ML Model Development using Python (Refinery Industry)

Duration 2 days
Prerequisites: Knowledge of Python programming

Day 1

Module 1: Python & Data Basics (for ML)

- Python quick tour: variables, lists/dicts, functions, importing libraries
- Jupyter workflow: cells, markdown, plotting
- pandas essentials: read_csv, filtering, joins/merges, groupby, missing values
  Lab 1 (Refinery—Basic): Load a small CSV of **daily refinery operations**
  - Columns: date, crude_feed_tph, column_temp_C, ambient_temp_C, steam_tph
  - Tasks: handle missing values, create a clean DataFrame, quick summary stats and plots (line/scatter)

Module 2: Problem Framing & Metrics

- Regression vs classification; targets and features
- Train/test split; data leakage basics
- Metrics: MAE, RMSE, R^2 (intuition and when to use)

  Lab 2: Split Day-1 dataset into train/test; compute MAE/RMSE/R^2 baselines (mean predictor)

Module 3: Simple & Multiple Linear Regression

- Linear regression idea; assumptions (at a basic level)
- One-feature vs multi-feature models; coefficient meaning (direction & magnitude)
- Feature scaling when needed; one-hot encoding for simple categories (e.g., **crude_type**)
  Lab 3 :

- **Use Case A — Predict Steam Usage**
  - Target: steam_tph
  - Features: crude_feed_tph, column_temp_C, ambient_temp_C
  - Fit LinearRegression, evaluate on test set, interpret coefficients (e.g., higher feed ⇒ more steam).

- **Stretch:** Add crude_type (3 categories) via one-hot encoding; re-evaluate.

Day 2

Module 4: Model Diagnostics (Basics)

- Residual plots; under/over-fitting intuition
- Simple outlier handling strategies (cap, remove, or log-transform target/feature)
  Lab 4: Make a residual plot for Use Case A; check if any single variable dominates error.

Module 5: Feature Engineering & Polynomial Regression

- Creating simple derived features: ratios, moving averages, interactions

- Polynomial features (degree 2) for gentle non-linearity
  Lab 5 (Refinery):
- **Use Case B — Predict Diesel Output**
    - Dataset with columns: crude_feed_tph, column_temp_C, reflux_ratio, diesel_output_tph
    - Add polynomial/interaction features (e.g., feed×temp, temp²), compare to plain linear model.

Module 6: Regularization & Model Selection

- Why regularize: bias–variance trade-off
- Ridge vs Lasso vs Elastic Net (basic intuition)
- Cross-validation & GridSearchCV; picking hyperparameters simply
  Lab 6:
- Refit Use Case B with **Ridge** and **Lasso**; choose alpha via cross-validation; compare metrics.