WEEK 6 — MACHINE LEARNING ENGINEERING

(Data Science + Feature Engineering + Model Building + Deployment + Monitoring)

WEEK 6 OBJECTIVES
Interns will learn:
Professional ML pipeline architecture
Clean, modular ML system design (Data → Features → Model → Evaluation → Deployment)
Advanced feature engineering + selection techniques
Model optimization + regularization + hyperparameter tuning
Training pipeline orchestration
Model evaluation + explainability
Model deployment + monitoring

This week produces engineers who can work on **production ML systems**.

DAY 1 — DATA PIPELINE + EDA + PROJECT ARCHITECTURE

🔹 **Learning Outcomes:**
ML project architecture
Dataset versioning
Exploratory Data Analysis (EDA)
Professional folder structure
Data preprocessing pipelines

🔹 **Mandatory Folder Structure (No shortcuts)**

```
src/
  data/
    raw/
    processed/
    external/
  notebooks/
  features/
  pipelines/
```

```
models/
training/
evaluation/
deployment/
monitoring/
utils/
config/
logs/
```

🔹 **Topics to Learn**

- Train/validation/test splitting strategy
- Handling missing values (mean/median/interpolate)
- Outlier detection (Z-score / IQR)
- Data scaling (StandardScaler / MinMaxScaler)
- Class imbalance (SMOTE/weights)
- Dataset versioning (DVC / folder hashing)

🔹 **Exercise**

Build a **data loader + EDA pipeline** that:

1. Loads dataset from `/data/raw`
2. Cleans data (missing, duplicates, outliers)
3. Creates `/data/processed/final.csv`
4. Generates EDA report:
    - Correlation matrix
    - Feature distributions
    - Target distribution

    - Missing values heatmap

✔ Data loaded
✔ Cleaned dataset saved
✔ EDA report generated

🔹 **Deliverables**

```
/pipelines/data_pipeline.py
/notebooks/EDA.ipynb
/data/processed/final.csv
DATA-REPORT.md
```

DAY 2 — FEATURE ENGINEERING + FEATURE SELECTION

🔹 **Learning Outcomes:**
Create meaningful features
Encoding strategies
Advanced transformations
Feature selection techniques

🔹 **Topics**

- Categorical encoding (OneHot, Target, Label)
- Numerical feature transformations (log, sqrt, power)
- Date/time feature extraction
- Text vectorization (TF-IDF / embeddings)
- Feature selection:
    - Correlation threshold
    - Mutual information
    - Recursive Feature Elimination (RFE)

🔹 **Exercise**

Build a feature engineering pipeline that:

- Encodes categorical features
- Normalizes numerical features
- Generates:
  - 10+ new features
- Applies feature selection
- Produces `X_train, X_test, y_train, y_test`

✔ Feature pipeline saved
✔ Feature importance plotted

🔹 **Deliverables**

```
/features/build_features.py
/features/feature_selector.py
/features/feature_list.json
FEATURE-ENGINEERING-DOC.md
```

DAY 3 — MODEL BUILDING + ADVANCED TRAINING PIPELINE

🔹 **Learning Outcomes:**
Multi-model training
Pipeline building
Cross-validation
Overfitting control

🔹 **Topics**

- Models:

    - Logistic Regression
    - Random Forest
    - XGBoost / LightGBM
    - Neural Network
- Cross-validation (k-fold)
- Overfitting vs underfitting
- Regularization (L1/L2)
- Model comparison

🔹 **Exercise**

Create a unified training pipeline that:

- Trains 4 models
- Performs 5-fold cross-validation
- Outputs:
    - Accuracy
    - Precision
    - Recall
    - F1 Score
    - ROC-AUC

Save the best model automatically.

✔ Best model selected
✔ Metrics saved
✔ Confusion matrix plotted

🔹 **Deliverables**

```
/training/train.py
/models/best_model.pkl
/evaluation/metrics.json
MODEL-COMPARISON.md
```

DAY 4 — HYPERPARAMETER TUNING + EXPLAINABILITY + ERROR ANALYSIS

🔹 **Learning Outcomes:**
Optimize a model like an ML engineer
Interpret model decisions
Perform deep error analysis

🔹 **Topics**

- GridSearch / RandomSearch / Bayesian tuning (Optuna)
- SHAP values
- Feature importance
- Error clustering
- Bias/variance analysis

🔹 **Exercise**

Implement:

- Hyperparameter tuning with Optuna/GridSearch
- Generate:
    - SHAP summary plot
    - Feature importance chart
    - Error analysis heatmap

✔ Improvement over baseline
✔ Explainability added

🔹 **Deliverables**

```
/training/tuning.py
/evaluation/shap_analysis.py
/tuning/results.json
MODEL-INTERPRETATION.md
```

DAY 5 — MODEL DEPLOYMENT + MONITORING + MLOPS CONCEPTS (CAPSTONE)

🔹 **Learning Outcomes:**
Deploy real ML systems
Monitor drift
Production-ready pipelines

🔹 **Topics**

- Serving model using FastAPI/Flask
- Creating `/predict` endpoint
- Saving & loading with joblib/pickle
- Input schema validation
- Monitoring:
  - Data drift
  - Accuracy decay
- Logging predictions

🔹 **Exercise (Capstone)**

Deploy model as an API:

```
POST /predict
{
  "feature1": value,
  "feature2": value,
  ...
}
```

Add:
✔ Prediction logging
✔ Request ID tracking
✔ Input validation
✔ Versioned model loading
✔ Basic dashboard (Streamlit optional)

Prepare:

- Dockerfile
- requirements.txt
- .env.example
- README.md

🔹 **Deliverables**

```
/deployment/api.py
/deployment/Dockerfile
/monitoring/drift_checker.py
/prediction_logs.csv
DEPLOYMENT-NOTES.md
```

WEEK- 6 COMPLETION REQUIREMENTS

Skill Area	Requirement
Data Engineering	Clean, versioned datasets

Feature Engineering	Automated + selected features
Model Building	Multiple models trained
Optimization	Hyperparameter tuning
Explainability	SHAP + feature importance
Deployment	Working API
Monitoring	Drift detection
Documentation	Full reports

EXPECTED OUTCOME AFTER WEEK- 6

Interns can now:

✔ Design end-to-end ML pipelines
✔ Engineer advanced features
✔ Train, tune and compare models
✔ Explain model behaviour
✔ Deploy ML systems as APIs
✔ Monitor model performance
✔ Work in real-world ML teams
