Complete Roadmap: Fraud Detection Project

Week 11 (November 25 - 29) - Proposal & Setup

Goal: Get approved and set up your environment

Tasks:

- ✅ Submit proposal to AI Lab (November 25th)
- Form your team (2-4 members) and assign roles
- Set up GitHub repository for version control
- Install Python environment (Anaconda recommended)
- Install required libraries:

```
pip install pandas numpy scikit-learn matplotlib seaborn xgboost imbalanced-learn streamlit
```

- Download the dataset from Kaggle (Credit Card Fraud Detection)
- Create project folder structure:

```
fraud-detection/
├── data/
├── notebooks/
├── models/
├── src/
├── ui/
└── reports/
```

Deliverable: Approved proposal + Setup complete

Week 12 (December 2 - 6) - Data Exploration & Preprocessing

Goal: Understand your data deeply

Tasks:

1. **Load and Explore Dataset**
   - Load CSV using pandas
   - Check dimensions, data types, missing values

- Understand each feature (Time, Amount, V1-V28)

2. **Exploratory Data Analysis (EDA)**
   - Plot class distribution (fraud vs legitimate)
   - Visualize feature distributions
   - Check correlations between features
   - Analyze transaction amounts for fraud vs normal
   - Create visualizations (histograms, box plots, correlation heatmaps)
3. **Data Preprocessing**
   - Handle missing values (if any)
   - Normalize/Standardize the 'Amount' and 'Time' features
   - Split data: 70% training, 15% validation, 15% testing
   - Save processed data for modeling

Deliverable: Jupyter notebook with EDA + cleaned dataset

Week 12-13 (December 9 - 13) - Handle Class Imbalance & Baseline Model

Goal: Tackle imbalance and build your first model

Tasks:

1. **Address Class Imbalance**
   - Implement SMOTE (Synthetic Minority Oversampling)
   - Try Random Undersampling
   - Compare different sampling strategies
2. **Build Baseline Model**
   - Start with Logistic Regression (simplest model)
   - Train on balanced dataset
   - Evaluate using:
     - Confusion Matrix
     - Precision, Recall, F1-Score
     - ROC-AUC Curve

- Document baseline performance

3. **Feature Importance Analysis**
   - Identify which features matter most
   - Visualize feature importance

Deliverable: Working baseline model + performance metrics

Week 13 (December 16 - 20) - Advanced Models & Comparison

Goal: Implement multiple ML algorithms and compare

Tasks:

1. **Implement Multiple Models**
   - Random Forest Classifier
   - Support Vector Machine (SVM)
   - XGBoost (Gradient Boosting)
2. **Train Each Model**
   - Use same train/test split for fair comparison
   - Record training time for each model
3. **Hyperparameter Tuning**
   - Use GridSearchCV or RandomizedSearchCV
   - Optimize for F1-score or ROC-AUC
   - Fine-tune the best performing model
4. **Model Comparison**
   - Create comparison table (precision, recall, F1, ROC-AUC)
   - Plot ROC curves for all models on same graph
   - Analyze confusion matrices
   - Select best model based on metrics

Deliverable: Trained models + comparison analysis + best model saved

---

Week 14 (December 23 - 27) - UI Development & Integration

Goal: Build user interface for your model

Tasks:

1. **Create Web Interface (Using Streamlit)**
   - Input form for transaction features
   - Display prediction (Fraud/Legitimate)
   - Show prediction probability/confidence
   - Add visualization of input data
2. **UI Features to Include**
   - Transaction amount input
   - Real-time prediction button
   - Result display with color coding (red for fraud, green for safe)
   - Confidence score visualization
   - Sample transaction testing
   - Model performance metrics display
3. **Test the Complete System**
   - Test with known fraud cases
   - Test with legitimate transactions
   - Check edge cases
   - Get feedback from team members

Deliverable: Working web application

Week 14 (January 6 - 10) - Report Writing & Documentation

Goal: Document everything professionally

Report Structure:

1. **Introduction** (1-2 pages)
   - Background on fraud detection
   - Problem significance
   - Project objectives

2. **Literature Review** (2-3 pages)
    - Existing fraud detection methods
    - Related research papers
    - ML techniques used in industry
3. **Dataset Description** (1-2 pages)
    - Dataset source and features
    - Data statistics and visualization
    - Class imbalance discussion
4. **Proposed Model/Methodology** (3-4 pages)
    - Data preprocessing steps
    - Handling class imbalance
    - Models implemented
    - Evaluation metrics explanation
5. **Experiments & Results** (3-4 pages)
    - Model comparison tables
    - Performance graphs (ROC curves, confusion matrices)
    - Analysis of results
    - Feature importance discussion
6. **Conclusion & Future Work** (1 page)
    - Summary of achievements
    - Limitations
    - Future improvements (deep learning, real-time streaming, etc.)

Code Documentation:

- Add comments to all functions
- Create README.md with setup instructions
- Document API endpoints (if any)
- Add requirements.txt file

Deliverable: Complete project report (15-20 pages)

Week 14 (Before Submission) - Presentation Preparation

Goal: Create killer 8-10 minute presentation

Presentation Structure:

1. **Slide 1**: Title + Team Members
2. **Slide 2**: Problem Statement + Motivation
3. **Slide 3**: Dataset Overview
4. **Slide 4**: Methodology (flowchart)
5. **Slide 5**: Model Comparison Results
6. **Slide 6**: Best Model Performance
7. **Slide 7**: Live Demo (UI showcase)
8. **Slide 8**: Conclusion + Future Work

Demo Preparation:

- Prepare 2-3 test cases (1 fraud, 2 legitimate)
- Record a backup demo video (in case of technical issues)
- Practice the demo multiple times
- Ensure your UI runs smoothly

Deliverable: PowerPoint/PDF + Demo ready

Final Submission Checklist (Friday, Week 14)

Must Include:

- ✅ Complete Project Report (PDF)
- ✅ GitHub Repository Link (with all code)
- ✅ Working UI (Streamlit app or web app)
- ✅ Presentation Slides
- ✅ Trained Model Files (.pkl or .h5)
- ✅ Requirements.txt

- ✅ README.md with instructions

GitHub Repository Structure:

```
fraud-detection/
├── README.md
├── requirements.txt
├── data/
│   └── creditcard.csv
├── notebooks/
│   ├── 01_EDA.ipynb
│   ├── 02_preprocessing.ipynb
│   └── 03_modeling.ipynb
├── models/
│   ├── logistic_model.pkl
│   ├── random_forest_model.pkl
│   └── xgboost_model.pkl
├── src/
│   ├── preprocessing.py
│   ├── train_model.py
│   └── evaluate_model.py
├── ui/
│   └── app.py (Streamlit app)
└── reports/
    ├── project_report.pdf
    └── presentation.pptx
```

Pro Tips for Success:

1. **Weekly Check-ins**: Meet with your team every week to sync progress
2. **Version Control**: Commit to GitHub regularly (don't wait till the end)
3. **Documentation**: Document as you go, not at the end
4. **Testing**: Test your code frequently to catch bugs early
5. **Ask for Help**: If stuck, consult your instructor or use online resources
6. **Time Buffer**: Aim to finish 2-3 days before the deadline for buffer time

---

Team Role Distribution (Suggested):

Member 1 (Data + EDA): Data preprocessing, EDA, visualization **Member 2 (Modeling)**: Model implementation, training, hyperparameter tuning **Member 3 (Evaluation + Report)**: Model evaluation, report writing, literature review **Member 4 (UI + Integration)**: Build UI, integrate model, prepare demo

Note: All members should understand the entire project and be able to explain it during presentation

---

Emergency Contingency Plan:

If behind schedule:

- Focus on 2-3 models instead of 4
- Use simpler UI (basic Streamlit forms)
- Simplify report (focus on methodology and results)
- Use pre-made visualizations from EDA phase

If dataset issues:

- Have backup dataset ready (IEEE-CIS Fraud Detection)
- Simplify feature engineering

If model performance is poor:

- Document why (imbalanced data challenges)
- Show multiple attempts and iterations
- Explain limitations and learnings

You Got This! 💪

Follow this roadmap week by week and you'll have a solid project. Start early, stay consistent, and don't hesitate to reach out if you need help with code, debugging, or any specific implementation!