Credit Risk Assessment System - FYP 2025

Complete credit risk assessment system using machine learning with PD (Probability of Default), LGD (Loss Given Default), and EAD (Exposure at Default) models.

Project Overview

This system provides:

- **PD Models**: Logistic Regression, Random Forest, Neural Network (weighted ensemble)
- **LGD Models**: Random Forest + XGBoost ensemble
- **EAD Models**: Random Forest + XGBoost ensemble
- **Expected Loss Calculator**: PD × LGD × EAD
- **Stress Testing**: Economic scenario analysis
- **Counterfactual Engine**: AI-powered loan improvement recommendations
- **Interactive UI**: Jupyter notebook interface for loan applications

Quick Start

1. Setup Environment

```
cd "c:\Users\Faheem\Desktop\Umair FYP\FYP2025"
python -m venv venv
venv\Scripts\activate
pip install -r credit_risk_fyp/requirements.txt
```

2. Run the Interactive Loan Application

```
jupyter notebook credit_risk_fyp/notebooks/interactive_loan_application.ipynb
```

Then:

1. Run all cells (Cell → Run All)
2. Use the sliders to input loan application details
3. Click "Check Loan Application"
4. View approval decision, risk metrics, and recommendations

3. Run Other Components

Stress Testing:

```
python run_stress_testing.py          # PD/LGD stress testing
python run_stress_testing_el.py       # Expected Loss stress testing
```

Counterfactual Recommendations:

```
python run_counterfactual_recourse.py  # Generate "what-if" scenarios
```

Expected Loss Calculator:

```
python run_expected_loss.py            # Calculate EL for portfolio
```

Project Structure

```
FYP2025/
├── credit_risk_fyp/
│   ├── data/
│   │   ├── processed/              # Train/test datasets
│   │   └── raw/                    # Original data
│   ├── models/                     # Trained model files (.pkl, .keras)
│   ├── notebooks/
│   │   ├── interactive_loan_application.ipynb  # Main UI ⭐
│   │   ├── ensemble_models.ipynb               # Ensemble training
│   │   ├── model_comparison.ipynb              # Model evaluation
│   │   ├── logistic_regression_clean.ipynb     # LR training
│   │   ├── random_forest_clean.ipynb           # RF training
│   │   ├── xgboost_improved_clean.ipynb        # XGBoost training
│   │   └── neural_network_results.ipynb        # NN training
│   ├── results/
│   │   ├── figures/                # Plots and visualizations
│   │   └── reports/                # CSV reports
│   ├── src/
│   │   ├── counterfactual_recourse.py    # AI recommendations
│   │   ├── ead_simulation.py             # EAD data generation
│   │   ├── ead_training.py               # EAD model training
│   │   ├── expected_loss_calculator.py   # EL computation
│   │   ├── lgd_simulation.py             # LGD data generation
│   │   ├── lgd_training.py               # LGD model training
│   │   ├── stacking_ensemble.py          # Stacking model
│   │   ├── stress_testing.py             # Stress test engine
│   │   ├── stress_testing_el.py          # EL stress testing
│   │   └── weighted_ensemble.py          # Weighted ensemble
│   └── requirements.txt
```

```
├── run_*.py                      # Convenience runner scripts
└── README.md                     # This file
```

Key Features

1. Interactive Loan Application UI

- **Location**: `credit_risk_fyp/notebooks/interactive_loan_application.ipynb`
- **Features**:
  - Real-time loan approval decisions
  - Tiered approval system (Prime/Standard/Subprime/Rejected)
  - Risk-based pricing with APR calculation
  - Monthly payment calculator
  - AI-powered improvement recommendations
  - Quick test scenarios (Excellent/Good/Fair/Poor profiles)

Approval Thresholds:

Tier	PD Threshold	Decision	Base APR
Prime	< 20%	✅ Approved	4.5%
Standard	20-30%	✅ Approved	6.0%
Subprime	30-40%	⚠️ Conditional	9.0%
High Risk	≥ 40%	❌ Rejected	-

Auto-Rejection Criteria:

- FICO < 580
- DTI > 43%
- Delinquencies ≥ 3
- Expected Loss > $8,000

2. Model Performance

PD Models (Test Set):

Model	AUC-ROC	Precision	Recall	F1-Score
Logistic Regression	0.7086	0.3425	0.5223	0.4138
Random Forest	0.7249	0.3415	0.6269	0.4421
XGBoost	0.7249	0.3415	0.6269	0.4421
Neural Network	~0.72	~0.34	~0.60	~0.43
Weighted Ensemble	**0.73**	**0.35**	**0.63**	**0.45**

LGD & EAD Models:

- Both use Random Forest + XGBoost ensemble
- LGD clipped to [0, 1]
- EAD with reverse standardization to actual loan amounts

3. Stress Testing

Economic Scenarios:

- **Baseline**: Normal conditions
- **Mild Recession**: +10% PD, +5% LGD
- **Severe Recession**: +25% PD, +15% LGD
- **Financial Crisis**: +50% PD, +30% LGD

Outputs:

- Stress test results: `credit_risk_fyp/results/stress_test_results.csv`
- Visualizations: `credit_risk_fyp/results/figures/stress_testing_*.png`

4. Counterfactual Recommendations

Strategies:

1. FICO Score improvement
2. DTI reduction
3. Loan amount reduction
4. Credit utilization improvement
5. Delinquency aging

Features:

- Immutable vs. actionable feature constraints
- Cost-based ranking (easier changes ranked higher)
- Feasibility checks (realistic improvement bounds)

Model Files

All models saved in `credit_risk_fyp/models/`:

- `logistic_regression_smote.pkl`
- `random_forest_smote.pkl`
- `xgboost_smote_improved.pkl`
- `neural_network_model.keras`
- `weighted_ensemble_metrics.pkl`
- `lgd_random_forest.pkl`, `lgd_xgboost.pkl`
- `ead_random_forest.pkl`, `ead_xgboost.pkl`

Data Files

Located in `credit_risk_fyp/data/processed/`:

- `train.csv` - Training data (SMOTE balanced)
- `val.csv` - Validation data
- `test.csv` - Test data
- `lgd_train.csv`, `lgd_test.csv` - LGD datasets
- `ead_train.csv`, `ead_test.csv` - EAD datasets

Running Individual Components

Train Models (if needed)

```
# Run notebooks in credit_risk_fyp/notebooks/
jupyter notebook credit_risk_fyp/notebooks/logistic_regression_clean.ipynb
jupyter notebook credit_risk_fyp/notebooks/random_forest_clean.ipynb
jupyter notebook credit_risk_fyp/notebooks/xgboost_improved_clean.ipynb
jupyter notebook credit_risk_fyp/notebooks/neural_network_results.ipynb
jupyter notebook credit_risk_fyp/notebooks/ensemble_models.ipynb
```

Generate Reports

All results automatically saved to:

- Figures: `credit_risk_fyp/results/figures/`
- Reports: `credit_risk_fyp/results/reports/`
- Logs: `credit_risk_fyp/results/logs/`

System Requirements

- Python 3.8+
- 8GB+ RAM recommended
- Windows/Linux/macOS
- Optional: NVIDIA GPU for neural network training

Dependencies

Key packages (see `requirements.txt` for full list):

- scikit-learn
- xgboost
- pandas, numpy
- matplotlib, seaborn
- tensorflow (for neural network)
- ipywidgets (for interactive UI)
- imbalanced-learn (for SMOTE)

Educational Value

This project demonstrates:

1. **Complete ML Pipeline**: Data → Models → Evaluation → Deployment

2. **Credit Risk Framework**: Industry-standard PD/LGD/EAD approach
3. **Ensemble Learning**: Combining multiple models for better performance
4. **Class Imbalance**: SMOTE for handling rare events
5. **Model Explainability**: Counterfactual recommendations
6. **Stress Testing**: Economic scenario analysis
7. **Interactive ML**: User-friendly model deployment

Important Notes

⚠️ **This is an educational/demonstration system**

- Not for actual lending decisions
- Requires regulatory compliance for production use
- Additional verification needed (income, employment, etc.)
- Human oversight required for real-world lending

✅ **Best used for**:

- Learning credit risk modeling
- Understanding ML in finance
- FYP/thesis demonstrations
- Educational purposes
- Prototype development

Troubleshooting

Jupyter widgets not showing?

```
pip install ipywidgets
jupyter nbextension enable --py widgetsnbextension
```

Models not loading?

- Ensure you've run the training notebooks first
- Check that model files exist in `credit_risk_fyp/models/`

Import errors?

```
pip install -r credit_risk_fyp/requirements.txt
```

Contact & Support

For FYP-related questions, refer to:

- Code comments in each module
- Notebook markdown cells
- This README

Built for FYP 2025 - Credit Risk Assessment Using Machine Learning