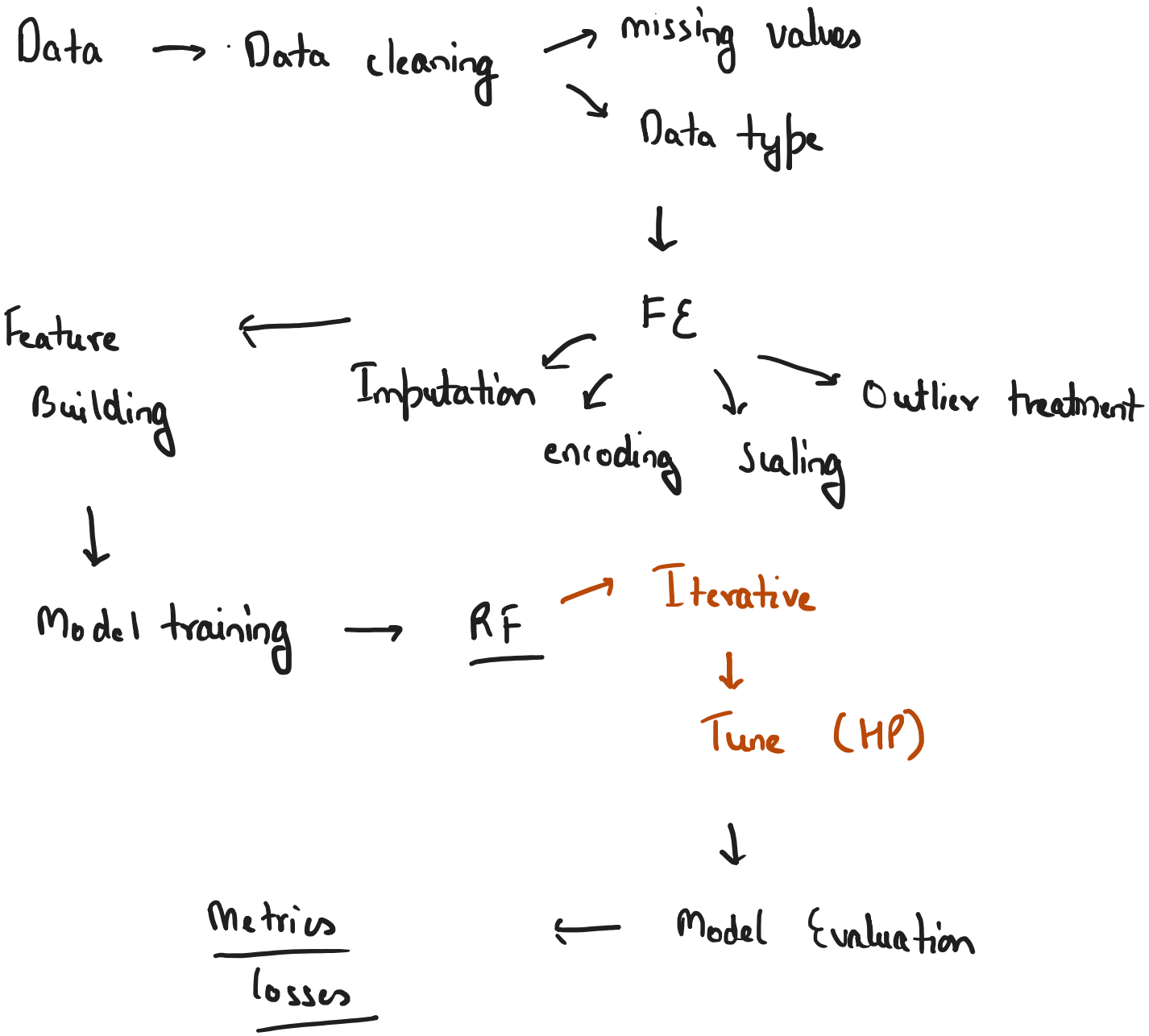
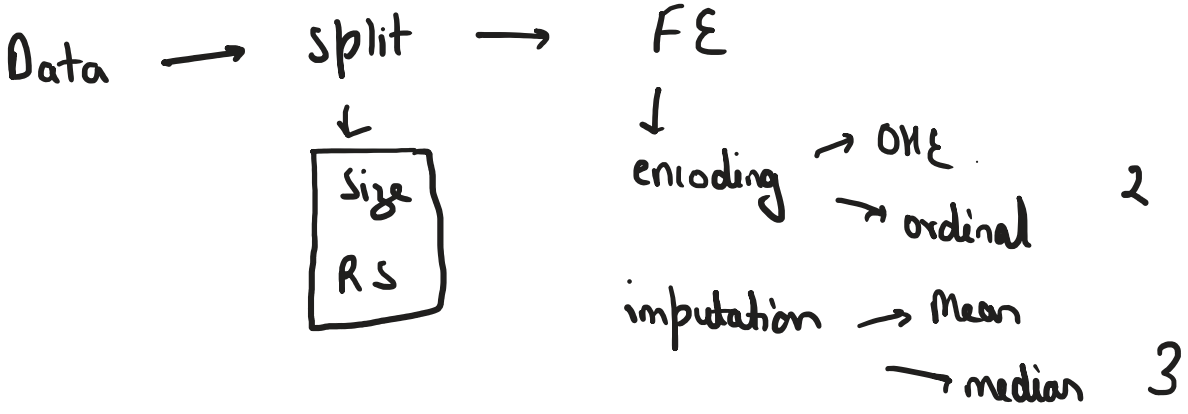

Data
Data cleaning
missing values
Data type
FE
Imputation
encoding
Scaling
Outlier treatment
Feature Building
Model training
RF
Iterative
Tune (HP)
Model Evaluation
Metrics
losses


Data
split
Size
RS
FE
encoding
OHE
ordinal
2
imputation
Mean
median
3

12

↘ constant

Scaling → Min Max
Standard

2

What is Experiment Tracking

08 July 2024 19:40

Experiment Tracking in MLOps

Experiment tracking is the process of recording all the details of machine learning experiments. This includes configurations, code versions, datasets, metrics, and results.

Why It's Used in MLOps:

1. **Reproducibility**:
   - Ensures experiments can be repeated with the same settings, helping to verify results.

2. **Comparison**:
   - Makes it easy to compare different models and experiments to find the best performing one.

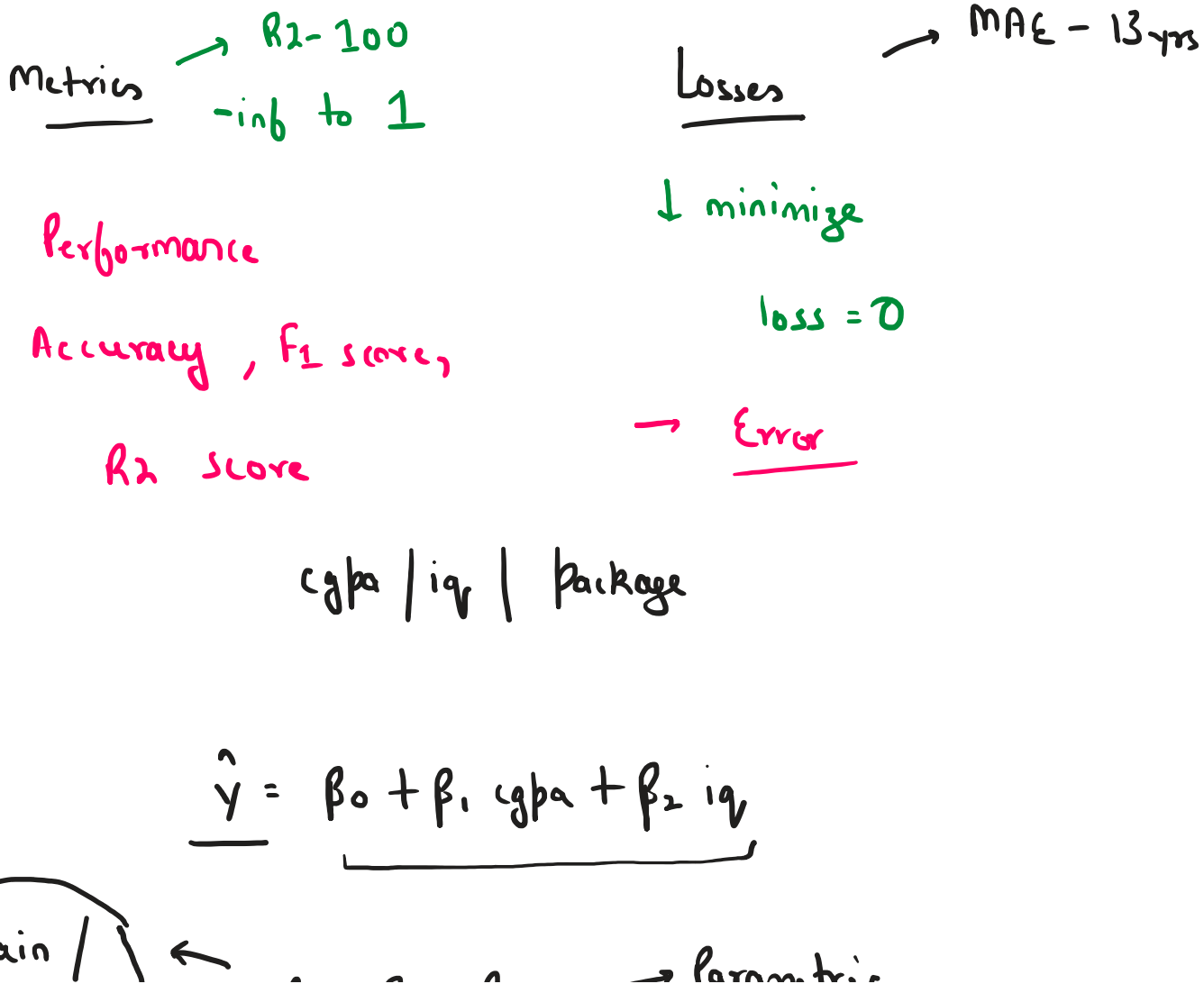
3. **Collaboration**:
   - Allows team members to share and review each other's work, enhancing teamwork.

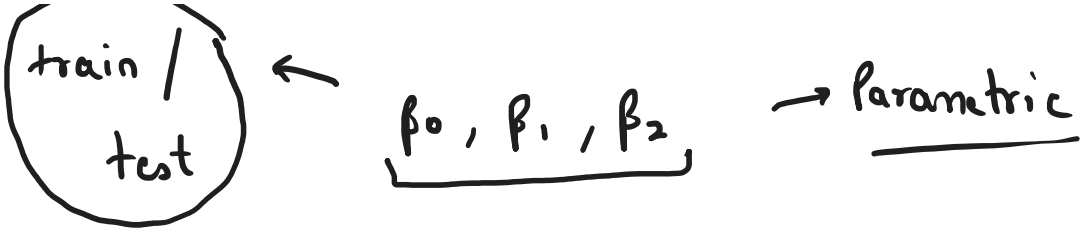
4. **Efficiency**:
   - Saves time by avoiding repeated work and helps in quickly finding the best model settings.

5. **Auditability**:
   - Keeps a history of all experiments, useful for tracking progress and compliance purposes.

Best model

LR





Cloud → EC2

MLFlow → address

Where does MLFlow fit

08 July 2024 19:42

Where does MLFlow fit:

1. **Experimentation**:
   - Tracking: MLflow helps log parameters, metrics, and artifacts of each experiment. This ensures that all details are recorded and can be compared later.

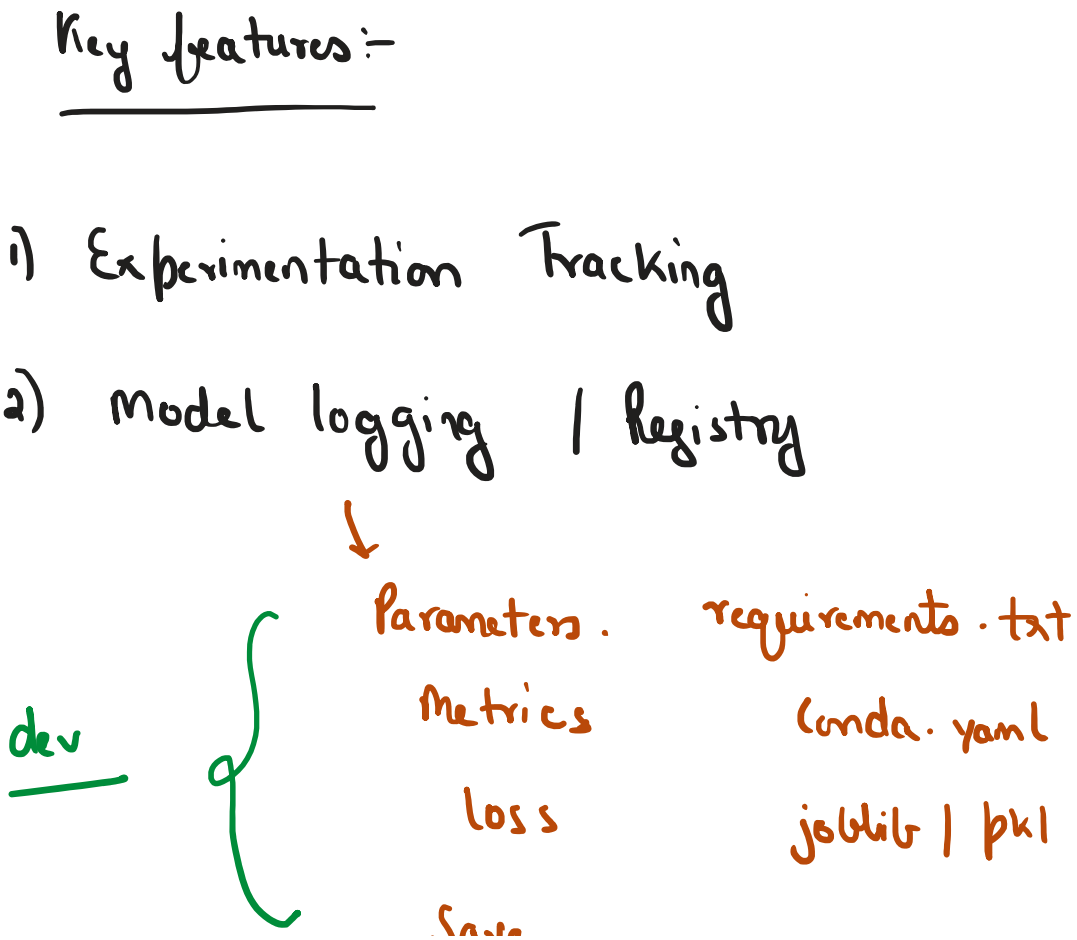
2. **Model Development**:
   - Projects : Standardizes the way to package and share machine learning code. MLflow Projects can be used to run experiments in a consistent environment.

3. **Model Validation**:
   - Tracking: Continues to log validation metrics and results, making it easier to evaluate model performance.

4. **Deployment**:
   - Models: MLflow allows you to register, version, and deploy models with ease. Models can be served directly via APIs or integrated into existing systems.

5. **Monitoring**:
   - Tracking: Helps monitor deployed models by logging predictions and performance metrics, ensuring the model remains effective over time.

6. **Lifecycle Management**:
   - Registry: Manages the full lifecycle of machine learning models, from development to deployment to retirement.



Save

Model Registry → LR → Parametrics

↓ linear Regression V.1

Deployment

linear Regression V2.

Packaging → Docker

Model Evaluation → Regression , Classification

↓ ↓

evaluate UI

Confusion matrix

AUC ROC ,

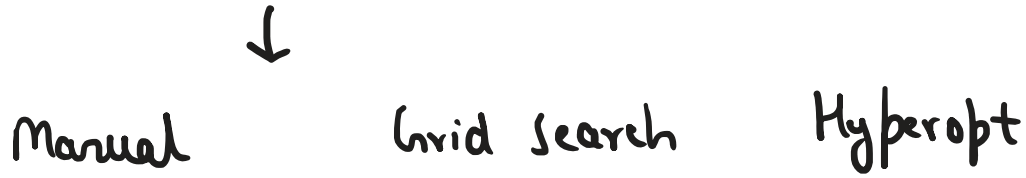
Model deployment → Registry → Best models

v3 ↓

Docker

v2

Hyperparameter tuning → Multiple experiments


Manual
Grid search
Hyperopt

MLFlow Unique Features

08 July 2024 19:46

1. **End-to-End ML Lifecycle Management**:
   - MLflow covers the entire ML lifecycle from experimentation to deployment and monitoring in one platform.

2. **Modular Design**:
   - **Tracking**: Logs and queries experiments, including code, data, configurations, and results.
   - **Project**: Standardizes the way to package and share ML code.
   - **Models**: Manages model packaging and deployment across various environments.
   - **Registry**: Facilitates model versioning, staging, and deployment.

3. **Flexible Deployment Options**:
   - Supports multiple deployment targets including REST APIs, cloud services, and on-premise systems.
   - Provides tools for deploying models to diverse platforms like Azure ML, Amazon SageMaker, and more.

4. **Interoperability**:
   - Works well with various ML libraries and tools, such as TensorFlow, PyTorch, Scikit-learn, and XGBoost.
   - Supports multiple programming languages including Python, R, and Java.

5. **Open Source and Extensible**:
   - As an open-source project, it encourages community contributions and extensions.
   - Easily customizable to fit specific organizational needs.

6. **Comprehensive Model Registry**:
   - Tracks model versions, stages (e.g., staging, production), and metadata.
   - Provides a centralized repository for managing and deploying models.

7. **Built-in Experiment Comparison**:
   - Allows for easy comparison of different experiment runs side by side.
   - Facilitates identification of the best performing models with visualizations and metrics tracking.

8. **Seamless Integration**:
   - Integrates with CI/CD pipelines, enabling automated model training, testing, and deployment.
   - Supports popular CI/CD tools like Jenkins, GitLab CI, and GitHub Actions.

9. **User-friendly Interface**:
   - Provides an intuitive web UI for managing experiments, models, and deployments.
   - Enables users to visualize metrics, parameters, and other experiment details easily.

10. **Scalability**:
    - Designed to scale with organizational needs, from small teams to large enterprises.
    - Handles a large number of experiments and models efficiently.