Data Science & Machine Learning Project Roadmap (3 Months)

Month 1 – Fraud Detection System for Financial Transactions

- **Problem Statement:**
  Financial institutions need to detect fraudulent transactions in real-time. Your goal is to build a robust fraud detection model using historical transactional data and machine learning classification techniques.
- **Tech Stack:**
  `Python`, `Pandas`, `NumPy`, `Scikit-learn`, `XGBoost`, `Matplotlib`, `Seaborn`, `Jupyter Notebook`, `SQL`
- **Development Timeline:**
  - **Week 1:**
    - Ingest transactional data (from Kaggle or simulated).
    - Perform EDA, identify patterns in fraud vs. non-fraud cases.
    - Address class imbalance using techniques like SMOTE or undersampling.
  - **Week 2:**
    - Feature engineering: create time-based, amount-based, and behavior-based features.
    - Train models (Logistic Regression, Decision Trees, XGBoost).
    - Evaluate using F1 score, Precision-Recall, AUC-ROC.
  - **Week 3:**
    - Hyperparameter tuning using GridSearchCV.
    - Create a pipeline for real-time input testing.
  - **Week 4:**
    - Final deployment-ready model.
    - Create a dashboard (basic Streamlit) to simulate real-time prediction.
    - Document findings, risks, and improvements.

Month 2 – AI-based Resume Screening System

- **Problem Statement:**
  HR departments struggle to screen thousands of resumes for job roles. This project will develop a model that reads resume text and classifies them into suitable job categories.

- **Tech Stack:**
  `Python`, `NLTK`, `Spacy`, `TF-IDF`, `Scikit-learn`, `FastText / BERT (optional)`, `Streamlit`
- **Development Timeline:**
  - **Week 1:**
    - Gather text-based resume data.
    - Preprocess text (stop words, stemming, tokenization).
    - Apply TF-IDF to convert text to features.
  - **Week 2:**
    - Use classification models like Naive Bayes, SVM, or Logistic Regression.
    - Evaluate precision, recall, confusion matrix for job category classification.
  - **Week 3:**
    - Upgrade model with pre-trained embeddings (FastText or BERT).
    - Fine-tune embeddings on resume data if required.
  - **Week 4:**
    - Create a resume uploader interface using Streamlit.
    - Model reads resume and suggests the best-fit job roles.
    - Create documentation and demo deck for HR demo.

Month 3 – Predictive Maintenance for Manufacturing Equipment

- **Problem Statement:**
  Predicting machine failure before it happens is crucial in manufacturing. This project aims to analyze sensor data from machines to predict which machines are likely to fail soon.
- **Tech Stack:**
  `Python`, `Pandas`, `NumPy`, `Seaborn`, `Matplotlib`, `Scikit-learn`, `Keras/TensorFlow`, `Jupyter Notebook`
- **Development Timeline:**
  - **Week 1:**
    - Load and explore time-series sensor data.
    - Handle missing values, normalize readings, and extract time-based features.
  - **Week 2:**
    - EDA on failure vs. non-failure scenarios.
    - Train classification model (Random Forest, XGBoost, or LSTM for time series).
  - **Week 3:**
    - Use survival analysis or binary classification to predict Remaining Useful Life (RUL).
    - Evaluate model performance using precision and recall.
  - **Week 4:**
    - Deploy model with a web-based interface showing real-time predictions.
    - Visualize with time-to-failure and alert mechanism.