MLOps Roadmap for Fraud Detection System

1. Problem Understanding

- Define fraud types (transaction fraud, identity fraud, account takeover).
- Identify KPIs: False Positives, False Negatives, Precision/Recall, Latency.
- Understand data sources: transaction logs, user behavior, cards data.

2. Data Pipeline & Engineering

- Ingestion: Kafka / Kinesis / Azure Event Hub
- Storage: S3 / ADLS / GCS
- Processing: Spark / Databricks / AWS Glue
- Feature Store: Feast or Databricks Feature Store
- Tools Needed: Python, PySpark, SQL, Great Expectations for data validation

3. Model Development

- Algorithms: XGBoost, LightGBM, Random Forest, Autoencoders (for anomaly detection)
- Steps:
• Exploratory Data Analysis
• Feature engineering (transaction patterns, velocity features, user profile deviations)
• Model training & tuning
- Tools Needed:
• Scikit-learn, Pandas, NumPy
• MLflow for experiment tracking
• Jupyter / VS Code

4. Model Evaluation

- Metrics: ROC-AUC, PR-AUC, F1, Recall priority (fraud = high recall)
- Drift Monitoring Setup
- Tools: Evidently AI, MLflow Metrics, Grafana dashboards

5. MLOps Pipeline (CI/CD/CT)

- CI: Linting, unit tests (pytest), code quality checks
- CD: Automated deployment to staging/production
- CT: Automated retraining pipelines
- Tools:
• GitHub Actions / GitLab CI / Azure DevOps
• Docker

• Kubernetes or SageMaker / Vertex AI / Azure ML

6. Model Deployment

- Options:

• Real■time API using FastAPI + Docker + Kubernetes

• Batch inference using Airflow / Dagster

- Tools:

• FastAPI

• Kubernetes

• HELM

• Load testing with Locust

7. Monitoring & Alerting

- Monitor:

• Model performance (drift, degraded metrics)

• System health (CPU, memory, response time)

- Tools:

• Prometheus + Grafana

• MLflow model monitoring

• Evidently AI

• Loki for logging

8. Retraining Strategy

- Trigger-based retraining (drift, threshold breaches)

- Scheduled retraining (weekly/monthly)

- Automate using Airflow, Jenkins or Azure Data Factory

9. Infrastructure Setup

- Cloud Provider: AWS / Azure / GCP

- Infrastructure as Code:

• Terraform

- Storage & Security:

• IAM roles

• Secrets Manager / Key Vault

• Network restrictions (VPC/Subnet model)

10. End-to-End Architecture (High-Level)

- Data ingestion → Data validation → Feature store → Model training
- Model registry → CI/CD pipeline → Deployment → Monitoring
- Retraining loop → Redeployment

11. Resources Needed

- Skills:
• Python, SQL, Docker, Kubernetes
• MLflow, FastAPI
• Terraform
• Kafka or EventHub
- Services:
• Cloud storage, compute clusters, monitoring stack
- Libraries:
• Scikit-learn, XGBoost, PyTorch (optional), Pandas
• Great Expectations, Evidently AI

12. Learning Resources

- MLflow: Databricks Free Course
- Feast Feature Store: Official docs
- Kubernetes: Kubernetes.io tutorials
- Fraud ML Papers: IEEE, Kaggle Fraud Datasets
- MLOps Books: “Practical MLOps”, “Designing Machine Learning Systems”