

Manifold AI Learning - Registered Learners Only

Prometheus Monitoring



Manifold AI Learning - Registered Learners Only

Client Libraries



Before you can monitor your services, you need to add instrumentation to their code via one of the Prometheus client libraries.



These implement the Prometheus metric types.



Choose the Prometheus client library that matches the language in which your application is written.



Define & Expose internal metrics via HTTP endpoint on the application's instance

Official Client Libraries

- + Go
- + Java/Scala
- + Python
- + Ruby
- + Rust

Unofficial Client Libraries

- + Bash
- + C
- + C++
- + Dart
- + Node.js
- + Erlang
- + R

Example Code

```
from prometheus_client import start_http_server, Summary
import random
import time

# Create a metric to track time spent and requests made.
REQUEST_TIME = Summary('request_processing_seconds', 'Time spent processing request')

# Decorate function with metric.
@REQUEST_TIME.time()
def process_request(t):
    """A dummy function that takes some time."""
    time.sleep(t)

if __name__ == '__main__':
    # Start up the server to expose the metrics.
    start_http_server(8000)
    # Generate some requests.
    while True:
        process_request(random.random())
```

https://github.com/prometheus/client_python

Querying the Metrics using PromQL

Manifold AI Learning - Registered Learners Only

Monitor the Fast API Application using Prometheus

Manifold AI Learning - Registered Learners Only

Working with

Manifold AI Learning - Registered Learners Only



Grafana

Monitor ML Application using Application

Manifold AI Learning - Registered Learners Only

Deploy Applications with Docker Compose



Docker Compose



Compose is a Docker Tool for running Multi Container Applications



It uses a YAML formatted file to configure the Application Services



Using the commands, the user can create and start the services from the configurations

Docker Compose: Working



Manifold AI Learning ®
Docker Compose Only



Docker Compose works in three steps:

- Create **Dockerfile** for the application
- Define the services that make the application in the **docker-compose.yml**
- Run **docker-compose up** to start the applications

Multi-Container Deployment with Compose

- Docker Compose make use of project names to isolate environments
- The data for all the containers in a service is preserved in volumes
 - If a service has restarted but nothing has changed, Compose re-uses the existing containers
- We can use variables in the Compose files to customize the service for different environments

Compose Common Commands

- **docker-compose build** – Builds or rebuilds a service from the provided Dockerfile
- **docker-compose run** – Allows user to run a one-off command in the service
- **docker-compose up** – Creates and runs the service containers
- **docker-compose down** – Removes the containers, networks, images, and volumes related to the service

Hands On:

Deploy Multiple Containers using
Docker Compose

Manifold AI Learning® Registered Learners Only



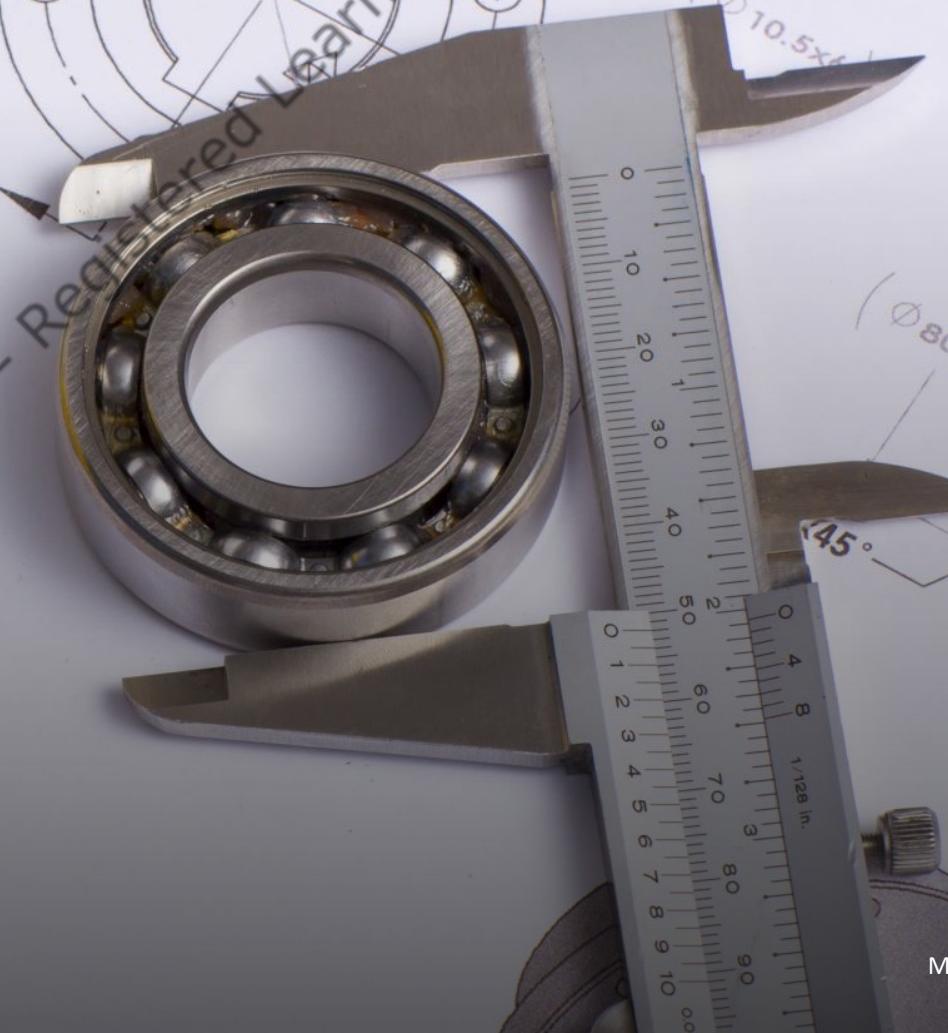
Docker Compose



Continuous Monitoring of Machine Learning Application

Architecture

Manifold AI Learning - Registered Learners Only



Manifold AI Learning ®

Intro to ML Monitoring

Getting Started with ML Monitoring & AI Observability

Manifold AI Learning - Registered Learners Only



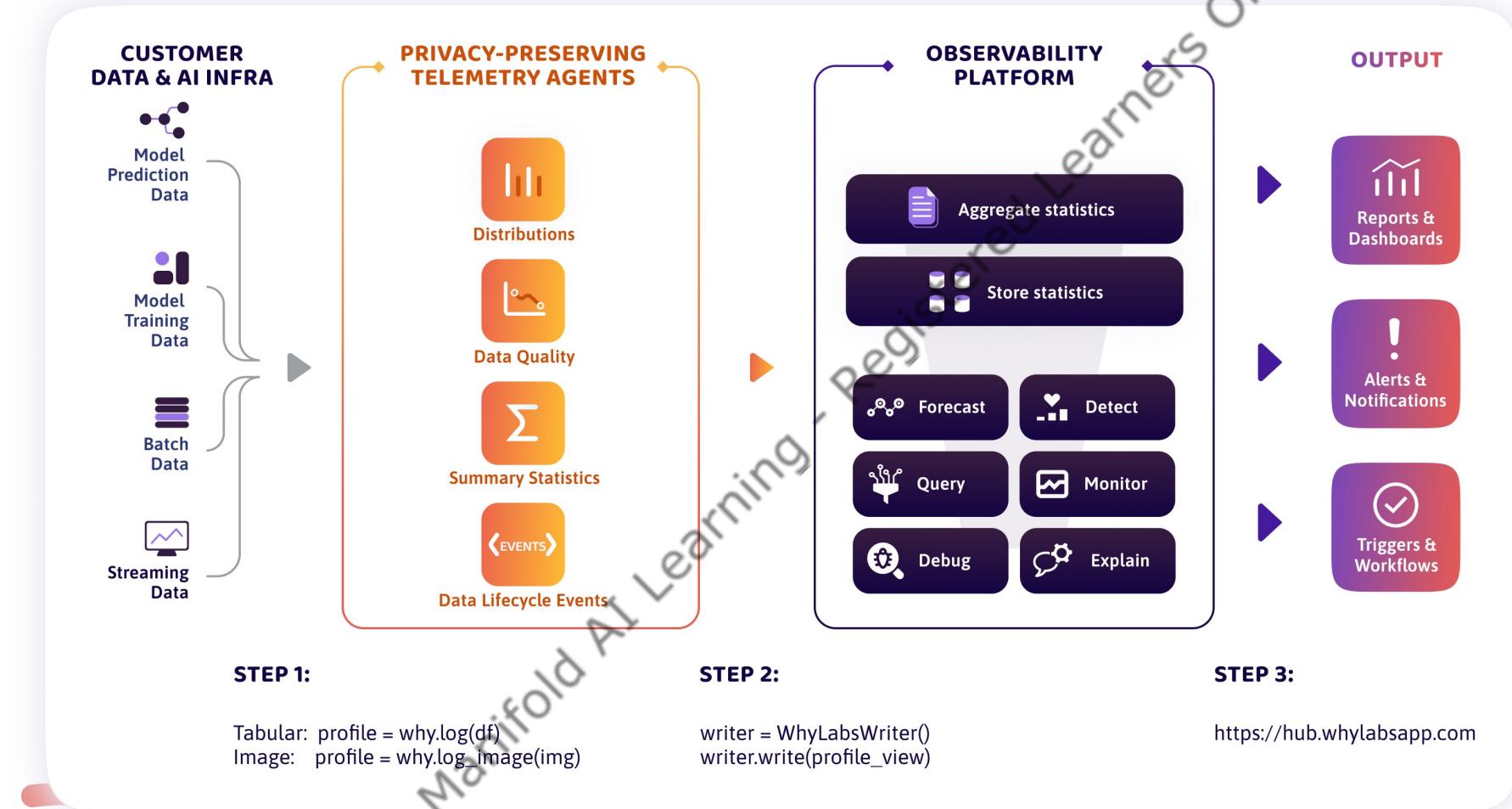
WHYLABS

Agenda

- Machine Learning Monitoring Concepts
- Data Drift, Model Outputs, and Performance
- Bias Tracing & Explainability
- Open Source ML Monitoring with Whylogs

Manifold AI Learning - Registered Learners Only

What is ML Monitoring ?



Source : whylabs
Manifold AI Learning ®

Why Need ML Monitoring ?

Manifold AI Learning ®



Model Inputs:

Data Drift
Data Quality
Schema



Model Outputs

Concept Drift
Volume
System Metrics



Model Performance

Accuracy/Recall
Bias/Fairness
Explainability
Business KPIs

Data Quality Instances

Manifold AI Learning ®

External Changes:

- An Input sensor/device for data collection causing bad data
- Data was manually entered wrong
- Lighting issues on the Camera

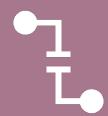
Schema Changes:

- Software library outputs a different format than before

Pipeline Bugs:

- Anywhere!

Data Drift Instances



Input Data no longer matches the distribution of the data that has been trained on.



In ML : Input Data → Training → Trained Model → Generate Output



Business Expansion/Marketing Event :

Seasonality Changes
New group of users
Change in Customer Preference

Concept Drift

- No change in input data, but output is not in-line with current requirement.
 - House Prices
 - Covid-19

Manifold AI Learning - Registered Learners Only

Bias Detection

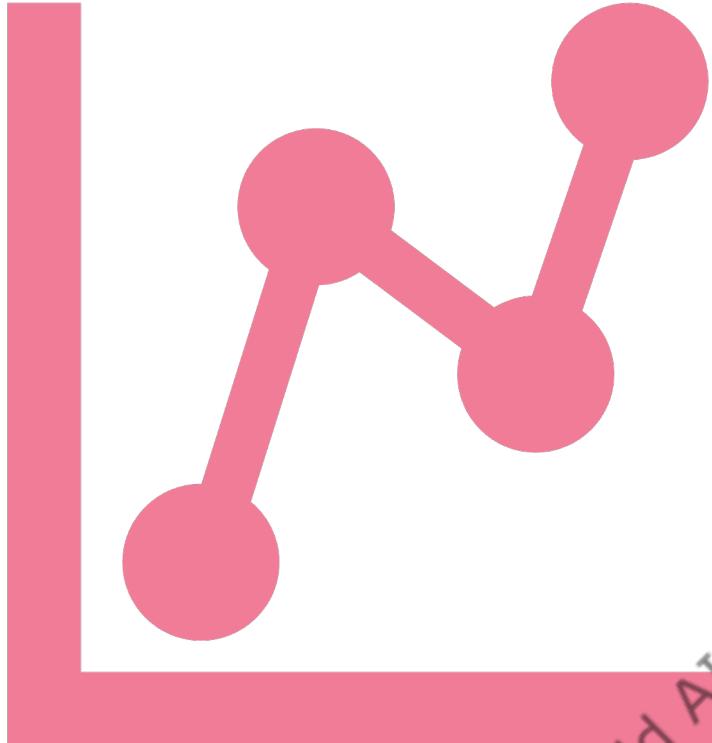
- It's a Phenomenon that skews the result of an algorithm in favor/against an idea.
 - Example: Credit Card Approval Rate is higher for Men, than Women



Manifold AI Learning - Registered Learners Only

Model Explainability

- Understanding the Prediction from Model, will build confidence, and identify the errors



Places Where Change can Occur in ML Application

- ML/Data Pipeline
- Model Inputs & Outputs
- Predictions
- Ground Truth
- KPIs

Ways to Analyze the Change in ML Application



Training – Split the Dataset into Train & Test Part



Serving – Use the Sliding Window to analyze the change in ML Application



Quick Setup

- <https://whylabs.ai/free>

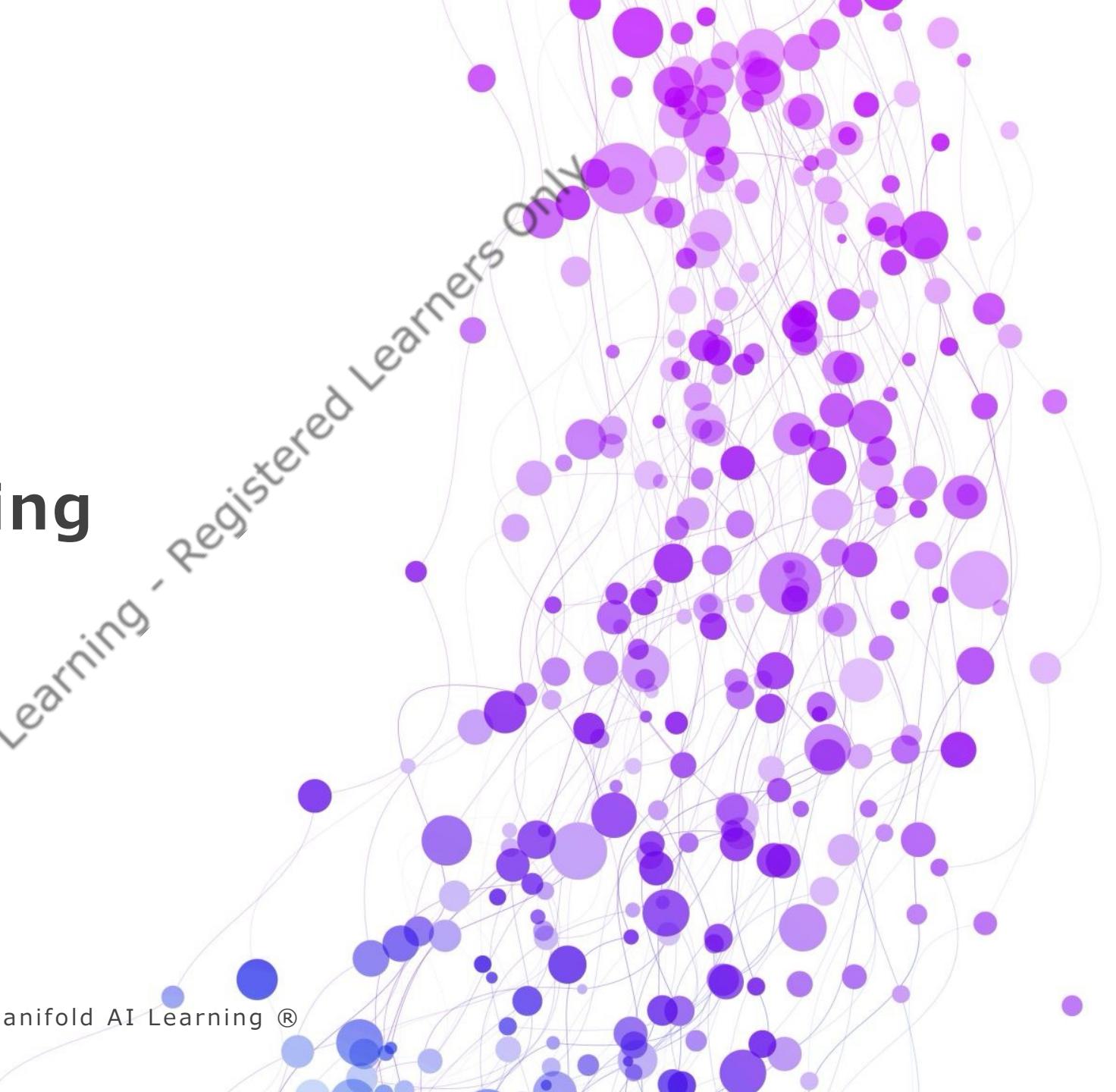
Manifold AI Learning - Registered Learners Only

Post- Productionalizing ML Models

What Next ?

Manifold AI Learning - Registered Learners Only

Manifold AI Learning ®



As part of your Learning Journey:

Packaged Machine Learning Models

Build & Test the ML Package using
various tools

Deploy the Model

Monitor the ML Models

Learnt the Various ways of creating
the ML Application



Important Part:

Getting the Business Value out of the Model.

Security of Machine Learning Models

Manifold AI Learning - Registered Learners Only



Bridging the Gap Between ML Model and Creation of Business Value

- Business users utilize ML model predictions for strategic decision-making.
- High accuracy in ML models does not necessarily equate to positive business impact.
- Bridging the gap between ML solution development and deriving business value is crucial.
- Data scientists must translate model predictions into actionable insights.
- Classification models can be optimized by organizing results into ranked probability-based buckets.
- After deploying an ML model, ensuring its benefit to business users through MLOps is essential.

Bridging the Gap Between ML Model and Creation of Business Value (Contd.)

- Obtaining feedback from stakeholders helps refine the model's performance in a production environment.
- Adjusting models based on user feedback enhances their utility for business purposes.
- Data scientists should communicate the model's functionality to business users at a high level.
- Building trust in model outputs is important, especially as many users may not be familiar with ML terminology.
- Providing output in a readable form, such as through dashboards or chatbots, is preferred by business users and customers.
- Business Intelligence tools like Tableau or Power BI can be employed to create interactive dashboards.

Model Security

Manifold AI Learning ®

Manifold AI Learning Reserved for Trained Learners Only

Model Security

- Model security plays a crucial role in MLOps, safeguarding sensitive information during data processing.
- Attacks can occur at different stages, including model training and production.



Model Security

- Terms related to model security include:
- **Poisoning**: Introducing malicious data during training to alter the model output.
- **Extraction**: Building a new model mirroring the targeted model's functionality.
- **Evasion**: Trying to manipulate the label to a specific class through minor input variations.
- **Inference**: Determining if a particular dataset was part of the training data.
- **Inversion**: Extracting information about training data from the trained model by reverse engineering.



Adversarial Attack



Hostile data input generation is a technique employed by attackers.



Attackers deliberately supply malicious data to the model to induce inaccurate predictions.



This intentional introduction of hostile data disrupts the model's prediction accuracy as it adapts to new patterns.



The impact of this attack is significant, as even slight alterations in input data can lead to substantial changes in model predictions.



Despite appearing normal to humans, the malicious input data adversely affects the model's learning process and subsequent predictions.

Categories of Adversarial Attack

Targeted Attacks:

- Objective: Change the label to a specific target.
- Method: Alter the input data source to a predefined target.
- Resource Intensity: Demands more time and effort.

Non-targeted Attacks:

- Objective: Change the label without a specific prediction target.
- Method: Modify the label without a predefined target.
- Resource Intensity: Requires less time and effort.

Methods for Adversarial Attacks

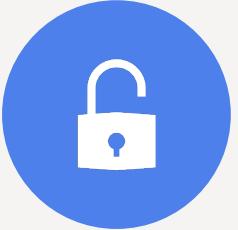
Black-box Method:

- Process: Attackers send input data to the model and receive corresponding output.

White-box Method:

- Information: Attackers possess comprehensive knowledge about the ML model, including training data and feature weights.

Data Poisoning Attack



Access: Attackers gain entry to input or source data.



Method: Introduce noise to the original data, strategically altering the model's predictions.



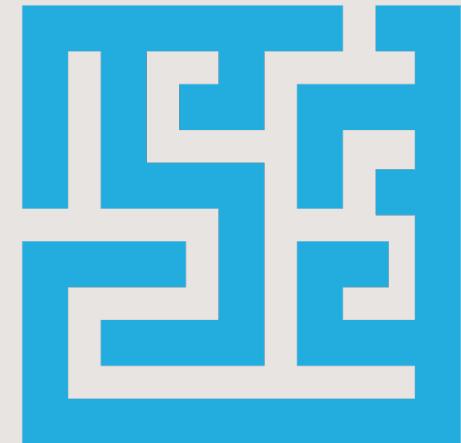
Objective: Modify the input data to induce incorrect and unreliable model predictions.



Focus: Targets the training data, intelligently modifying it for deceptive outcomes.

Distributed Denial of Service Attack (DDoS)

- **Resource-Intensive Attack:**
- **Definition:** Involves supplying intricate data to models, prolonging prediction times.
- **Implication:** Limits the utility of models for users.
- **Method:** Attackers may introduce malware to manipulate the system or server for disruptive effects.



Data Privacy Attack

- **Data Privacy Definition:**
- **Scope:** Encompasses the confidentiality of Personally Identifiable Information (PII) and Personal Health Information (PHI).
- **Target:** Attackers seek to extract sensitive information, including details about the model or its training data.
- **ML Model Vulnerability:**
- Example: Support Vector Machines (SVMs) may inadvertently leak information, as support vectors are derived from training data.
- **Categories of Data Privacy Attacks:**
- Membership Inference Attack
- Input Inference Attack
- Model Extraction Attack

How to Mitigate the Risk of Model Attacks



ML Pipeline

ML Pipeline Phases:

Division: The ML pipeline consists of two primary phases—training and test.

Addressing Model Attacks:

Strategy: Various ML model attacks can be tackled by focusing on these distinct training and test phases.

Manifold AI Learning - Registered Learners Only



Training Phase

- **Data Scientist Activities:**
- **Tasks:** Data scientists engage in activities such as data gathering, data cleansing, feature engineering, algorithm selection, hyperparameter tuning, and model building.
- **Training Phase Vulnerability:**
- **Risk:** Attackers exploit this phase through data poisoning, compromising the reliability of model predictions.
- **Mitigation Techniques:**
- **Strategies:** Implement the following techniques to mitigate training phase attacks:
 - Data encryption
 - Safeguarding the integrity of training data
 - Employing robust statistics
 - Data sanitization

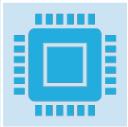
Test Phase

- **Test Phase Target:**
- **Focus:** Attackers direct their efforts towards ML models in this phase.
- **Common Attack:** Model extraction attacks are frequently employed, aiming to ascertain if input X belongs to the training data or to pilfer model parameters established during training.
- **Risk Mitigation Techniques:**
- **Strategies:** Implement the following techniques to reduce risk during the test phase:
 - Adversarial training
 - Utilize autoencoders
 - Apply distillation methods
 - Employ ensemble techniques
 - Restrict the number of requests per user

Tools for Model Security



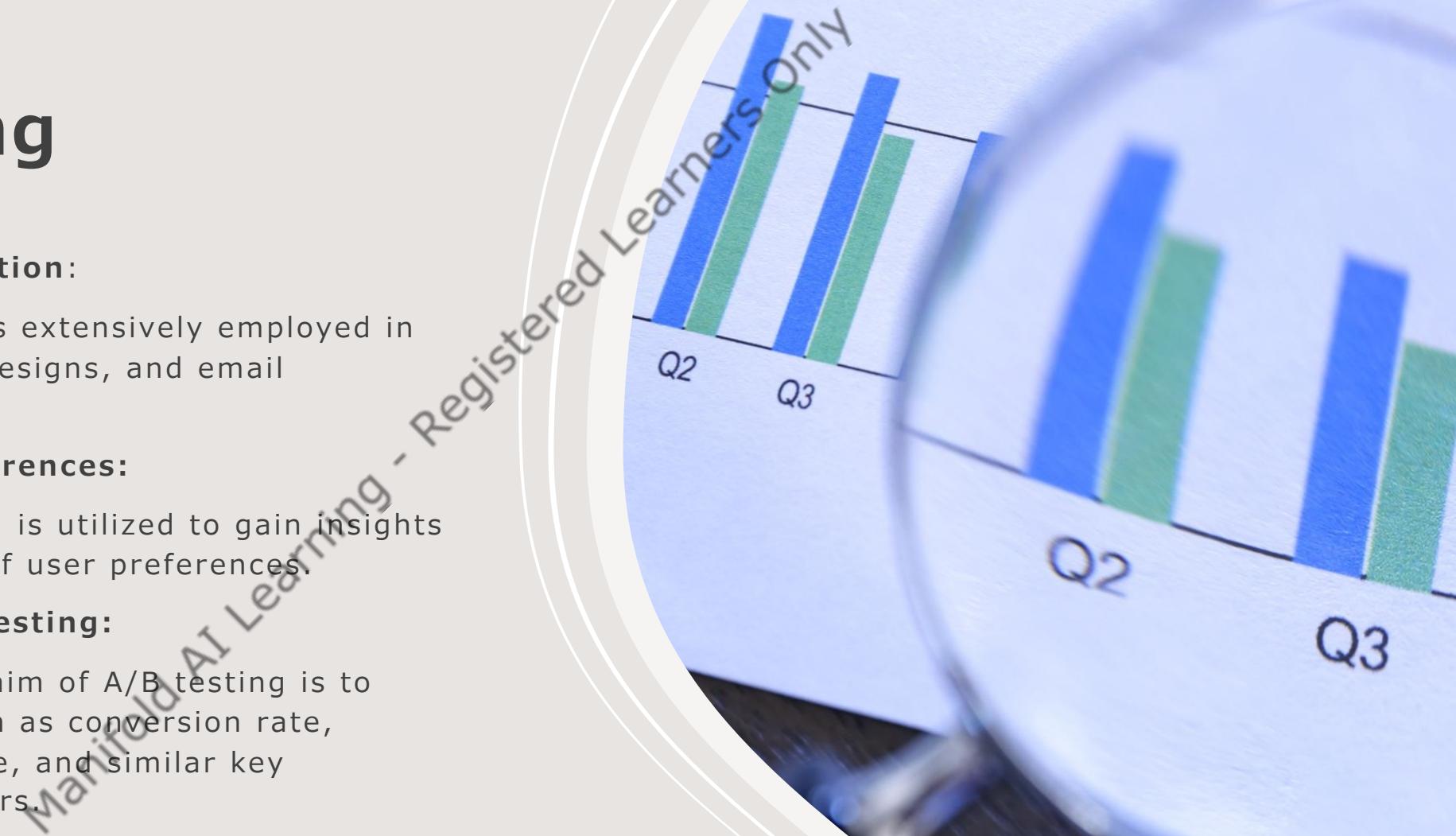
Resources: Use the Python library Adversarial Robustness Toolbox (ART) and the command-line tool Counterfit to enhance ML model security.



Features: These tools provide defense against common ML attacks and offer support for various ML frameworks, libraries, and data types.

A/B Testing

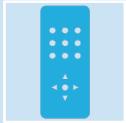
- **Widespread Application:**
- **Usage:** A/B testing is extensively employed in marketing, website designs, and email campaigns.
- **Learning User Preferences:**
- **Purpose:** A/B testing is utilized to gain insights and comprehension of user preferences.
- **Objectives of A/B Testing:**
- **Goals:** The primary aim of A/B testing is to enhance metrics such as conversion rate, success rate, revenue, and similar key performance indicators.



Step 1: A/B Testing Process



A/B testing divides audiences into equal sets.



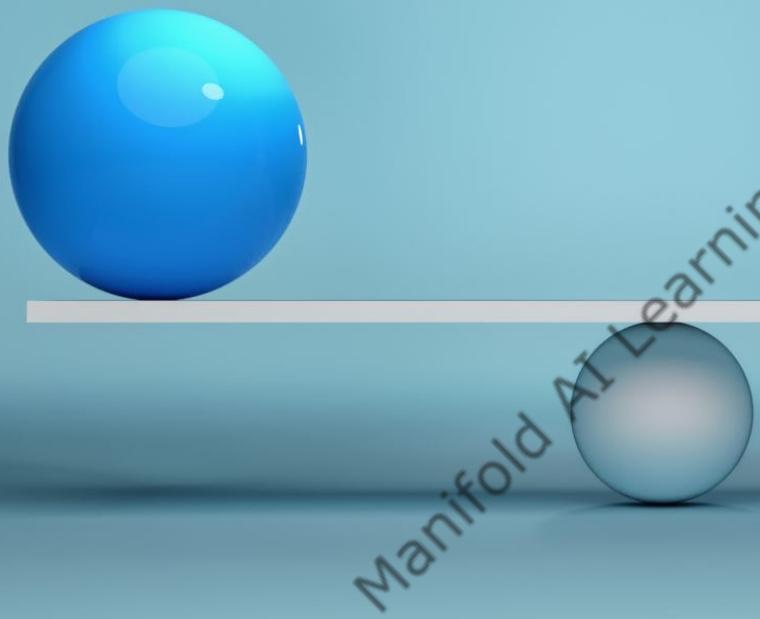
Control version: Current state.



Experimental version: New or challenger iteration.

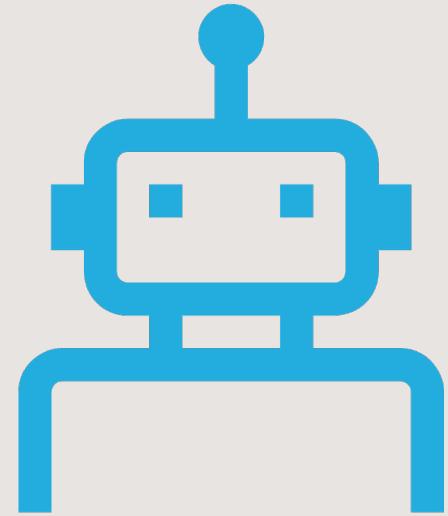
Step 2: Preparation Steps

- Define problem statement for A/B test.
- Formulate null and alternative hypotheses.



Step 3: Experiment Design

Structure the experiment to track and analyze metrics.



Manifold AI Learning - Registered Learners Only



Step 4: Execution and Validation

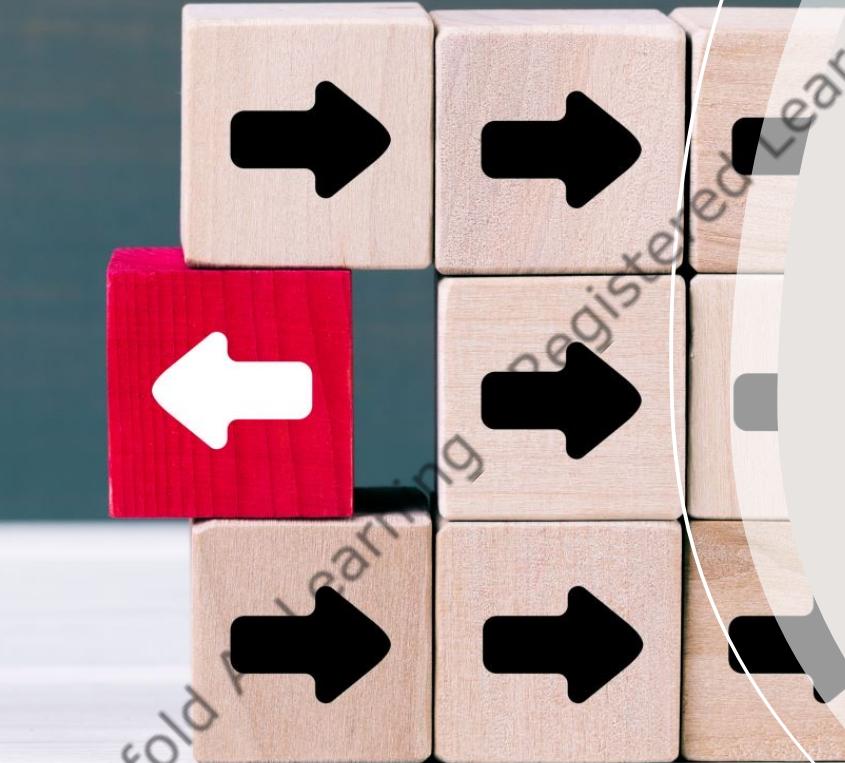
Run and validate the experiment for data collection.

Step 5: Statistical Analysis

Compare statistics
derived from the
experiment's output.

Step 6: Decision Making

- Make informed decisions based on results.



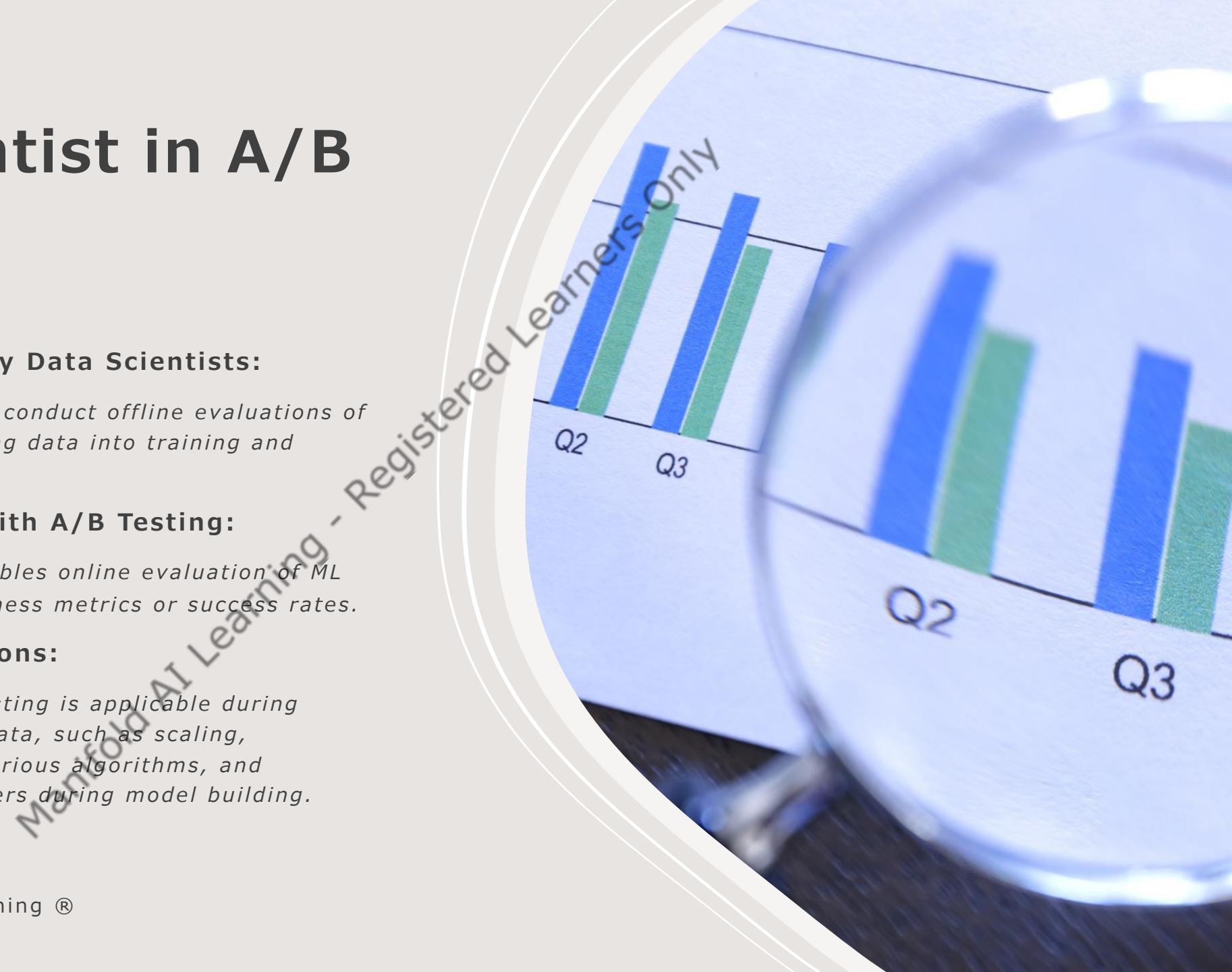
Step 7: Reliability Warning

- Highlight the importance of correct A/B test execution.
- Emphasize that incorrect execution may yield unreliable results.

Manifold AI Learning ®
- Registered Learners Only

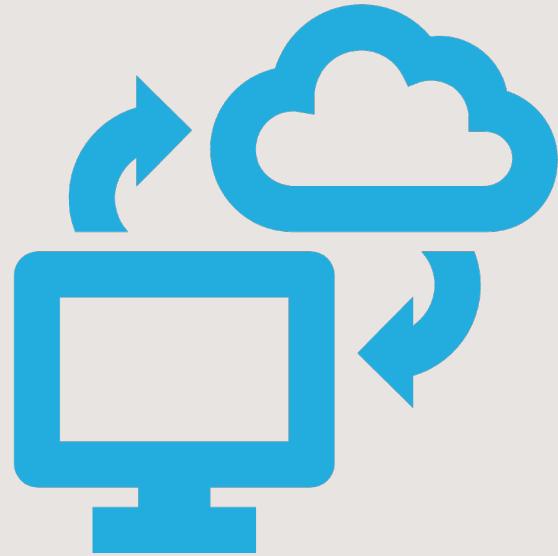
Data Scientist in A/B Testing

- **Offline Evaluation by Data Scientists:**
- *Process: Data scientists conduct offline evaluations of ML models by partitioning data into training and validation sets.*
- **Online Evaluation with A/B Testing:**
- *Method: A/B testing enables online evaluation of ML models, measuring business metrics or success rates.*
- **A/B Testing Operations:**
- *Implementation: A/B testing is applicable during operations on training data, such as scaling, normalizing, applying various algorithms, and adjusting hyperparameters during model building.*



Data Scientist in A/B Testing (Contd.)

- **Simultaneous Model Deployment:**
- *Role: MLOps engineers or data scientists can deploy multiple models concurrently for production testing.*
- **Cloud Deployment Platforms:**
- *Platforms: Google Cloud (GCP) and Amazon SageMaker facilitate the deployment of multiple models within the production environment behind the same endpoint. This enables assessment of model performance from a business perspective.*
- **Multi-Armed Bandit Strategy:**
- *Introduce the concept of Multi-Armed Bandit, a strategy for dynamically allocating resources to models based on their performance.*



Future of MLOps

Manifold AI Learning - Registered Learners Only



Future of MLOps



Booming Machine Learning Field:

Trend: Machine Learning is flourishing, with industries integrating it as a crucial element in business development.



Challenges in ML:

Dynamics: ML systems are addressing numerous challenges, becoming integral to industries.



MLOps Demand Surge:

Reason: MLOps manages ML model complexity and scalability, leading to a growing demand across industries.



MLOps Engineer Shortage:

Issue: Shortage of MLOps engineers exists due to the unique intersection of Machine Learning and software development.

Future of MLOps



Team Setup or Upskilling:

Solution: Companies need dedicated teams for MLOps engineers or can upskill data scientists to perform MLOps tasks.



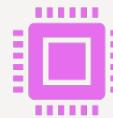
Cost and Effort Reduction:

Impact: MLOps reduces costs and manual efforts in the ML life cycle, enabling focus on more productive tasks.



MLOps vs. DevOps Popularity:

Trend Shift: MLOps is gaining more popularity than DevOps in managing ML models effectively.



Deployment Challenge:

Deployment Reality: Over 85% of ML models do not reach the production environment, limiting their business impact.

Conclusion

Hopefully this course now gave you a complete picture of MLOps and its importance and its implementation

Manifold AI Learning - Registered Learners Only

Congratulations!!

I will see you on the next journey with us!!!

Manifold AI Learning - Registered Learners Only

