AIOps

Instructors:

Sunny Bhaveen Chandra:

Sr. Data Scientist and lecturer at iNeuron.ai with working experience in computer vision, natural language processing and embedded systems. Hands-on experience leveraging machine learning, deep learning, transfer learning models to solve challenging business problems. Also, he has a vast interest in Robotics.

Sourangshu Pal:

Visual Computing Engineer and instructor at iNeuron.ai having 3 years of diverse experience in the discipline of visual computing with specialization in Deep Learning and Computer Graphics. Loves to analyze, process, and model visual data then interpret the insights to create actionable plans for solving challenging business problems.

Curriculum:

AIOps introduction theory

- Introduction 1
- Introduction 2
- Introduction 3
- Challenges
- AIML generic steps
- Level 0 workflow
- Level 0 characteristics and observations
- Level 1 workflow
- Level 1 aim

- Level 1 characteristics
- Frequently used terms
- Data validation
- Model validation - Offline
- Model validation - Online
- Feature store
- Metadata storage
- Pipeline trigger
- Final summary
- Level 2 aim
- Level 2 CI CD workflow detail discussion part 1
- Level 2 CI CD workflow detail discussion part 2
- Level 2 more on CI
- Level 2 more on CD
- Level 2 deployment types
- Level 2 summary final

Linux introduction

- Introduction to Linux
- What is Linux
- Important pieces in Linux
- Features of Linux
- Evolution of Linux
- Differences between Windows and Linux

Setting up our Linux space

- Downloading necessary tools
- Installing Ubuntu in Windows
- What is SSH?
- Install SSH clients
- Setting up SSH in Ubuntu VM
- How to do SSH to your Ubuntu VM?
- Setting up passwordless SSH

Linux concepts

- What is Kernel
- Types of Kernel
- What is Shell
- Types of Shell
- Distros in Linux
- Linux boot process
- File system
- Run levels in Linux
- File types of Linux

Package management

- Package management
- Package managers & DPKG
- Working with APT & APT GET
- Apt-get advanced part - 1

- Apt-get advanced part - 2

Linux commands

- Linux commands part - 1
- Linux commands part - 2
- Linux commands part - 3
- Linux commands part - 4
- Cat command usages

Working with terminal

- File archival
- File compression
- Files and patterns search
- Input output redirection
- Working with VI editor
- Advanced VI editor part - 1
- Advanced VI editor part - 2

Permissions & security

- Types of account in Linux
- User management
- Group management
- Files access controls
- Linux file permissions
- Modifying file ownership

- Sudoers in Linux
- Cronjobs
- SCP
- Special permissions
- System management
- System tools
- Hard link and Soft link
- Aliasing in Linux
- Creating users in multiple ways

Linux in AWS cloud deploy an app in EC2

- Launching an Ubuntu VM and SSH setup
- Package installation in VM
- Running our calculator app
- Gunicorn & Nginx setup
- Creating a Gunicorn service
- Attaching an Elastic IP
- Attaching OpenSSL certificates for https

GitHub introduction

- Git introduction
- What is version control?
- Types of version control
- What is git?
- Why git?

- Git installation on Windows
- Git installation on Linux
- Git setup
- Git terminologies
- Repositories in GIT
- Creating repository
- Checking repository history
- Doing commits
- Git diff
- Git restore
- Gitignore
- Tagging
- Branching
- Branching practicals
- Merging
- Merge conflicts
- Remote repository
- Cloning repository
- Working with remote repository
- Pushing to remote failed in GitHub
- Personal access token setup in Windows
- Personal access token setup in Linux
- Pull request
- Git fetch & pull

- Fork
- Rebasing
- Interactive rebasing
- Git rewrite history
- Git rewrite history continued
- Cherry picking
- Modify recent commits
- Git revert
- Git checkout
- Git reset
- Git stash
- Git reflog
- Course outro

DVC introduction

- What is DVC?
- Installation

Automate ML pipelines with DVC

- Workflow
- Basic setup
- Stage 01 implementation
- Stage 01 added to dvc.yaml
- Stage 02 implementation
- Stage 02 added to dvc.yaml

- Stage 03 implementation
- Stage 03 added to dvc.yaml
- Stage 04 implementation
- Stage 04 added to dvc.yaml
- Final update

Getting started with DVC

- Data versioning 01
- Data versioning 02
- Data versioning 03

Automate DL pipelines with DVC (dl-tensorflow)

- Workflow description
- Creation of project skeleton
- Stage 01 implementation
- Stage 01 added to dvc.yaml
- Stage 01 final update
- Stage 02 main file creation
- Stage 02 base model creation
- Stage 02 python scripting
- Stage 02 logging model summary in the logs
- Stage 02 added to the dvc.yaml file
- Stage 03 preparing directory creation for callbacks
- Stage 03 adding callback utility
- Stage 03 adding to dvc.yaml

- Stage 04 loading binary file of callbacks
- Stage 04 load untrained model and start training
- Stage 04 training valid generator
- Stage 04 model training added
- Stage 04 save model

Docker

- Docker introduction
- What is Docker?
- Why Docker?
- Benefits of Docker
- What is container?
- Containers vs VM
- Containers vs image
- Docker editions
- What docker is not?
- Important terminologies
- Docker setup in Windows
- Docker setup in Linux
- Docker setup in Mac

Docker basic usage

- Docker basic commands part 1
- Docker basic commands part 2

Docker run

- Docker run part 1
- Docker run part 2

Docker images

- Docker images
- Creating a new image
- Environment variables
- Commands & entrypoints

Docker compose

- Docker compose
- Voting application understanding
- Docker compose versions
- Docker compose networks
- Voting application with docker run
- Voting application with docker compose

Docker concepts

- Docker engine
- Docker storage
- Docker networking
- Docker registry

Kubernetes

- Course introduction
- What is Kubernetes?
- Why Kubernetes?
- Containers
- Containers orchestration

Kubernetes setup

- Kubernetes setup on Windows
- Kubernetes setup in Linux
- Kubernetes setup in Mac
- Minikube
- Kubeadm
- Kubernetes architecture

Kubernetes concepts

- Pods
- Node architecture
- Replication controller
- Deployments
- Services
- Labels
- Healthchecks
- Readiness probe
- Pod state
- Pod lifecycle

- Secrets
- Webui

Services

- Nodeport
- Clusterip
- Loadbalancer

Advanced

- Service discovery
- ConfigMap
- Ingress controller
- External DNS
- Volumes
- Volumes autoprovisioning
- Pod presets
- Statefulsets
- Daemonsets
- Resource usage monitoring
- Autoscaling

Deploying apps

- Microservices architecture
- Deploying in Kubernetes
- Deploying in kubernetes with deployments

Packaging & deployment

- Introduction to Helm
- Creating your own Helm charts
- Setting up a Helm repository on S3
- Building and deploying Helm charts with Jenkins

MLFlow introduction

- What is MLFlow?
- Installation
- Where runs are recorded
- How runs and artifacts are recorded
- Scenario 1: MLFlow on localhost
- Scenario 2: MLFlow on localhost with sqlite
- Scenario 3: MLFlow on localhost with tracking server
- Scenario 4: MLFlow with remote tracking server, backend and artifact stores
- Logging data to runs
- Logging functions
- Launching multiple runs in one program
- Performance tracking with metrics
- Visualizing metrics

Automatic logging

- Scikit-Learn
- Tensorflow and Keras

- Gluon
- Xgboost
- Pytorch

MLFlow tracker

- Organizing runs in experiments
- Managing experiments and runs with the tracking service API
- Tracking UI
- Querying runs programmatically
- MLFlow tracking servers
- Storage
- Networking
- Logging to a tracking server

MLFlow projects

- Overview
- Specifying projects
- Running projects
- Iterating quickly
- Building multistep workflows

MLFlow models

- Storage format
- Model signature and input example
- Model API

- Built-In model flavors
- Model customization
- Built-In deployment tools
- Deployment to custom targets

Model registry

- Model registry workflows
- UI workflow
- Registering a model
- Using the model registry
- API workflow
- Adding an MLFlow model to the model registry
- Fetching an MLFlow model from the model registry
- Serving an MLFlow model from model registry
- Adding or updating an MLFlow model descriptions
- Renaming an MLFlow model
- Transitioning an MLFlow model’s stage
- Listing and searching MLFlow models
- Archiving an MLFlow model
- Deleting mlflow models

Mlflow integration with project

- Mlflow integration with project

Kubeflow foundation

- What is Kubeflow?
- Core Kubeflow components
- How to set up Kubeflow on Kubernetes
- How to develop basic ML models in Kubeflow notebooks
- How to train and deploy models in kubeflow
- How to use Kubeflow pipelines
- How to use kfserving to deploy models
- How to manage logs with Kubeflow metadata component
- Katib hyper parameter tuning
- Kubeflow pipelines to kfserving

AWS MLOps

- Amazon sagemaker
- Amazon s3
- AWS codebuild
- AWS codecommit
- Sagemaker training job
- Sage maker endpoint
- Amazon api gateway
- Sagemake model monitoring
- Cloudwatch synthetics
- Cloudwatch alarm

Azure MLOps

- Create an Azure machine learning workspace

- Setup a new project in Azure DevOps
- Import existing YAML pipeline to Azure DevOps
- Declare variables for CI/CD pipeline
- Create training compute
- Train ML model
- Register model
- Deploy model in AKS

GCP MLOps

- Creating Flask application using Python
- Best practices building Flask app
- Understanding docker files and dependencies
- Creating container image
- Walkthrough of different deployment options
- Serverless deep dive
- Deploying on GCP app engine
- Deploying on serverless framework
- Hosted Kubeflow pipelines
- Start hosted pipelines
- Cluster permissions
- Development environment
- Launch AI platform notebook
- CI/CD production environment
- Set up continous integration (CI)
- Verify CD

Digital ocean

- Droplets
- File transfers
- Gitops
- Jenkins
- Creating jobs
- Creating pipelines in jenkins
- Docker images
- Kubernetes flow
- Creating clusters
- Load testing