Credit Risk Prediction Using Ensemble and Deep Learning Models

Module 28: Advanced Machine Learning Project

1. Introduction

In the financial industry, accurately identifying customers who are likely to default on credit payments is a critical task. Incorrect predictions can lead to significant financial losses or missed risk signals. This project focuses on building an end-to-end **credit risk prediction system** using **advanced machine learning techniques**, with an emphasis on **model comparison, evaluation, and business-aware decision making**.

The goal is not only to achieve good performance but also to understand **which models are appropriate for tabular financial data** and **how evaluation metrics and thresholds affect real-world decisions**.

---

2. Problem Statement

The objective of this project is to **predict whether a customer will default on their credit payment in the next month** based on historical financial and demographic data.

This is formulated as a **binary classification problem**, where:

- 1 → Customer will default
- 0 → Customer will not default

In a financial context, **false negatives (missed defaulters)** are more costly than false positives, so evaluation focuses heavily on **recall and ROC-AUC**, not accuracy alone.

---

3. Dataset Description

The dataset used is the **Credit Card Default dataset**, containing records of **30,000 customers** with financial and demographic attributes.

Key Feature Groups:

- **Demographic features**: Credit limit, age, education, marital status
- **Payment history**: Past repayment status

- **Billing information**: Monthly bill amounts
- **Payment behavior**: Monthly payment amounts
- **Target variable**: Default in the next month

The dataset exhibits **class imbalance**, which reflects real-world financial risk distributions.

---

4. Data Preprocessing

The following preprocessing steps were performed:

- Train–test split with **stratification** to preserve class distribution
- Feature scaling using **StandardScaler** for models sensitive to feature magnitude (Logistic Regression, PCA, ANN)
- Raw (unscaled) features retained for **tree-based models**, which are scale-invariant

This separation ensured **correct preprocessing without data leakage**.

---

5. Baseline Model

A **Logistic Regression** model was used as the baseline to establish a minimum performance benchmark.

- Advantages: Interpretability, simplicity
- Limitations: Linear decision boundary

Despite its simplicity, the baseline achieved reasonable recall but showed limited ranking ability (ROC-AUC), motivating the use of more advanced models.

---

6. Ensemble Learning Models

Several ensemble models were implemented and evaluated:

6.1 Random Forest

- Reduced variance through bagging
- Captured non-linear feature interactions
- Provided stable but moderate recall

6.2 Gradient Boosting

- Sequentially corrected residual errors
- Improved ranking performance
- Required careful tuning to avoid overfitting

6.3 XGBoost

- Introduced regularization and optimized tree construction
- Handled class imbalance effectively
- Achieved the best balance between recall and ROC-AUC after tuning

Hyperparameter tuning was performed using cross-validation, optimizing for **recall**, which aligns with financial risk priorities.

---

7. Dimensionality Reduction (PCA)

Principal Component Analysis (PCA) was applied to scaled features:

- Retained **95% of variance**
- Reduced feature dimensionality
- Improved training efficiency for linear and neural models

PCA was found to be more beneficial for **linear and neural models** than for tree-based ensembles.

---

8. Neural Network (ANN)

A feed-forward **Artificial Neural Network (ANN)** was implemented to capture non-linear relationships.

- Architecture: Dense layers with ReLU activation
- Regularization: Dropout
- Optimization: Adam optimizer with binary cross-entropy loss

The ANN demonstrated reasonable ranking ability but required careful regularization and threshold handling to achieve acceptable recall.

---

9. Sequential Deep Learning Extension (LSTM)

As an advanced extension, a **Long Short-Term Memory (LSTM)** network was used to model **temporal payment behavior**.

Sequence Construction:

Monthly payment amounts were reshaped into sequences representing customer behavior over time.

Key Observations:

- Default threshold (0.5) resulted in zero recall due to class imbalance
- Lowering the threshold significantly improved recall
- ROC-AUC remained stable, demonstrating correct ranking behavior

This experiment highlighted the importance of **threshold tuning** in risk-sensitive applications.

---

10. Model Evaluation Strategy

Models were evaluated using finance-appropriate metrics:

- **Recall** – ability to catch defaulters
- **F1-score** – balance between precision and recall
- **ROC-AUC** – ranking quality independent of threshold

Accuracy was intentionally deprioritized due to asymmetric risk costs.

---

11. Final Model Selection

After comparison and tuning, **Tuned XGBoost** was selected as the final model due to:

- Highest ROC-AUC
- Strong recall performance
- Robust handling of non-linearity and class imbalance

The LSTM model served as a **behavioral modeling extension**, demonstrating how sequential data could be leveraged when richer transaction histories are available.

---

12. Conclusion

This project demonstrates the development of a **complete credit risk prediction system**, combining classical machine learning, ensemble methods, and deep learning.

Key takeaways:

- Tree-based ensemble models are highly effective for tabular financial data
- Deep learning models require careful threshold calibration
- Evaluation metrics must align with business risk, not just accuracy
- Model selection should be driven by both performance and practicality

---

13. Future Work

Potential improvements include:

- Incorporating transaction-level time series data
- Cost-sensitive learning
- Threshold optimization using ROC curves
- Model explainability techniques (SHAP)

---

14. Technologies Used

- Python
- Pandas, NumPy
- Scikit-learn
- XGBoost
- TensorFlow / Keras

---

15. Author

Jeswin K Reji
Aspiring Data Scientist
January 2026
jeswinkr7@gmail.com
LinkedIn GitHub