Materials
MLE Best practices

Homework

During the course you will be using Linux, thus if your OS is not Linux, install VirtualBox and use any Linux distribution (Ubuntu is the best choice for a Linux newbie).

1. Download any Linux distributive, use Ubuntu by default https://ubuntu.com/desktop
2. How to install Linux on VirtualBox, watch up to 9th minute
3. Install Docker and Docker Compose on Linux. How to run docker without sudo
4. Create a repository on your github.

A repository should contain separate folders for each further task. For each homework in addition to the code, you should provide a README file with the commands to perform the task and your description/comments.

*Useful materials & links for the course (optional, for further self-study of topics)**:**

Topic	Links
Common	Useful git repository
Cloud solutions	Google cloud Cloud ML Engine AWS SageMaker Azure Machine Learning studio
Platforms	Use cases of existing architectures, ● Uber ● Netflix ● Airbnb Ready to use solutions ● Domino ● Datalab ● MLFlow ● MetaFlow (AWS only) EPAM accelerators ● Legion Kubernetes-based: ● Kubeflow ● Polyaxon
ML development/research process	Defining iterations: https://blog.insightdatascience.com/how-to-deliver-on-machine-learning-projects-c8d82ce642b0 Caveats: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf Workflow managers ● Luigi

	● Airflow ● Dagster Dividing batch data flows from streaming data flows Version control / model versions / collaboration tools ● DVC Experiment management: ● TRAINS (separate client-server, good-looking UI, better configuration then in MLFlow)
Optimizations	Scaling: Spark, distributed training/inference for Deep Learning ● Horovod ● Ray Low-level: optimizing frameworks for inference, low-level memory manipulations/parallelism to optimize models training/inference ● Treelite
Domain specific knowledge	Storage solutions for large amount of dense matrices ● TileDB Text processing: metadata governance, layers (raw -> cleaned -> tokenized -> etc.) CV: storage and basic manipulations (up/down sampling, cleaning, augmentations) ● PicPoc
AUX	Configuration management ● Hydra: configuration management/cli tool (github) Containers REST API deployment option Monitoring pipelines: visualizations, reproducibility Tooling/reporting: ● Streamlit: Fast reporting with advanced data caching and interactive elements