# 'Specialization - 04'

## # Week - 01

### ⓐ Model Serving

- we want to minimum latency and amaximize throughput.

- Low latency databases. → use caching and Feature Lookup
  - NOSQL
    - gc Memorystore
    - gc Firestore
    - gc Bigtable
    - Amazon DynamoDB

Tools
- clipper
- Tensorflow serving

## # Week - 02

### ⓐ Model Serving Architecture

- Model Servers : Tensorflow Serving, Torch Serve, KF Serve (Kubeflow)
  NVIDIA TRITON Inference server.

### ⓐ Scaling Infrastructure

- Horizontal and vertical Scaling
- Container Orchestration ⟺ Kubernetes, Docker Swarm

A open-source system for automating, deployment, Scaling, management system for containerize application

- ML workflows on Kubernetes – KubeFlow
  - KubeFlow
    ↳ ML workflows on Kubernetes simple portable and scalable.

- Online Inference

- Data preprocessing and Inference.
  ↳ Apache Beam, Tensorflow transform.

  → Suppose, you deploy a model on the server, to make prediction the model needs input. But user doesn't specify what kind of input the models needs. Here, Apache Beam helps to preprocess the input the model requires.

- Batch Inference.
  - Maximize throughput

- Batch procesing with ETL and stream processing.

# Week 03

- Experiment Tracking
  - Tools for managing notebook code
    - nbconvert - Nbdime . jupytext
    - neptune-notebooks

→ Tools for experiments

- Data versioning – Neptune, pachyderm, Data Lake
  Git LFS, Dolt, Lake FS,
  DVC, ML-Metadata
- Tensorboard.

→ MLOPS

- ML Lifecycle Managment
- Model versioning
- Model monitoring
- Model governance
- Model security
- Model Discovery.

– MLOps Methodology.

- Building model from scratch is called Level-0
  Manually
- MLOPS Level 0 → Manually.
- MLOPS Level 1 & 2
  - introduce pipeline, automation
  - Data validation
  - Model validation
  - Feature store
  - Metadata store

- Managing Model versions.
  - Need to track code, Data, config
  - Major : Minor : pipeline
  - Metadata stored by model registry.
- Continuous Delivery (CD)
  - Deploys new code and trained models to the target environment
  - Ensures compatibility of code and models with target environment.
  - Checks the prediction service performance of the model before deploying
- Continuous Integration
  - Targeted when new code is pushed
  - Build packages, containers, images
  - Deliver the final packages to CD pipeline
- A/B Testing (you have to at least 2 business model)
  - Users are divided into two groups
  -
- Multi-Armed Bandit (MAB)
- Contextual Bandit

# Week 04

- Model Monitoring and Logging
  - observability measures how well the internal states of a system can be inferred by knowing the inputs and outputs. (TFMA)
    - Alertable
    - Actionable
  - Logging for ML Monitoring
  - Tracing for ML system - Monolithic System
    - Help troubleshoot
  - Mitigating model decay
    - Divergence : KL Divergence, JS Divergence, K-S Test

- Responsible AI
  - GDPR
  - PATE
  - CCPA
  - Anonymization
  - Pseudonymisation