

Credit Risk Assessment Project Report

Project Overview

This project implements a credit risk assessment system that predicts the likelihood of loan default using machine learning. The system is designed to help financial institutions make informed lending decisions by evaluating the creditworthiness of loan applicants.

Key Components

1. Core Machine Learning

- **Model:** Random Forest Classifier
 - Handles class imbalance using class weights
 - Tuned hyperparameters for optimal performance
 - Includes model evaluation metrics (accuracy, classification report)

2. Data Processing

- Data preparation and feature engineering
- Train-test split with stratification
- Handling of missing values and feature scaling

3. API Layer

- **Framework:** FastAPI
- **Endpoints:**
 - `GET /`: Health check endpoint
 - `POST /predict`: Main prediction endpoint
- **Input Validation:** Pydantic models for request/response validation
- **Error Handling:** Comprehensive error handling and logging

4. Deployment

- Containerized using Docker
- Environment variable configuration
- CORS middleware for web client access

Technical Stack

- **Language:** Python 3.x
- **Machine Learning:** scikit-learn, joblib
- **Web Framework:** FastAPI
- **Containerization:** Docker, docker-compose
- **Development Tools:** Pydantic, logging, pytest

Project Structure

```
.
├── src/
│   ├── api/
│   │   ├── main.py          # FastAPI application
│   │   └── pydantic_models.py # Data models
│   ├── data_processing.py    # Data preparation
│   ├── predict.py           # Prediction logic
│   └── train.py              # Model training
├── tests/                   # Unit tests
├── notebooks/               # Jupyter notebooks for analysis
├── data/                    # Training data
├── models/                  # Trained models
├── Dockerfile               # Container configuration
├── docker-compose.yml       # Service orchestration
└── requirements.txt         # Python dependencies
```

Key Features

1. Model Implementation

- Random Forest algorithm for robust predictions
- Class weight balancing to handle imbalanced datasets
- Model persistence using joblib

2. API Features

- RESTful endpoints with proper HTTP status codes
- Input validation using Pydantic models
- Detailed error responses
- CORS support
- Request logging

3. Development Best Practices

- Type hints throughout the codebase
- Comprehensive docstrings
- Modular code organization
- Configuration management

Usage

Training the Model

```
python -m src.train --data_path data/credit_data.csv
```

Running the API

```
uvicorn src.api.main:app --host 0.0.0.0 --port 8000
```

Making Predictions

```
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'Content-Type: application/json' \
  -d '{
    "income": 50000,
    "debt": 10000,
    "credit_score": 720
  }'
```

Future Improvements

1. **Model Explainability:** Add SHAP or LIME for model interpretability
2. **Monitoring:** Implement model performance monitoring
3. **Feature Store:** Add a feature store for consistent feature engineering
4. **A/B Testing:** Support for model versioning and A/B testing
5. **Authentication:** Add API key authentication

Conclusion

This project provides a solid foundation for credit risk assessment with a focus on model performance, API reliability, and maintainability. The modular design allows for easy extension and integration with existing systems.