



Community

MLOps Understanding :

What is MLOps?

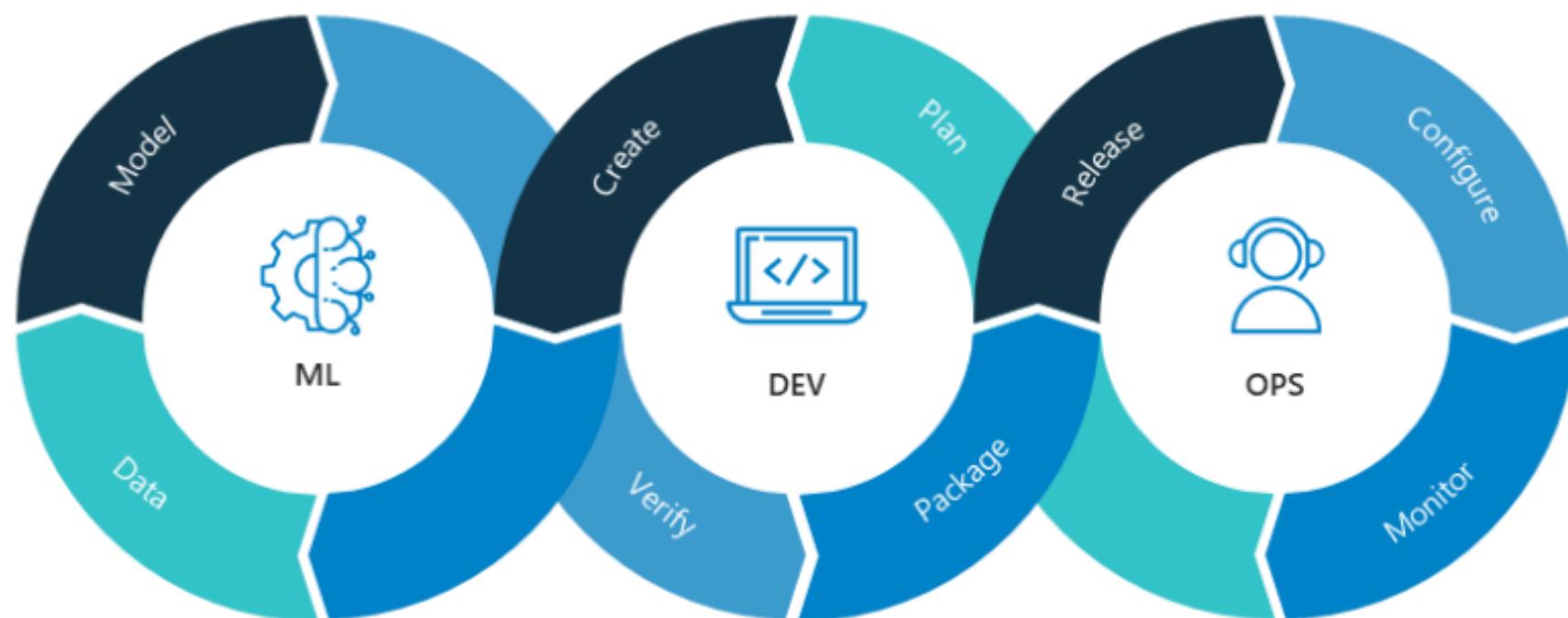




MLOps Understanding

What is MLOPS?

MLOps adalah praktik yang **menggabungkan** tiga disiplin ilmu: Machine Learning, Software Engineering (Dev), dan Data Engineering/Operations (Ops) untuk **mengotomatisasi** seluruh siklus hidup model (dari eksperimen sampai produksi) agar pengiriman model ke pengguna menjadi lebih cepat, stabil, dan terukur.





MLOps Understanding

Machine Learning



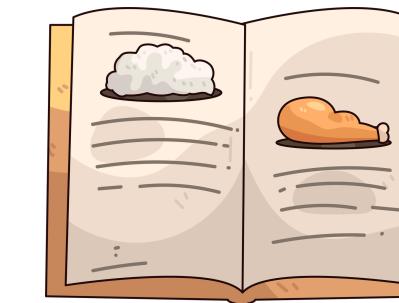
Data Scientist



Data



cooking model



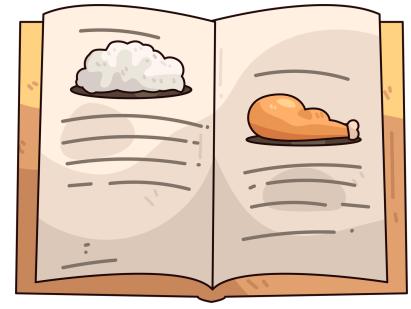
Model

- Data preprocessing
- Feature engineering
- Training model
- Evaluasi model
- Experiment tracking



MLOps Understanding

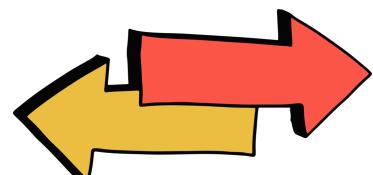
Developers



Model



Flask API



Pipeline



Service

- bikin API model
- versioning code
- testing

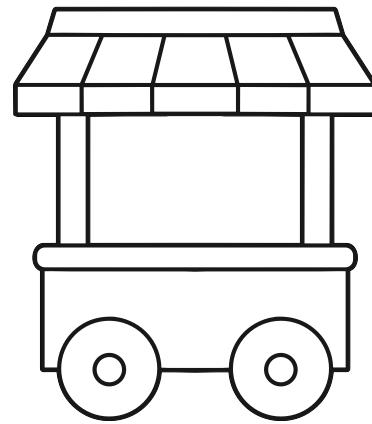


MLOps Understanding

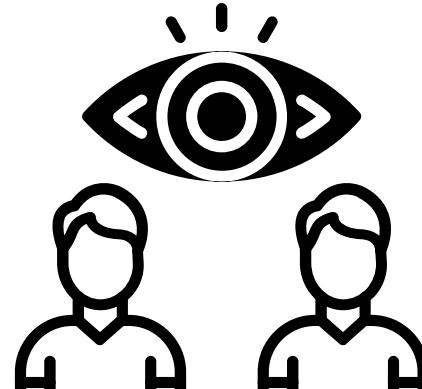
Operations



Menyediakan Data



Docker



Monitor



Menjaga sistem

- deployment infrastructure
- Docker
- monitoring
- automation

MLOps Understanding :

Deploy Machine Learning Model using Flask





Deploy Machine Learning Model using Flask

Data Preprocessing

Data Cleaning

- Handling Missing Values
- Removing Duplicates
- Handling Outliers

Encoding Categorical Variables

- Label Encoding
- One-Hot Encoding
- Binary Encoding

Simplifying Categorical Data

- Grouping Categories
- Reducing Cardinality

Simplifying Categorical Data

- Standardization/Normalization



Deploy Machine Learning Model using Flask

Data Cleaning

Fokusnya adalah membersihkan dataset agar model tidak bingung saat belajar.

- **Handling Missing Values** : Mengatasi data yang kosong (NaN/null). Pilihannya: hapus barisnya jika sedikit, isi dengan nilai rata-rata (mean), atau nilai tengah (median).
1. Hapus Baris (Drop Data): Menghapus seluruh baris yang ada data kosongnya.
- Kapan dipakai:
 - Jika jumlah data yang kosong sangat sedikit (misal di bawah 5% dari total data).
 - Jika baris tersebut adalah Target/Label. Kita nggak bisa menebak jawaban dari soal yang nggak ada kuncinya.
 - Resiko: Jangan pakai ini kalau datanya sudah sedikit.



Deploy Machine Learning Model using Flask

Data Cleaning

2. Isi dengan Nilai Rata-rata (Mean): Mengambil total semua angka lalu dibagi jumlah data untuk mengisi yang kosong.
 - Kapan dipakai:
 - Jika data berjenis Numerik (angka).
 - Jika persebaran datanya Normal/Simetris (tidak ada angka yang terlalu ekstrem/aneh).
 - Resiko: Sangat bahaya kalau ada Outlier (data pencilan).
3. Isi dengan Nilai Tengah (Median): Mengurutkan data dari terkecil ke terbesar, lalu ambil nilai yang paling tengah.
 - Kapan dipakai:
 - Jika data berjenis Numerik (angka).
 - Jika data memiliki Outlier atau Skewed (miring/tidak rata).
 - Arahan: Ini pilihan paling aman (robust) kalau nggak yakin datanya bersih atau nggak.



Deploy Machine Learning Model using Flask

Data Cleaning

- **Removing Duplicates** : Membuang baris data yang sama persis dalam satu dataset.
 - Kapan dipakai: Dipakai jika ada input data ganda yang tidak sengaja terekam, agar model tidak memberikan bobot berlebih pada informasi yang sama.
- **Handling Outliers** : Mendeteksi data yang "aneh" atau terlalu jauh dari rata-rata dan memutuskan untuk dibuang atau diperbaiki.
 1. Hapus Baris (Drop Data)
 2. Imputasi (Imputation) : Ganti dengan mean atau median
 3. Winsorization (Capping) : Mengganti nilai ekstrem dengan nilai batas atas atau bawah yang ditentukan (misalnya persentil ke-5 atau ke-95).
- Kapan dipakai: Sangat efektif jika kita ingin mengurangi dampak data ekstrem tanpa membuang informasinya sepenuhnya. Data yang terlalu tinggi "dimentokkan" ke batas atas, yang terlalu rendah ke batas bawah.



Deploy Machine Learning Model using Flask

Simplifying Categorical Data

Tujuannya adalah menyederhanakan variasi data agar proses komputasi lebih efisien.

- **Grouping Categories** : Menggabungkan kategori yang punya arti mirip atau jumlahnya terlalu sedikit menjadi satu grup besar (misal: "SD", "SMP", "SMA" digabung jadi "Pendidikan Dasar/Menengah").
 - Kapan dipakai: Saat ada terlalu banyak label yang serupa atau label dengan jumlah data sangat sedikit yang tidak signifikan untuk dipelajari model.
- **Reducing Cardinality** : Mengurangi jumlah label unik yang terlalu banyak (misal: dari 100 jenis pekerjaan, diciutkan menjadi 5 kategori industri utama saja).
 - Kapan dipakai: Digunakan untuk mencegah model menjadi terlalu kompleks dan menghindari beban memori yang besar saat proses encoding nanti.



Deploy Machine Learning Model using Flask

Encoding Categorical Variables

Ini adalah tahap menerjemahkan bahasa manusia ke dalam bahasa angka agar mesin bisa melakukan perhitungan matematika.

- **Label Encoding** : Mengubah setiap kategori menjadi urutan angka (misal: Rendah = 0, Sedang = 1, Tinggi = 2).
 - Kapan dipakai: Sangat cocok untuk data yang memiliki tingkatan atau urutan (Ordinal).
- **One-Hot Encoding** : Membuat kolom baru untuk setiap kategori yang berisi angka 1 (jika sesuai) atau 0 (jika tidak).
 - Kapan dipakai: Sangat cocok untuk data yang tidak memiliki tingkatan atau urutan (Nominal).
- **Binary Encoding** : Mengubah kategori menjadi angka, lalu angka tersebut diubah menjadi kode biner (0 dan 1).
 - Kapan dipakai: Solusi cerdas jika kategori sangat banyak (misal: 100+ jenis pekerjaan) agar tabel data tidak menjadi terlalu lebar.

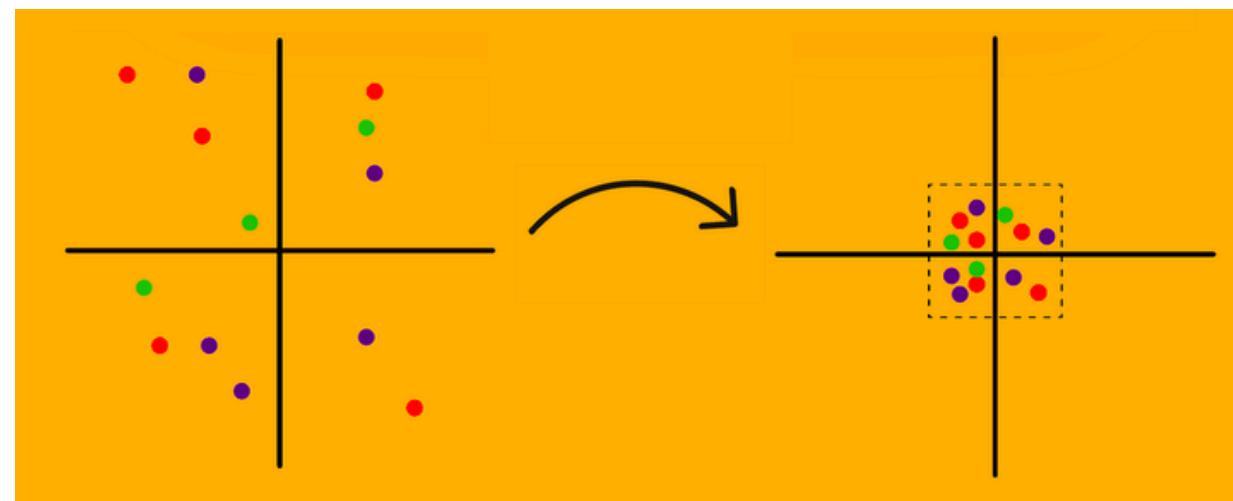


Deploy Machine Learning Model using Flask

Feature Scaling

Sering dilupakan, tapi krusial jika satu kolom punya rentang angka yang beda jauh (misal: Umur 1-100 vs Gaji 1juta-100juta).

- **Standardization/Normalization** : Menyamakan skala semua angka (misal diubah ke rentang 0 sampai 1) agar model adil dalam melihat setiap fitur.
 - Kapan dipakai: Wajib digunakan pada algoritma yang sensitif terhadap jarak atau besaran angka (seperti KNN, SVM, atau Neural Networks) agar kolom dengan angka besar (Gaji) tidak dianggap lebih penting daripada angka kecil (Umur).

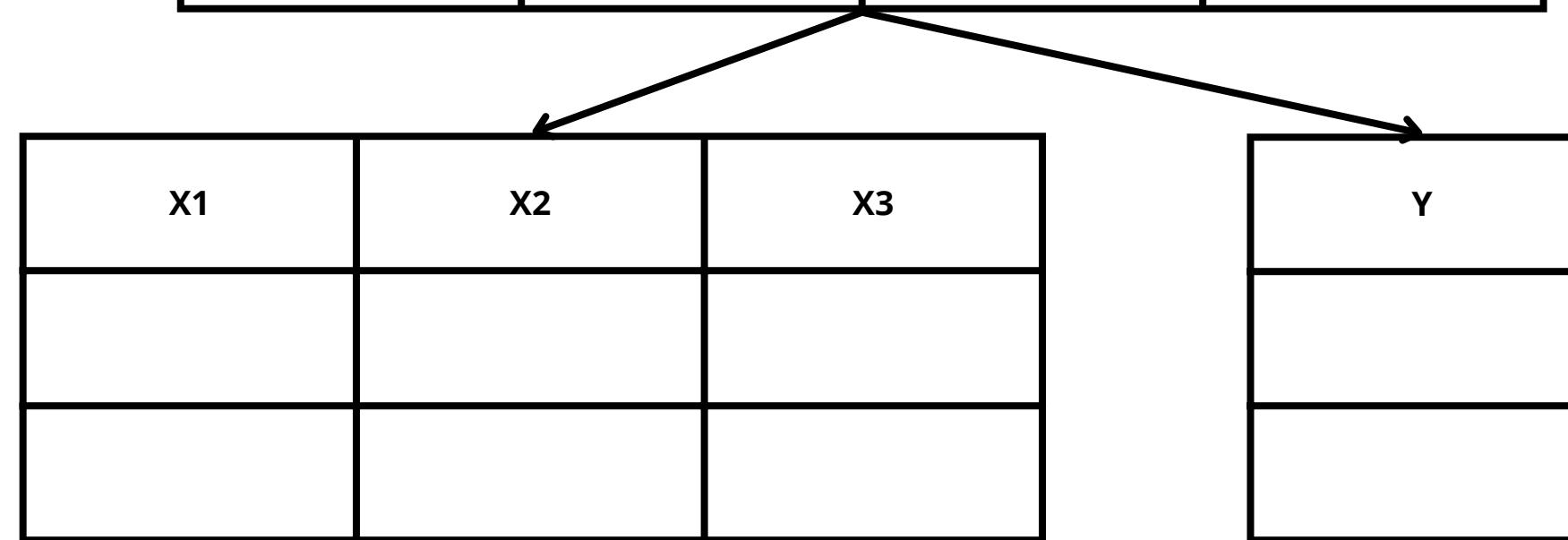




Deploy Machine Learning Model using Flask

Splitting Features and Target

X1	X2	X3	Y





Deploy Machine Learning Model using Flask

Training and Saving Model

Fokusnya adalah bagaimana "mengawetkan" hasil kerja agar bisa digunakan di tempat lain.

- Jangan cuma simpan model.pkl. **wajib** juga untuk simpan juga scaler.pkl (jika ada scaling) dan encoder.pkl (untuk data kategori). Anggap scaler dan encoder sebagai kamus untuk membaca data mentah
- Versi library **harus Sama** pas dilatih pakai dengan flask



Deploy Machine Learning Model using Flask

Aplikasi

Fokusnya adalah membuat "jembatan" yang kokoh antara User dan Model.

- Logika preprocessing data di Flask **HARUS SAMA PERSIS** dengan saat training. Kalau pas training data kosong diisi Median, di Flask juga harus pakai Median yang sama.
- Error Handling - Jika model butuh angka, tapi User kirim teks, Flask harus menolak sebelum data itu sampai ke model dan bikin sistem crash.
- Flask tidak boleh menyimpan memori jangka panjang. Dia terima data -> prediksi -> kasih hasil -> selesai. Urusan simpan database itu tugas lain, bukan tugas API prediksi.



Community

Q&A