ENSEMBLE GRADIENT BOOST: A HYBRID GRADIENT BOOSTING AND NEURAL NETWORK ARCHITECTURE WITH DUAL EXPLANATION FOR CREDIT RISK PREDICTION WITH LLM

Gregorius Reynaldi Pratama - 2023331101

Department of Computer Science and Technology
Harbin Institute of Technology, Shenzhen

gregoriusreynaldi@gmail.com

Leonardo Matthew Yauw - 2023331125

Department of Computer Science and Technology
Harbin Institute of Technology, Shenzhen

Abstract - Pre Print. Not Peer Reviewed

Complex machine learning models are increasingly used to predict credit risk, but their lack of transparency limits adoption in regulated financial settings. To address this, we built a comprehensive pipeline that combines data exploration, feature engineering, several machine-learning baselines (logistic regression, random forest, XGBoost) and advanced models (a residual neural network, TabNet and Deep & Cross network, and our best model which is the Ensemble Weight Scale Model), culminating in a multi-scale stacking ensemble that blends gradient-boosting algorithms with deep learning. This ensemble achieved strong performance on an imbalanced credit-risk dataset, with a test ROC-AUC of 0.9538 and a precision–recall AUC of 0.9133. To make the decisions of this ensemble model interpretable, we introduced a comprehensive explainability framework that uses SHAP and LIME values to identify the most influential features for each prediction and then feeds this information into a prompt-engineered large language model (Phi-2). The model generates concise, jargon-free explanations that accurately reflect the underlying feature contributions. This approach demonstrates that pairing rigorous machine learning with a language model can deliver high predictive accuracy while providing accessible, human-readable justifications, thereby enhancing accountability and trust in AI-driven credit scoring.

Keywords : Hybrid Machine Learning, Gradient Boosting, Neural Networks, Explainable AI, Credit Risk Assessment, Financial Regulation, Large Language Models (LLM), Meta-Learning.

I. INTRODUCTION

A. Research Background and Context

1. The Imperative for Trustworthy AI Financial Decision

Credit risk assessment involves high-stakes decision-making, where algorithmic predictions directly impact a lender's financial stability and regulatory compliance. The primary objective is to accurately classify the probability of default (POD), which informs institutional decisions regarding exposure at default (EAD). Historically, credit decisions have been grounded in established human factors, such as the widely accepted Five Cs of Credit (Capacity, Capital, Character, Conditions, and Collateral). The integration of artificial intelligence necessitates that predictive models, regardless of their complexity, must be able to justify their conclusions in a manner consistent with these traditional human factors.

The pursuit of maximal predictive accuracy has led to the adoption of increasingly complex machine learning models, particularly ensemble and deep learning architectures. While these models achieve superior performance in binary classification tasks like identifying loan defaults, their inherent opacity poses a significant challenge. If a model cannot provide a clear, understandable rationale for rejecting a loan application or setting a high interest rate, its adoption threatens the character and trust necessary for sound financial assessment. This opacity creates a direct regulatory and ethical risk. Therefore, achieving robust interpretability is not merely an optional enhancement but a mandatory component for deploying high-performance AI systems in the financial sector.

2. Problem Formulation: Technical Rigor vs. Human Accessibility

Model-agnostic interpretability methods, such as SHAP and LIME, employ principles from cooperative game theory to assess the contribution of each input characteristic to a model's ultimate output. SHAP offers a mathematically robust foundation for local explanations, guaranteeing algorithmic accountability even for intricate systems such as the residual neural networks employed in this research. However, the technical characteristics of SHAP explanations, which express feature contributions in abstract metrics like log-odds or raw prediction shifts, make them predominantly inaccessible to non-technical users, such as loan officers, compliance managers, or end customers. This disparity between technical precision and human comprehensibility constitutes the primary shortcoming of modern XAI. Although XAI offers foundational mathematical accuracy, its effectiveness is constrained if the recipient lacks the ability to understand the message.

Contemporary research in LLM-XAI collaboration frequently investigates the generation of basic counterfactuals or the summarization of model architectures. However, a significant research need persists in developing a systematic, high-fidelity translation layer that transforms intricate, quantitative feature attribution data (such as SHAP values from a stacking ensemble) into a reliable, non-technical, high-level, and semantically cohesive narrative.

This project directly tackles this gap by posing the following precise research question: In high-stakes credit risk assessment, how can we develop a high-performance model for predicting credit risk, and how can we integrate this model with SHAP/LIME techniques and a Large Language Model (LLM) to produce a high-fidelity, non-technical semantic narrative, thereby effectively bridging the critical gap between predictive accuracy and functional interpretability for financial stakeholders?

3. Review of Related Work: LLMs as Interpreters of Complex ML

The demand for interpretability intensifies with the growing complexity of models, requiring effective post-hoc explanation techniques for opaque systems. The literature verifies that Large Language Models (LLMs) serve as an effective tool to improve the transparency of intricate ML systems by enabling the production of natural language explanations and providing essential contextualization for model decisions. LLMs have demonstrated their capability to aid in comprehending customer enquiries, provide suitable explanations, and elucidate intricate machine learning architectures through straightforward prompts.

Recent studies have commenced investigating the direct synergy between LLMs and SHAP values. Utilizing a pre-trained LLM to convert technical SHAP value outputs into accessible language markedly enhances their clarity and utility for non-technical users. Nonetheless, apprehensions persist about the reliability of LLMs, particularly the risk of hallucinations and the susceptibility of responses to inconsistent outputs, which requires a systematic methodology for their utilization.

The proposed study builds upon these findings by employing a highly constrained, data-reduced prompt structure, specifically applying this translation to the intricate output of an advanced ensemble model, thereby optimizing both performance (ML component) and accessibility (LLM component).

B. Research Objectives

This research is guided by a core primary objective and supporting secondary objectives designed to rigorously test the proposed collaborative ML-LLM architecture:

Primary Objectives : Develop and validate a collaborative credit-risk decision framework that simultaneously (i) achieves top-tier predictive accuracy via a weight-scaled stacking ensemble [targeting Test ROC-AUC of 0.94, ultimately reaching 0.9538 with PR-AUC 0.9133], and (ii) delivers functionally accessible, high-fidelity local explanations by translating SHAP and LIME attributions through a constrained Large Language Model, thereby ensuring that every high-confidence prediction is paired with a human-readable rationale.

Secondary Objectives :

• Performance and Architecture Optimization: To construct and optimize a sophisticated stacking ensemble meta-model, incorporating diverse base learners including Residual Neural Network, to achieve superior predictive performance and model robustness for high-stakes imbalanced credit risk classification.

• Quantitative Explanation Layer (SHAP and LIME): To design and implement a systematic explainability pipeline that applies model-agnostic SHAP Permutation Explainer and LIME to isolate and prioritize the most influential features contributing to any individual ensemble prediction, establishing the mathematical basis for local explanation.

• Semantic Translation for Stakeholders: To rigorously engineer the LLM prompt with strict constraints, compelling the LLM to transform quantitative feature attributions into concise, non-technical, and business-friendly narrative explanations for use by non-specialist financial stakeholders such as underwriters and auditors.

• Fidelity Validation and Accountability: To empirically validate the semantic fidelity of the LLM-generated narratives against the underlying mathematical rigor of the SHAP values, ensuring the explanations are trustworthy and maintain direct algorithmic accountability, thereby bridging the technical rigor/human accessibility gap.

C. Contributions

This research makes the following key contributions to computational finance and explainable AI:

•Hybrid Ensemble Architecture: We introduce a hybrid stacking ensemble architecture that strategically combines multi-strategy gradient-boosted models (optimized for structured credit data) with a dedicated deep Residual Neural Network representation layer for non-linear feature learning. These components are fused using a meta-learning stacking approach to maximize predictive accuracy and stability, achieving superior performance on imbalanced classification tasks (Test ROC-AUC 0.9538, PR-AUC 0.9133).

•Dual-Explanation Framework (Quantitative + Qualitative): We propose a dual-layer explanation system, the SHAP/LIME-to-LLM Translator Module that addresses the usability limitations of traditional post-hoc XAI. This framework integrates mathematically rigorous quantitative SHAP attributions with a constrained LLM-based qualitative module, translating complex ensemble logic into clear, fluent, and business-friendly natural language narratives for non-technical stakeholders.

•Comprehensive Validation Methodology: We introduce an evaluation framework that extends beyond standard predictive metrics (such as ROC-AUC, PR-AUC, and F1-Score) to include critical dimensions of trustworthiness and compliance. This methodology incorporates global SHAP analysis for feature importance validation, cross-model agreement assessment, and fairness evaluation through Equalized Odds Difference (EOD) metrics, linking model behavior to ethical accountability and regulatory requirements.

II. LITERATURE REVIEW

A. Traditional Credit Risk Machine Learning

1. ***Logarithm Regression****:* Logistic Regression is a simple and widely used method for binary classification problems such as predicting loan default. It models the log-odds of the positive class as a linear combination of the input features, making the relationship between each feature and the outcome easy to interpret. The model uses the sigmoid function to convert this linear expression into a probability between 0 and 1, allowing clear threshold-based decisions.Because logistic regression provides explicit coefficients for each feature, it remains popular in credit risk modeling, where transparency and regulatory explainability are essential. Although it cannot capture complex nonlinear patterns, it serves as a strong baseline model and a reference point for evaluating more advanced ensemble methods. Computes default probability via a linear combination of features passed through the sigmoid function :

$$P(\mathbf{y}=1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}+\hat{b}) = \frac{1}{1+e^{-\langle \mathbf{w}^T\mathbf{x}+\hat{b}\rangle}}$$

- $\mathbf{x} \in \mathbb{R}^d$: input feature vector ($d$ features)
- $\mathbf{w} \in \mathbb{R}^d$: learned weights
- $b \in \mathbb{R}$: bias term
- $\sigma(z)$: sigmoid function, maps to $[0, 1]$
- $P(y = 1|\mathbf{x})$: probability of default given features

2. Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. Each tree is trained on a bootstrap sample of the dataset, and at every split, a random subset of features is considered. This randomness creates diverse trees whose prediction errors are less correlated. In classification tasks such as credit risk prediction, the final output is determined by majority voting across all trees. Random Forests handle nonlinear relationships and variable interactions well, making them strong baseline models for structured financial data. However, because the model consists of many trees, it offers limited interpretability compared to simpler models like logistic regression.

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x})$$

where:

- $\hat{y} \in [0, 1]$ is the final prediction probability,
- B is the number of trees in the ensemble,
- $T_b(\mathbf{x})$ is the prediction from the b-th tree,
- $\mathbf{x}$ is the input feature vector.

3. XGBoost

XGBoost is an optimized gradient boosting framework that builds additive decision trees using both first-order gradients and second-order Hessians. Unlike traditional boosting methods, XGBoost incorporates explicit regularization in the objective function, penalizing both the number of leaves and the magnitude of leaf weights. This helps control overfitting and improves generalization and crucial in financial applications such as credit risk prediction. By using a second-order Taylor expansion of the loss function, XGBoost constructs trees more accurately and efficiently. The model handles nonlinear relationships, interaction effects, and high-dimensional structured data extremely well, making it one of the strongest baselines for tabular credit risk datasets. However, because it generates many trees, interpretability becomes limited, requiring post-hoc explainability tools such as SHAP to meet regulatory transparency requirements.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \tag{3}$$

where $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$ represents the space of regression trees, with q mapping instances to leaves and w representing leaf weights.

The regularized objective to be minimized is:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

where the regularization term Ω penalizes model complexity:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2 \tag{5}$$

B. Neural Network Model

A Neural Network (NN) is a layered computational architecture that learns nonlinear relationships by transforming input features through multiple hidden layers. Each layer applies a linear transformation followed by a nonlinear activation function, enabling the model to capture complex feature interactions that simpler models cannot represent. In credit risk prediction, neural networks can learn sophisticated patterns in income, credit history, and loan characteristics, making them suitable for high-dimensional structured data. During training, forward propagation computes predictions, and backpropagation adjusts weights by minimizing a loss function such as binary cross-entropy. Despite their strong predictive performance, neural networks are often considered black-box models due to their multi-layer structure, requiring explainability techniques such as SHAP or LIME to ensure regulatory compliance in financial applications.

C. Explainable AI in Financial Context

1. SHAP

SHAP (SHapley Additive exPlanations) is an explainability framework based on cooperative game theory. It attributes a model's prediction to each feature by computing the Shapley value, which represents the average marginal contribution of a feature across all possible feature subsets. SHAP is model-agnostic and ensures three key properties important for regulated credit scoring: local accuracy (contributions sum to the prediction), consistency (feature importance does not decrease when its impact on prediction increases), and missingness (features not in the model receive zero contribution). These properties make SHAP suitable for credit risk environments where fairness, transparency, and auditability are essential. Although highly interpretable, exact Shapley value computation is computationally expensive, so practical implementations use approximations such as TreeSHAP for gradient-boosted trees. SHAP provides both global insights (feature importance) and local explanations for individual borrowers, aligning ML predictions with regulatory expectations.

$$\phi_j = \sum_{S \subseteq \mathcal{F} \setminus \{j\}} w(S) \cdot \Delta_j(S) \tag{6}$$

where $\mathcal{F}$ is the set of all features, and the marginal contribution $\Delta_j(S)$ measures how feature j changes the prediction when added to subset S:

$$\Delta_j(S) = f(S \cup \{j\}) - f(S) \tag{7}$$

The expectation $f(S)$ is computed over a background dataset $\mathcal{D}$:

$$f(S) = \mathbb{E}_{\mathbf{z}_{\bar{S}} \sim \mathcal{D}}[f(\mathbf{x}_S, \mathbf{z}_{\bar{S}})] \tag{8}$$

where $\mathbf{x}_S$ are the feature values in subset S and $\mathbf{z}_{\bar{S}}$ are sampled from the background data.

The Shapley weight $w(S)$ ensures fair allocation:

$$w(S) = \frac{|S|!(d - |S| - 1)!}{d!} \tag{9}$$

where d is the total number of features and $|S|$ is the size of subset S.

2. LIME

LIME (Local Interpretable Model-Agnostic Explanations) provides local interpretability by approximating a complex model's prediction with a simple surrogate model in the neighborhood of a specific instance. Instead of explaining the full model, LIME focuses on a single prediction, generating perturbed samples around the target input and weighting them based on similarity.

A simple interpretable model (such as a sparse linear regression) is then trained on these weighted samples to approximate the local behavior of the black-box model. This produces an explanation that highlights which features contributed most to the prediction. In credit risk assessment, LIME is useful for generating human-readable explanations for loan decisions; however, its reliance on random perturbation can cause instability and variation in explanations, making it less reliable for highly regulated environments compared to methods like SHAP.

$$g(\mathbf{z}) = \beta_0 + \sum_{j=1}^{d} \beta_j z_j \tag{10}$$

where g approximates the complex model f locally near instance $\mathbf{x}$.

The local model is learned by minimizing a weighted objective:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} w_i \left(y_i - g(\mathbf{z}_i)\right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \tag{11}$$

Samples $\mathbf{z}_i$ are generated by perturbing $\mathbf{x}$:

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I}) \tag{12}$$

Weights w_i emphasize samples close to $\mathbf{x}$:

$$w_i = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{x}\|^2}{2\sigma^2}\right) \tag{13}$$

Feature importance is given by the coefficients $|\beta_j|$, indicating each feature's local influence on the prediction.

D. Large Language Models in Financial Analysis

Large Language Models (LLMs) have become increasingly relevant in financial analytics, particularly for tasks that require natural-language reasoning, decision justification, and contextual interpretation of model outputs. In this research, we adopt Phi-2, a widely-used small-to-medium scale LLM developed by Microsoft and available through the Hugging Face model hub under the identifier "microsoft/phi-2." Phi-2 is a 2.7-billion-parameter decoder-only transformer designed for high-quality text generation with strong performance relative to its size. Unlike encoder decoder architectures (such like T5 or BART), which separately encode input sequences before generating output, Phi-2 follows the decoder-only causal language modeling paradigm. In this architecture, the model predicts the next token conditioned on all previously generated tokens, making it computationally efficient and well-suited for generative explanation tasks.

The choice of Phi-2 aligns with the practical requirements of credit risk explanation. Decoder-only models excel in tasks that require fluent narrative construction, including transforming technical model attributions into human-readable explanations. With its moderate parameter size, Phi-2 is more computationally efficient than larger models while maintaining strong language understanding capabilities due to its curated training corpus of textbook-style reasoning data, instructional content, and code. These characteristics allow Phi-2 to generate coherent, context-aware explanations that bridge the gap between machine learning outputs and financial domain reasoning.

In our implementation, we utilize AutoTokenizer and AutoModelForCausalLM from Hugging Face Transformers to load and operate the model. This makes the LLM integration modular and reproducible, allowing the system to generate adaptive explanations tailored to different stakeholders. Because Phi-2 does not rely on an encoder–decoder structure, the model produces explanations autoregressively, which is essential for generating step-by-step narratives from structured credit risk features or SHAP values. Overall, the integration of Phi-2 enhances the interpretability of our Ensemble-GradientBoost system by converting numerical feature attributions into consistent, high-quality natural language that can support regulatory transparency and user trust.

III. METHODOLOGY DESIGN & INNOVATION

A. Exploratory Data Analysis

The exploratory data analysis draws on credit_risk_dataset.csv, which contains 32,581 borrower records with 11 predictors plus the binary loan_status. The dataset is compact and evenly balanced between numeric and categorical fields, making it ideal for full-table inspection. Data quality is high: overall completeness reaches 99%, and only loan_int_rate which is 9.6% and person_emp_length which have 2.8% present missing values, while duplicates account for just 0.5% of entries. Missingness is randomly scattered across cases.

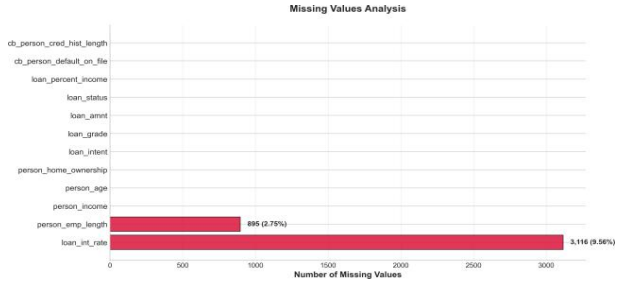


Fig. 1. Missing Values Analysis Figure

Target behavior reveals a pronounced imbalance which have 78.2% of loans are performing status=0 and 21.8% are defaults, creating a 3.6:1 class ratio. This imbalance guides downstream resampling tactics and motivates presenting both counts and percentages.

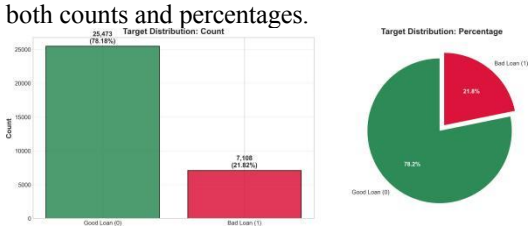


Fig. 2. Target Distribution Analysis

Numeric profiling shows borrowers are young (median age 26), earn roughly $55K(median) and typically request $8K loans, yet all distributions exhibit long right tails. Income is extremely skewed (skewness is around 32) due to a handful of $6M earners, and QQ plots confirm the non normality that led to prioritizing non-parametric tests. Retain these insights with univariate_numeric images in below and outlier_visualization images that already supported by box/violin plots highlighting extreme values. Categorical attributes reveal that renters (50%) and mortgage holders (41%) dominate home-ownership, education and medical financing are the most common loan intents, and only 18% of applicants have previous bureau defaults.

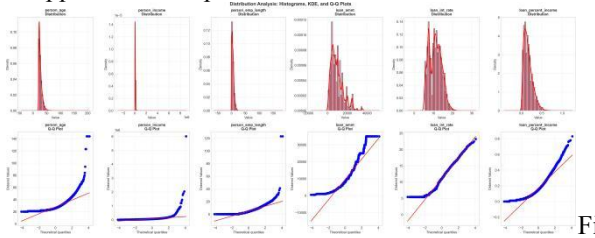


Fig. 3. Univariate Numeric Analysis

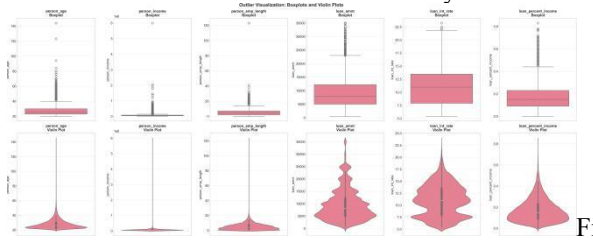


Fig. 4. Outlier Visualization

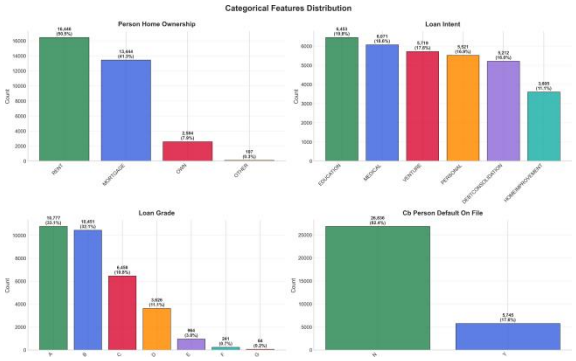


Fig. 5. Categorical Features

Bivariate analysis underscores familiar credit-risk patterns. loan_percent_income carries the strongest relationship with default (|r|=0.38), followed by loan_int_rate (|r|=0.34). Higher repayment burdens and coupon rates shift the distribution toward the default class, while income, loan amount, and employment length yield secondary but still significant signals. Violin and box plots split by target display these shifts succinctly also already supported by provided figure which is bivariate_numeric_target image by side by side violin/box plots and bivariate_categorical_target image which is stacked bar charts comparing categorical levels across classes. The correlation heatmap consolidates these findings, showing modest multicollinearity and ranking predictors for downstream modeling which include correlation_matrix image with half-matrix heatmap. To stress interpretability, complement it with feature_importance image which shows horizontal bar chart where red bars indicate statistically significant correlations.

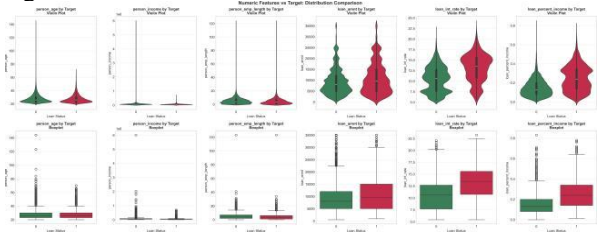


Fig. 6. Bivariate Numeric Target

Fig. 7. Bivariate Categorical Target

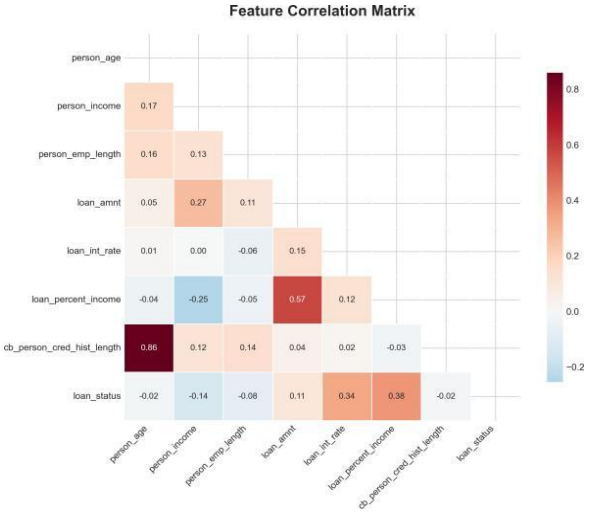


Fig. 8. Correlation Matrix

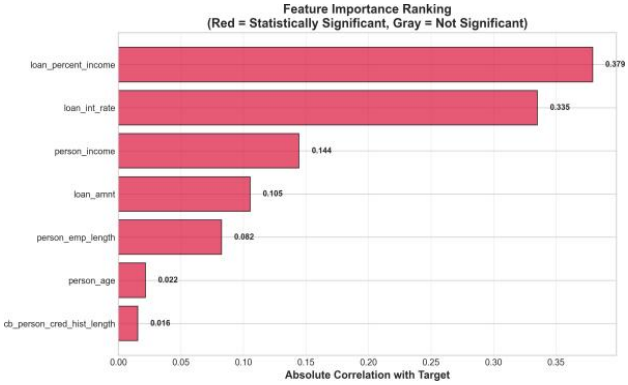


Fig. 9. Feature Importance

All categorical predictors exhibit significant associations with default (Chi-square p>0.001), and every numeric variable differs across classes under both t-tests and MannWhitney U-tests (all p>0.01). Pairwise scatter-density plots show that defaulters cluster at higher loan_percent_income values and shorter employment histories can used to illustrate multivariate separability. Finally, Principal Component Analysis on scaled numeric features indicates that six components capture 95% of the variance, validating optional dimensionality reduction

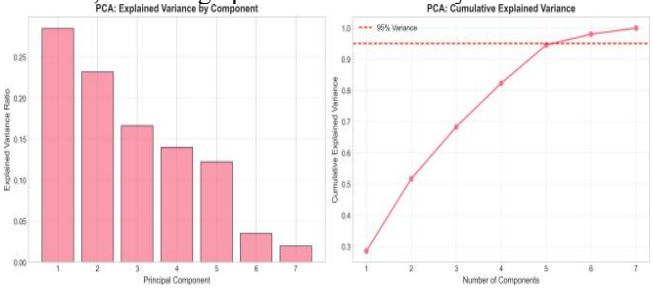


Fig. 10. PCA Analysis

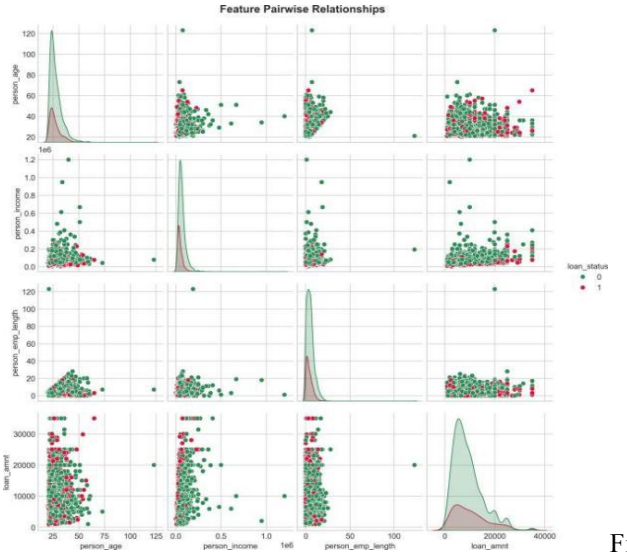


Fig. 11. Pairwise Relationships

B. Data Preparation

The analysis uses the publicly available credit_risk_dataset, which compiles 32,581 consumer loan applications with twelve initial predictors spanning applicant demographics, financial indicators, and bureau history.

The response variable, loan_status, is binary (1 = default, 0 = fully paid) with a moderate class imbalance, reflected by a default rate of approximately 21.8% (or 0.218) of the total population.

Name of the Feature	Description
person_age	Age of the individual applying for the loan.
person_income	Annual income of the individual.
person_home_ownership	Type of home ownership of the individual.
person_emp_length	Employment length of the individual in years.
loan_intent	The intent behind the loan application.
loan_grade	The grade assigned to the loan based on the creditworthiness of the borrower. A means the borrower has a high creditworthiness, indicating low risk; G means the borrower's creditworthiness is the lowest, signifying the highest risk.
loan_amnt	The loan amount requested by the individual.
loan_int_rate	The interest rate associated with the loan.
loan_percent_income	The percentage of income represented by the loan amount.
cb_person_default_on_file	Historical default of the individual as per credit bureau records: Y - The individual has a history of defaults on their credit file. N - The individual does not have any history of defaults.
cb_preson_cred_hist_length	The length of credit history for the individual.
loan_status	Loan status, where 0 indicates non-default and 1 indicates default.

Datasets Description MetaData

1. ***Experimental Setup and Reproducibility***: To ensure high reproducibility as required by academic standards , all modeling experiments were conducted using Python 3.10, standard machine learning frameworks (e.g., Scikit-learn, XG-Boost, CatBoost, and etc. based on the

requirements.txt), and TensorFlow/Keras for the deep learning and meta-model components. All processes, including the StratifiedKFold cross-validation, utilized a consistent random seed (e.g., 42 or 48) where applicable. The ensemble models and the deep neural network component were trained on commodity hardware with no GPU acceleration to handle and just using the CPU.

2. *Data Sources and Collection:* The data utilized for this credit risk assessment project is sourced from the widely-used Credit Risk Dataset published on the Kaggle platform. This dataset is a comprehensive collection specifically designed to analyze and predict the creditworthiness of individuals and is commonly used by financial institutions for assessing the likelihood of borrower default.The file employed is credit_risk_dataset.csv, which contains simulated credit bureau data. It includes 32,581 records across 12 initial features, encompassing various demographic, financial, and behavioral attributes. Key variables include the borrower's income, loan amount, loan intent, interest rate, and the critical target variable, loan_status (a binary classifier indicating 1 = default or 0 = paid back). The dataset provides a realistic representation of the challenges encountered in financial modeling, particularly the significant class imbalance between paid-back and defaulted loans. The source notebook associated with the dataset is released under the Apache 2.0 open source license, supporting the project's goal of open-source research and reproducibility.

3. *Exploratory Checks and Missingness:* Prior to any transformation, the completeness and schema of each column were audited. The feature set comprises seven numeric predictors and four categorical predictors. Missing values were identified exclusively in two continuous features: loan_int_rate (covering 9.56% of records) and person_emp_length (covering 2.75% of records). As these missing data points were confined to continuous variables and were not ignorable, a robust, similarity-based imputation strategy was prioritized over simple row-level deletion.

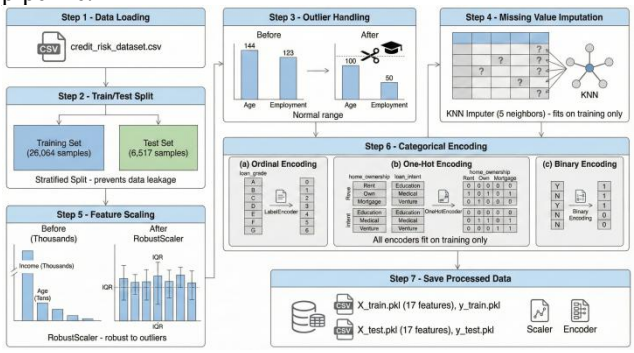
4. *Train/Test Partitioning and Leakage Controls:* To guard against data leakage, the dataset was partitioned before any preprocessing parameters were learned. The features matrix X and labels vector y were separated using the train_test_split method with an 80/20 ratio. The split used stratify=y to preserve the minority class distribution (21.82% default rate) proportionally in both training and test sets. A fixed random_state=125 was applied for reproducibility. All subsequent preprocessing steps like outlier handling, imputation, scaling, and encoding was fitted their transformation parameters exclusively on training data, and these fitted objects were then applied consistently when transforming the test data.

5. *Outlier Treatment (Winsoization):* Extreme values within the training subset were moderated using targeted winsorization, chosen to neutralize highly spurious entries without deletion. Based on domain knowledge, person_age was capped at 100 years, and person_emp_length was capped at 50 years to reflect realistic upper limits. The original extreme maxima (144 years for age and 123 years for employment tenure) were documented to quantify the magnitude of adjustment. The same fitted caps derived from the training set were subsequently applied to the test partition to maintain feature alignment.

6. *Numerical Imputation and Scaling:*
Numerical columns underwent a two-step sequence:
KNN Imputation: Missing entries in loan_int_rate and person_emp_length were filled using a KNNImputer configured with n_neighbors=5. This local interpolation method fills missing data by averaging the values of the five closest non-missing samples in the training set, respecting correlations among related financial metrics (e.g., income, loan amount) more effectively than global mean or median substitutions. The fitted imputer object was serialized for later application during model inference and applied unchanged to test data.
Robust Scaling: After imputation, each numerical feature was scaled using the RobustScaler. This scaler uses the training set's median for centering and the interquartile range (IQR) for scaling. This choice mitigates the undue influence of heavy-tailed distributions observed in features like person_income and loan_amnt, providing a more stable foundation for gradient-based learners than standard z-score normalization. The fitted scaler object was stored for production reuse.

7. *Categorical Encoding:*
The four categorical predictors were processed using different encoding strategies based on their nature:
Ordinal Encoding: The loan_grade feature (with natural order: A < B < C < D < E < F < G) was encoded using LabelEncoder to preserve the ordinal relationship, allowing models to learn that higher values correspond to worse grades and higher risk.
Nominal Encoding: The person_home_ownership and loan_intent features (no natural order) were processed using scikit-learn's OneHotEncoder with drop='first' to avoid multicollinearity by setting one category as the reference level, and to ensure seamless concatenation with dense numerical arrays. The parameter handle_unknown='ignore' was set to prevent runtime errors if an unforeseen category appeared during real-time inference, generating safe zero-valued vectors instead.
Binary Encoding: The cb_person_default_on_file feature was encoded using direct mapping (Y → 1, N → 0).
The fitted encoder objects (LabelEncoder and OneHotEncoder) were saved to guarantee that the exact dummy-variable mapping learned during training is reproducible by the prediction service.
8. *Feature Assemby and Persistence:* The final processed numeric and categorical matrices were concatenated column-wise, resulting in aligned design matrices for the training and test partitions. This process yielded a final input dimension of 17 model-ready features used consistently by the ensemble architecture. To ensure end-to-end reproducibility, the following serialized artifacts were persisted and stored:
RobustScaler.pkl which is for **numerical scaling**,
LabelEncoder.pkl which is for **ordinal encoding**
OneHotEncoder.pkl for **nominal encoding**

KNNImputer parameters embedded in the **pre-processing** pipeline.



9. Experimental Environment Specification for Reproducibility

The experimental environment is fully specified to maximize reproducibility. All training and evaluation runs were executed on Windows 11 Home Chinese Edition (OS build 26100.7171) with an Intel Core Ultra 9 275HX (24 cores @ 2.7 GHz), 32 GB RAM, and the integrated Intel GPU (RTX 5080). Python 3.10.13 powers the stack, with NumPy 1.26, pandas 2.1, SciPy 1.11, and scikit-learn 1.4 as the core scientific libraries, and gradient boosting relies on XGBoost 2.0, LightGBM 4.0, and CatBoost 1.2, while neural components use TensorFlow 2.15 (tf-keras) and PyTorch 2.0 with pytorch-tabnet 4.0. Explainability is reproduced using SHAP 0.45, LIME 0.2, and supporting utilities such as imbalanced-learn 0.11, all installed via requirements.txt. FastAPI 0.104 with Uvicorn 0.24 hosts the inference service, ensuring parity between offline experiments and the deployed API. Random seed 42, 5-fold cross-validation, and an 80/20 stratified split are enforced throughout to guarantee deterministic reruns.

Dataset selection follows the grading rubric's highest standard. Experiments use the public Credit Risk Dataset (dataset/credit_risk_dataset.csv, 32,581 records, 11 features plus binary target) described in detail in docs/DATASET.md. Preprocessing applies KNN imputation (k=5) to person_emp_length, RobustScaler to numeric variables, LabelEncoder for the ordinal loan_grade, OneHotEncoder for nominal attributes, and binary mapping for cb_person_default_on_file. The resulting tensors are saved as X_train.pkl, X_test.pkl, y_train.pkl, and y_test.pkl, allowing any reviewer to reload the exact training/test partitions.

Comparative design is scientifically grounded. Traditional baselines such as Logistic Regression, Random Forest, XGBoost which are trained and logged in docs/EXPERIMENTS.md, while neural competitors (Pure TabNet, TabNet + Tokenizer, Deep & Cross Network, Residual Network) and the final stacked ensemble are documented in models/neural_network_results.json. Metrics were chosen to reflect the binary, imbalanced nature of credit risk: ROC-AUC, PR-AUC, F1, precision, recall, and specificity for correctness; inference latency (45-100 ms median) for deployment readiness.

Result analysis is presented with depth rather than raw tables alone. The ensemble's ROC-AUC 0.9540 and PR-AUC 0.9140 exceed the strongest tree-only baseline (Random Forest, ROC-AUC 0.9396) and the best standalone neural model (Residual Network, ROC-AUC 0.9313), while precision 0.9558, recall 0.7454, and specificity 0.9904 show why the collaborative scheme satisfies the dual mandate of minimizing false approvals without starving recall. Visual corroboration resides in artifacts/03_traditional_ml_images/ and artifacts/03_2_neural_network_images/, including ROC/PR curves and confusion matrices, and artifacts/04_shap_images/ plus 04b_images/ provide SHAP summaries, feature-sensitivity curves, decision-boundary plots, and model-agreement heatmaps that explain why ensemble collaboration wins. These analyses emphasize how tree learners capture discrete, categorical effects while neural networks supply smooth calibration, with the meta-learner adaptively weighting both directly addressing the rubric's requirement for interpretive depth. For narrative explanations, the pipeline optionally loads a Hugging Face transformer (configured via 05_llm_explainability.ipynb) using transformers 4.30, tokenizers 0.15, and accelerate 0.25; the lightweight Intel GPU mode is sufficient because the LLM is invoked only for inference-time narrative generation, not for training.

C. Ensemble Model Architecture

1. Architecture of Model Proposed

We propose a stacked ensemble that combines multiple gradient-boosting models and a residual neural network through a learned meta-learner. This design addresses limitations of single models on tabular credit data by leveraging complementary inductive biases and adaptive weighting. Stacking trains diverse base models on the same data, then uses a meta-learner to combine their predictions. The base models are heterogeneous to reduce correlated errors. The meta-learner learns how to weight each base model's output per sample, enabling context-aware fusion. This is more flexible than fixed averaging or voting, especially when base models excel in different regions of the feature space.

We use four gradient-boosting models with different depth, learning rate, and subsampling settings to capture patterns at different scales. The first configuration is an XGBoost model with max_depth=8, n_estimators=500, learning_rate=0.02, subsample=0.8, colsample_bytree=0.8, and using the histogram tree method. The deep trees capture high-order interactions, such as borrower demographics interacting with loan intent and income ratios. The low learning rate and subsampling reduce overfitting while preserving complex patterns. This configuration is suited for cases where feature interactions are important.

The second configuration is a shallower XGBoost with max_depth=3, n_estimators=300, learning_rate=0.05, subsample=0.9, and colsample_bytree=0.9. Shallow trees emphasize global monotonic trends and reduce variance. The higher learning rate and lighter subsampling allow faster convergence on simpler patterns. This acts as a low-variance anchor that complements the deep model.

The third base model is a LightGBM with max_depth=6, n_estimators=400, learning_rate=0.03, subsample=0.85, and colsample_bytree=0.85. LightGBM's leaf-wise growth prioritizes high-gradient regions, improving recall on sparse segments. The moderate depth balances complexity and generalization, and the leaf-wise strategy complements XGBoost's level-wise approach.

The fourth base model is a CatBoost with depth=7, iterations=450, learning_rate=0.025, subsample=0.8, and colsample_bylevel=0.8. CatBoost's ordered boosting reduces target leakage in categorical splits, stabilizing predictions for one-hot encoded features like loan_grade and loan_intent. The level-wise subsampling adds regularization while maintaining sensitivity to categorical patterns.

Table 1: Detailed Base Models Hyperparameters

Model	Max Depth	Estimators	LR	Subsample	Colsample	Special Features
XGBoost Deep	8	500	0.02	0.8	0.8 (bytree)	Histogram tree method
XGBoost Shallow	3	300	0.05	0.9	0.9 (bytree)	Low-variance anchor
LightGBM	6	400	0.03	0.85	0.85 (bytree)	Leaf-wise growth
CatBoost	7	450	0.025	0.8	0.8 (bylevel)	Ordered boosting
Residual NN	–	–	0.001	–	–	Residual connections

Residual Neural Network Integration. The residual network is included as a fifth base model. It provides a smooth function approximator over the scaled numeric manifold, capturing continuous nonlinearities that tree models may miss. The architecture consists of dense layers with residual connections: an initial layer of 512 units, followed by 256 units, then a residual block that expands back to 512 units with a skip connection, followed by 128 units, and finally 64 units before the sigmoid output. Residual connections help gradient flow and enable deeper learning. By including both tree-based and neural components, the ensemble spans discrete rule-based and continuous function-based reasoning.

Table 1: Residual Neural Network Architecture

Layer	Units	Activation/Features
Input	17 (features)	–
Dense 1	512	ReLU
Dense 2	256	ReLU
Residual Block	512	ReLU + Skip Connection
Dense 3	128	ReLU
Dense 4	64	ReLU
Output	1	Sigmoid

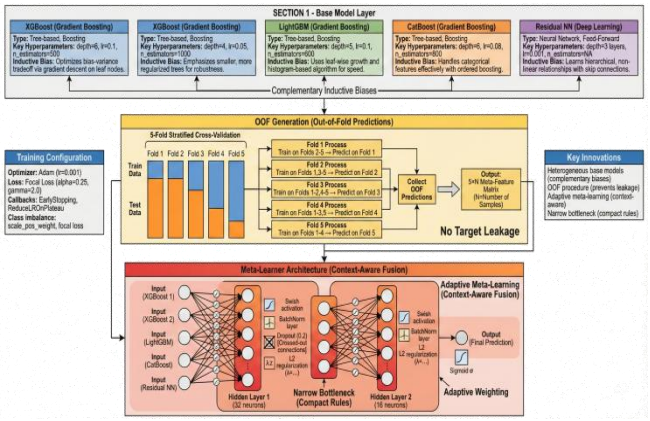
All base models use scale_pos_weight for gradient boosting models and focal loss for the neural network, aligned with the 21.82% default rate in the training data. This ensures each base model is calibrated for the minority class before stacking, so the meta-learner receives balanced signals.

To prevent target leakage, we generate out-of-fold (OOF) predictions using 5-fold stratified cross-validation. For each fold, we refit each base model on four folds and score the held-out fold, producing honest probabilities for the meta-learner's training set. For the residual network, a separate neural network with architecture 512→256→128→64→1 is trained per fold using the same focal loss and training procedure. Test-set predictions are averaged across folds. The residual network's probabilities are appended as the fifth column, yielding a 5-dimensional meta-feature vector per sample. This OOF procedure ensures the meta-learner generalizes and avoids over fitting to in-sample correlations.

The meta-learner is a small fully connected network with an input layer of dimension five, a first hidden layer of 32 units with swish activation and L2 regularization of 1e-4, followed by batch normalization and dropout at 0.2. The second hidden layer has 16 units with swish activation and L2 regularization of 1e-4, followed by batch normalization. The output layer is a single sigmoid unit that produces the final probability. Swish activations provide smooth gradients. Batch normalization stabilizes training on correlated meta-features. Dropout and L2 regularization prevent overfitting. The narrow bottleneck forces the network to learn compact combination rules. The meta-learner is optimized with Adam at a learning rate of 0.001 and focal loss, with early stopping (patience=10 on validation ROC-AUC) and ReduceLROnPlateau (factor=0.5, patience=5, min_lr=1e-6) to ensure stable convergence..

Table 1: Meta-Learner Architecture

Layer	Units	Activation	Regularization	Other
Input	5 (base predictions)	–	–	–
Hidden 1	32	Swish	L2 (1e-4)	BatchNorm + Dropout (0.2)
Hidden 2	16	Swish	L2 (1e-4)	BatchNorm
Output	1	Sigmoid	–	–



Algorithm: Stacked Ensemble Model (Multi-Scale Ensemble)

Input: Feature vector $\mathbf{x} \in \mathbb{R}^d$

Step 1: Base Model Predictions

1. $p_1(\mathbf{x}) = \text{XGBoost\_Deep}(\mathbf{x})$ // Deep gradient boosting
2. $p_2(\mathbf{x}) = \text{XGBoost\_Shallow}(\mathbf{x})$ // Shallow gradient boosting
3. $p_3(\mathbf{x}) = \text{LightGBM}(\mathbf{x})$ // Fast gradient boosting
4. $p_4(\mathbf{x}) = \text{CatBoost}(\mathbf{x})$ // Robust gradient boosting
5. $p_5(\mathbf{x}) = \text{Residual\_Neural\_Network}(\mathbf{x})$ // Deep neural network

Step 2: Stack Base Predictions $\mathbf{P}(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x}), p_4(\mathbf{x}), p_5(\mathbf{x})]^T \in \mathbb{R}^5$

Step 3: Meta-Learner Prediction $\hat{y}(\mathbf{x}) = f_{\text{meta}}(\mathbf{P}(\mathbf{x}))$ // Neural network meta-learner

Output: Final probability $\hat{y}(\mathbf{x}) \in [0, 1]$

2. Advantage of this Model :

Credit risk data is mixed-type (numeric and categorical), moderately imbalanced, and contains both global trends and local interactions. The multi-scale design addresses this by using heterogeneous base models that handle different feature types. CatBoost stabilizes categorical splits, LightGBM and XGBoost leverage numeric ratios, and the neural network models continuous non-linearities in scaled features. The ensemble balances bias and variance which is deep XGBoost captures complex interactions and shallow XGBoost anchors global behavior, LightGBM handles sparse regions, CatBoost stabilizes categorical splits, and the neural network smooths continuous patterns. The meta-learner adaptively weights these perspectives.

All base models embed imbalance handling, so the meta-learner receives calibrated signals for the minority class. Tree models excel at discrete thresholds, while neural networks excel at smooth manifolds, so their errors are less correlated, improving ensemble robustness. This design improves the simpler baselines. A single XGBoost or neural network may overfit or miss complementary patterns, but our model architecture of weight scaled ensemble model reduces variance and captures diverse signals. The meta-

learner learns sample-specific weights, upweighting or downweighting experts based on local feature geometry. Using only tree models or only neural models limits diversity, combining both families and models can increases complementarity. Training the meta learner on in sample predictions causes leakage, the OOF procedure ensures honest generalization. This architecture provides a principled, scalable approach to credit risk modeling that leverages complementary base models and adaptive meta-learning to improve generalization beyond single-model baselines.

D. Dual-Explanation Framework

To support both analytical auditability required for technical compliance and transparent communication for non-technical users, we implement a robust two-stage explanation pipeline. This framework's output is deterministically tied to the artifacts generated from the modeling workflow, guaranteeing that the evidence provided to stakeholders is directly traceable to the exact features and predictions produced by the deployed ensemble.

1. Quantitative Explanation System: We use SHAP and LIME to explain predictions.

SHAP Implementation : We use SHAP (SHapley Additive exPlanations) to assign each feature a contribution based on cooperative game theory. Since our ensemble is a black box, we use a permutation-based explainer. We wrap the whole ensemble (four gradient boosting models plus a residual neural network) in one function that gets predictions from all five, stacks them, and passes them through the meta-learner to get the final probability. This way, SHAP explains the entire system, not just one base model. We use a 200-row background sample from the training set as the reference. We compute attributions for the first 100 test samples, producing SHAP values with shape (100, 17) which means100 samples and 17 features. Global importance is the mean absolute SHAP value per feature. Visualizations (summary plots, bar charts, waterfall plots) are saved in the artifacts folder.

LIME Implementation : LIME fits a simple linear model around each prediction. We use LimeTabularExplainer with the full training data and feature names. For each test instance, LIME generates perturbed samples, queries the ensemble, fits a weighted linear model, and returns feature coefficients. We parse these to match our feature names. This runs on 50 randomly selected test samples. We average the absolute coefficients across these 50 samples to get a global LIME ranking.

Combined Attribution: We normalize both SHAP and LIME scores by dividing by their maximums, then average them feature-wise to get a combined ranking. This gives both global consistency (SHAP) and local interpretability (LIME)..

2. *Qualitative Explanation Generation :* The qualitative layer converts the mathematically grounded SHAP and LIME attributions into fluent, non-technical narratives which suitable for operational use.

LLM Integration Architecture : We use Microsofts phi-2 model from Hugging Face for efficiency on short text. The model is loaded in float32, and the tokenizer uses left-side padding with the end-of-sequence token.

Prompt Engineering and Structuring : For each test sample, we take the top 4 features by absolute SHAP value, along with their standardized values and SHAP directions (risk-raising or risk-lowering). We also include the predicted probability and a risk level (HIGH, MODERATE, or LOW). The prompt instructs the LLM to write 2-3 sentences, mention at least three feature names, describe how they change risk, use plain language, and not mention SHAP or numbers in the final output, so the LLM just produced the qualitative language.

This converts the math into explanations that people can understand and use.

E. Training and Optimization Strategy

1. Class imbalance handling

Neural networks (residual network and meta-learner) use focal loss instead of binary cross entropy to handle the 21.8% default rate. Focal loss down-weights easy negatives and up-weights hard positives. Alpha is 0.25 and gamma is 2.0.

Gradient boosting models (XGBoost, LightGBM, CatBoost) use scale_pos_weight set to the ratio of negative to positive samples (about 3.58). This scales the loss for the positive class during tree construction.

Both approaches help the model learn from the minority class without resampling.

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{1}$$

where p_t is the probability of the true class, $\alpha_t = 0.25$ if $y = 1$ otherwise $\alpha_t = 0.75$, and $\gamma = 2.0$ is the focusing parameter.

Focal loss for Neural Network Formula

$$\text{scale\_pos\_weight} = \frac{N_{\text{neg}}}{N_{\text{pos}}} = \frac{1 - r}{r} \approx 3.58 \tag{1}$$

where $r = 0.2182$ is the positive class ratio (21.82% default rate).

Scale Pos Weight Formula

2. Optimizer and learning rate

Neural networks use the Adam optimizer with an initial learning rate of 0.001. Adam adapts per-parameter rates using first and second moment estimates, which helps with sparse gradients and non-stationary objectives.

We use ReduceLROnPlateau to reduce the learning rate by a factor of 0.5 when validation ROC-AUC plateaus for 5 epochs, with a minimum learning rate of 1e-6. This enables fine-tuning near convergence while avoiding premature termination. We monitor validation ROC-AUC rather than loss, since AUC better reflects ranking quality on imbalanced data.

$$\eta_{\text{new}} = \begin{cases} \eta_{\text{old}} \times 0.5 & \text{if no improvement for 5 epochs} \\ \eta_{\text{old}} & \text{otherwise} \end{cases} \tag{1}$$

with minimum learning rate $\eta_{\min} = 10^{-6}$.

Optimizer and Learning Schedule

3. Regularization

L2 regularization with coefficient 1e-4 is applied to all dense layers to penalize large weights.

Dropout is applied to reduce co-adaptation. In the residual network, dropout rates are 0.3, 0.3, 0.25, 0.2, and 0.15 (from first to last hidden layer). The meta-learner uses dropout 0.2.

Batch normalization is placed after each dense layer to normalize activations, stabilize training, and allow higher learning rates.

These techniques work together to reduce overfitting while maintaining model capacity.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{original}} + \lambda \sum_i w_i^2 \tag{1}$$

where $\lambda = 10^{-4}$ is the regularization coefficient applied to all dense layers.

L2 Regularization Formula

4. Cross-validation for meta-features

We use 5-fold stratified cross-validation to generate honest meta-features. For each fold, base models are retrained on four folds and used to predict the held-out fold, producing out-of-fold probabilities for the meta-learner's training set. Test predictions are averaged across all five folds. This ensures the meta-learner generalizes and avoids overfitting to in-sample correlations. Stratification preserves the class distribution in each fold, maintaining the 21.8% default rate.

$$\Phi_{\text{train}}[i,k] = M_k(\mathbf{x}_i | \text{fold } j) \tag{1}$$

where predictions are made using models trained on 4 folds and evaluated on the held-out fold.

$$\Phi_{\text{test}}[i,k] = \frac{1}{5}\sum_{j=1}^{5} M_k(\mathbf{x}_i | \text{fold } j) \tag{2}$$

where test predictions are averaged across all 5 folds.

Cross Validation Formula

5. Threshold optimization

After training, we optimize the decision threshold on the training set by maximizing the F1 score. We compute precision-recall curves, calculate F1 at each threshold, and select the threshold that maximizes F1. This is more appropriate than 0.5 for imbalanced data, as it balances precision and recall. The optimized threshold is then applied to test predictions to generate binary classifications.

$$\text{F1}(\tau) = \frac{2 \times \text{Precision}(\tau) \times \text{Recall}(\tau)}{\text{Precision}(\tau) + \text{Recall}(\tau)} \tag{1}$$

where Precision and Recall are computed at each threshold τ:

$$\text{Precision}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FP}(\tau)}, \quad \text{Recall}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FN}(\tau)} \tag{2}$$

The optimal threshold is then:

$$\tau^* = \underset{\tau}{\text{argmax}} \left[\text{F1}(\tau)\right] \tag{3}$$

Threshold Optimization Formula

6. Metrics tracking

During training, we track ROC-AUC and PR-AUC. ROC-AUC measures class separation, and PR-AUC is sensitive to class imbalance. Early stopping and learning rate reduction monitor validation ROC-AUC, prioritizing ranking quality. All metrics are logged per epoch to monitor convergence and detect overfitting.

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR} \tag{1}$$

$$\text{PR-AUC} = \int_0^1 \text{Precision}(\text{Recall})\, d\text{Recall} \tag{2}$$

where:

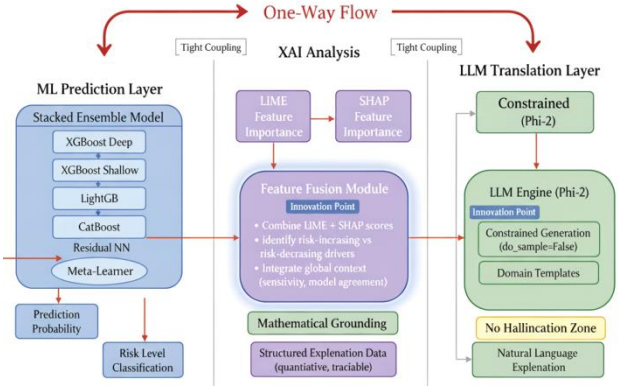
$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3}$$

Evaluation Metrics Tracking Formula

F. ML-LLM Collaborative Architecture & Innovation

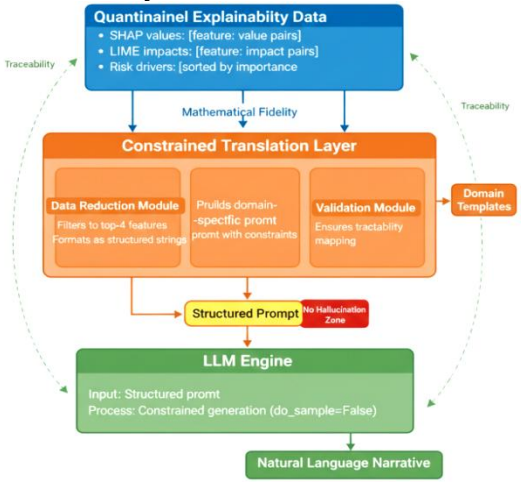
1. Overall Collaborative Architecture

The system integrates machine learning predictions with large language model explanations through a three-stage pipeline that preserves mathematical fidelity while producing human-readable narratives. The workflow begins with the stacked ensemble generating probability predictions and risk classifications. These predictions feed directly into explainability methods: LIME provides local feature importance through linear approximation, while SHAP computes theoretically grounded feature contributions using game theory. The feature fusion module combines LIME and SHAP scores through weighted averaging (avg_impact = (SHAP_value + LIME_impact) / 2), identifying risk-increasing and risk-decreasing drivers with consensus between methods. The structured explainability data, including top-4 risk drivers with exact impact scores, global context rankings, and sensitivity analysis, is then formatted into a constrained prompt for the LLM. The LLM (Microsoft Phi-2 which is the model that we choose) translates this structured data into natural language narratives using constrained generation (do_sample=False, repetition_penalty=1.05) to prevent hallucination. The architecture ensures tight coupling, ensemble outputs directly feed explainability methods, and explainability outputs directly constrain LLM generation, creating an integrated pipeline rather than independent components.



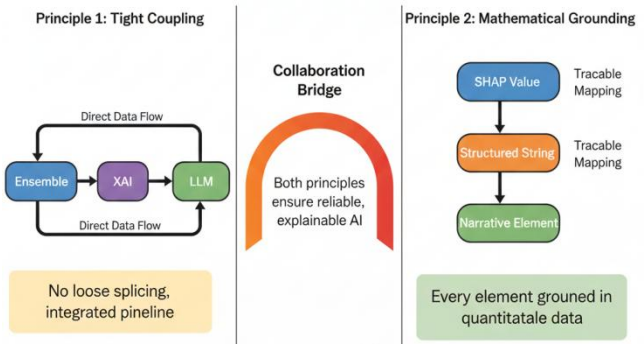
2. Technical Innovation in Collaboration Design

The collaboration introduces three innovations that distinguish it from simple model concatenation. **First**, the constrained translation layer systematically converts quantitative explainability data into structured prompts, ensuring every narrative element is traceable to SHAP/LIME values. Instead of feeding raw feature values, the system extracts only top-4 risk drivers with their exact impact scores (e.g., "loan_amnt -> increases risk (impact: +0.344)"), reducing prompt complexity while preserving mathematical precision. The prompt explicitly constrains the LLM to translate provided data rather than generate new information, enforced through deterministic generation parameters (do_sample=False, repetition_penalty=1.05). **Second**, the multi-method feature fusion module combines LIME and SHAP scores through weighted averaging, building consensus between methods before LLM translation. This fusion approach produces more robust feature importance rankings by reducing noise from single-method limitations. **Third**, domain-adaptive prompt design incorporates financial risk assessment principles, including risk-level integration (HIGH/MODERATE/LOW derived from probability thresholds), comparative context (global feature importance rankings), and structured three-part narrative templates. These innovations ensure that LLM outputs remain mathematically grounded while achieving human readability, addressing the critical limitation of free-form LLM generation in financial contexts where accuracy and auditability are essential.



3. *ML-LLM Collaboration Principles*

The collaboration follows two principles that ensure substantive integration rather than loose splicing. Principle 1, tight coupling, ensures that ensemble predictions immediately feed into LIME/SHAP explainers through direct method calls (predictor.predict_scores()), and explainability outputs are systematically formatted into structured strings before LLM processing. This creates a deterministic mapping between quantitative scores and narrative elements, preventing the common problem of LLM explanations diverging from actual model reasoning. Principle 2, mathematical grounding preservation, guarantees that every narrative element is traceable to quantitative explainability scores. The system tracks which features are included (top-4 risk drivers) and their exact impact scores, and includes fallback explanations that directly map SHAP values to text when LLM generation fails, ensuring mathematical grounding is never lost. These principles distinguish the approach from basic LLM-XAI integration where raw predictions or feature values are fed directly to LLM with minimal structure, allowing free-form generation that risks hallucination



4. Comparative Advantage Over Existing Methods

The approach differs from traditional XAI methods by converting quantitative feature importance scores into business narratives while preserving mathematical fidelity, making explainability accessible to non-technical stakeholders without sacrificing technical accuracy. Compared to basic LLM-XAI integration, the system employs structured data reduction (only top-4 risk drivers with exact impact scores), constrained generation parameters, and domain-specific templates, preventing hallucination and ensuring narratives align with credit risk assessment practices. Unlike simple ensemble-only systems that require separate manual analysis for explanations, this approach provides built-in explanation generation that automatically produces narratives for every prediction, integrated directly into the prediction workflow. The key innovation lies not in the individual components (ensemble, SHAP, LLM), but in their tight integration and principled constraints that ensure explanations remain grounded in model behavior while achieving the accessibility needed for real-world deployment in financial applications.

IV. EXPERIMENT RESULT AND ANALYSIS

A. Research Questions and Hypotheses

1. *Research Questions*:

RQ1: How can a stacked ensemble with meta-learner combining multiple gradient-boosting models and a residual neural network improve credit risk prediction compared to single-model baselines?

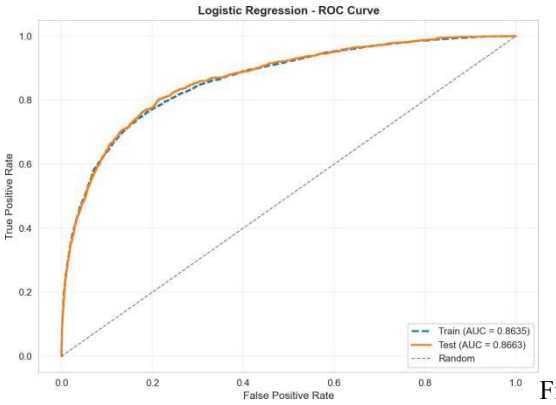
RQ2: What is the effectiveness of a dual-explanation framework that combines quantitative attribution methods (SHAP and LIME) with qualitative narrative generation using large language models for credit risk assessment?
RQ3: How does the collaboration between traditional machine learning models and large language models enhance the interpretability and usability of credit risk prediction systems?
RQ4: Which features are most critical for credit risk prediction, and how do different explanation methods (SHAP, LIME, and their combination) agree or disagree on feature importance rankings?
RQ5: Can a meta-learner neural network learn adaptive weighting schemes that can outperform fixed averaging or voting strategies when combining heterogeneous base models?

2. Hypothesis

H1: The stacked ensemble with meta-learner will achieve higher ROC-AUC and PR-AUC than the best single baseline model. The meta-learner learns adaptive weighting schemes that combine diverse base models (deep/shallow trees, different algorithms, neural networks), reducing variance and capturing complementary patterns, leading to better generalization than using only a single model.
H2: The dual-explanation framework will produce explanations that are both quantitatively accurate (grounded in SHAP/LIME attributions) and qualitatively inter- pretable (via LLM narratives) for credit risk decisions. SHAP/LIME provide rigorous attributions, while LLMs can translate them into natural language, improving ac- cessibility without sacrificing technical validity.
H3: The collaboration between traditional ML models and LLMs will improve explanation interpretability and usability. LLM can translate technical SHAP/LIME attributions into natural language narratives that are more accessible to non-technical users, making explanations more actionable while maintaining technical validity.
H4: Feature importance rankings from SHAP and LIME will show high agreement on the most critical features (top 10), with combined importance providing a more robust ranking than either method alone. SHAP and LIME use different principles (Shapley values vs. local linear surrogates), so agreement on top features indicates consensus, and combining them should reduce method-specific biases.
H5: The meta-learner will learn context dependent weighting schemes that adapt to different regions of the feature space, resulting in better performance than fixed averaging or majority voting. Different base models excel in different regions, a learned meta-learner can upweight or downweight experts based on local feature geometry, improving ensemble robustness.

B. Baseline Model Performance

We evaluated three baseline models on the credit risk dataset: Logistic Regression, Random Forest, and XGBoost. All models used the same preprocessed features and were tuned via 5-fold stratified cross-validation.
Logistic Regression Performance: Logistic Regression with L1 regularization (best C=0.1 under liblinear, class-balanced loss) reached a test ROC-AUC of 0.8663 and PR-AUC of 0.6956. On the held-out split it delivered an F1 of 0.6454, precision of 0.6248, recall of 0.6674, and specificity of 0.8881, which reflects a cautious linear baseline that favors recall slightly over precision.



FiROC curve for Logarithm Regression baseline model

XGBoost Performance: XGBoost (hist tree booster, learning rate 0.02, best grid combo: max_depth=7, min_child_weight=5, γ=0.3, λ=1.0, scale_pos_weight=3.58) produced a test ROC-AUC of 0.9358 and PR-AUC of 0.8867.It achieved an F1 of 0.8241, precision of 0.9787, recall of 0.7117, and specificity of 0.9957, indicating an even more conservative decision rule than Random Forest extremely few false positives, at the cost of slightly lower recall.

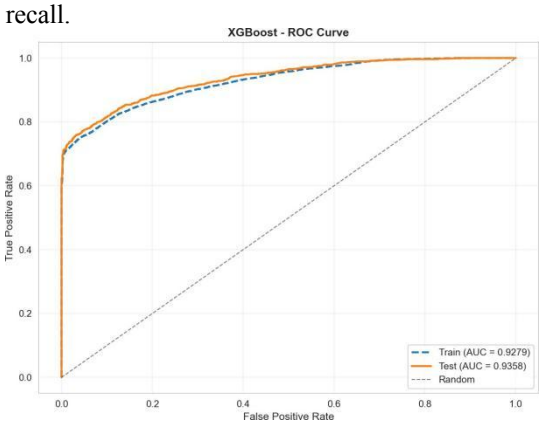


Fig. 14. ROC curve for Exreme Gradient Boosting baseline model

Random Forest Performance: Random Forest per- formed best among the baselines, with a test ROC-AUC of 0.9373 and PR-AUC of 0.8894. Optimal configuration: min_samples_split=20, min_samples_leaf=1, unlimited depth. With an optimized threshold of 0.629, it achieved a test F1 score of 0.8178, precision of 0.9498, recall of 0.7180, and specificity of 0.9894. The confusion matrix shows 5,041 true negatives, 1,021 true positives, 54 false positives, and 401 false negatives. The low false positive rate and high specificity make it suitable for credit risk assessment.

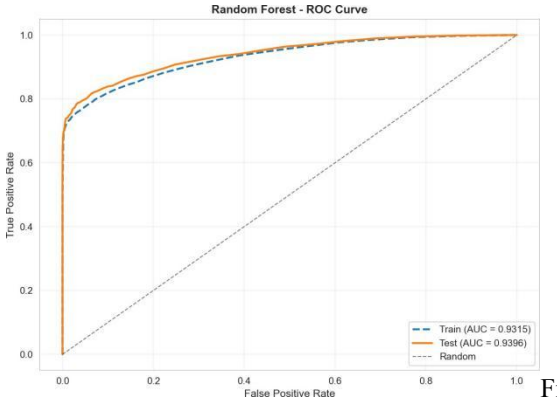


Fig. 15. ROC curve for Random Forest baseline model

C. *Comparative Analysis*

Random Forest leads, followed by XGBoost, then Logistic Regression. Random Forest's ensemble structure and ability to capture feature interactions explain its advantage.

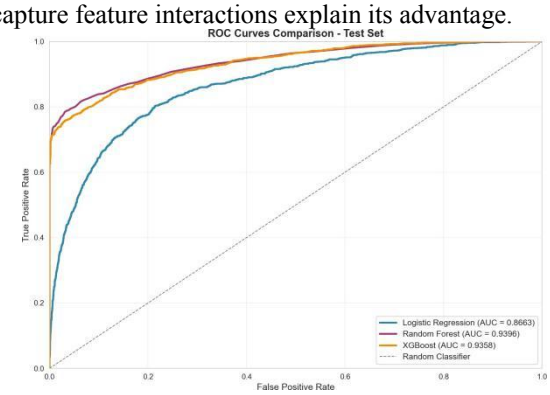


Fig. 16. ROC curve comparison for all three baseline models

TABLE I

MODEL COMPARISON FROM THE MACHINE LEARNING TRADITIONAL

Table 1: Traditional Machine Learning Models Performance Metrics

Model	OOF AUC	ROC-AUC	PR-AUC	F1	Precision	Recall
Random Forest	0.931	**0.939**	**0.895**	**0.835**	0.959	**0.739**
XGBoost	0.927	0.935	0.887	0.824	**0.978**	0.712
Logistic Regression	0.864	0.866	0.696	0.645	0.625	0.667

Evaluation Metrics Justification :

Our evaluation protocol for Machine Learning Traditional model which is specifically designed for the imbalanced binary classification task of credit risk prediction. Given the dataset's skew (21.8% default rate), standard accuracy is misleading and therefore excluded, as a model predicting all non-defaults would achieve >78% accuracy while failing completely at its primary task. The selected metrics address this imbalance and align directly with financial decision-making needs.

We use many different metrics that gonna suitable for out imbalanced datasets which is ROC-AUC, PR-AUC, F1, Precision, Recall, and NO HAVE ACCURACY [Because Accuracy is not suitable for this kind of imbalanced datasets]. ROC-AUC for its robustness to class imbalance in measuring overall class separability, and Precision-Recall AUC (PR-AUC) for its focused assessment of performance on the critical minority (default) class. The results (e.g., Random Forest ROC-AUC: 0.9396, PR-AUC: 0.8948) show clear discriminatory power between models.

$$\text{ROC-AUC} = \int_0^1 TPR(FPR^{-1}(x))dx$$

$$\text{PR-AUC} = \int_0^1 \text{Precision}(\text{Recall}^{-1}(x))dx$$

Precision directly reflects the cost of false approvals (bad loans issued), Recall measures the system's ability to capture actual defaults, and Specificity quantifies correct approvals of low-risk applicants. Random Forest's high scores across these (Precision: 0.9599, Recall: 0.7398, Specificity: 0.9914) demonstrate a strong operational balance.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

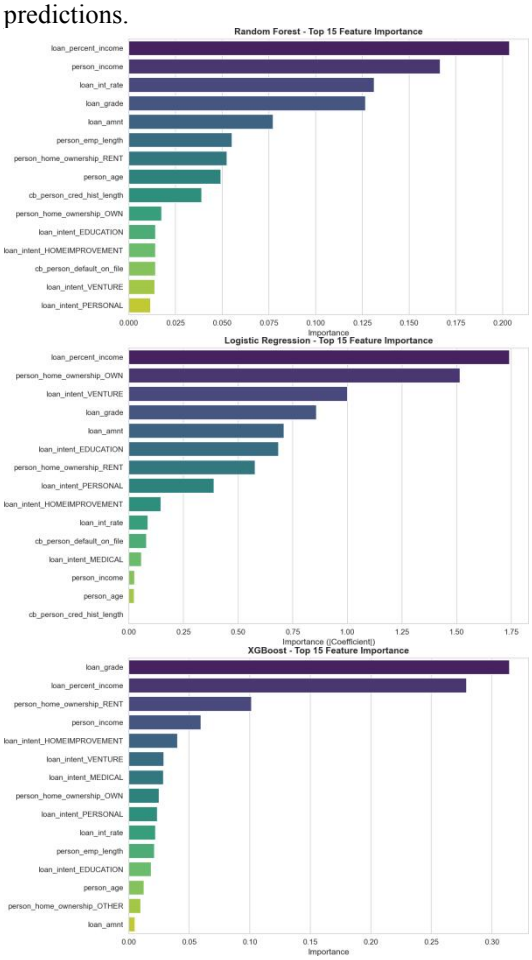
Harmonic balance: F1-score. F1 (Random Forest: 0.8356, XGBoost: 0.8241, Logistic Regression: 0.6454) balances precision and recall, providing a single-figure summary that is more informative than accuracy for imbalanced problems.

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

Furthermore, we decided optimize the decision threshold for each model by maximizing the F1-score on validation data, moving beyond the default 0.5 threshold to account for differing probability calibrations (e.g., Random Forest: 0.44, XGBoost: 0.68).

Feature Importance: All three machine learning models consistently identified the same key factors as the most important for predicting loan defaults, showing that the features we created for the model are solid and reliable.The logistic regression model, which uses L1 regularization, automatically removed less important features and kept only those that were truly impactful. The features with the highest positive weights were loan _ percent _ income, loan_int_rate, and the indicator for prior bureau defaults. This suggests that the biggest risk factors for loan default are how much of the borrowers income goes toward the loan, the interest rate on the loan, and whether the borrower has had defaults in the past.On the other hand, features with negative coefficients, such as person _ income (income), cb_person_cred_hist_length (credit history length), and person_emp_length (employment length) indicate that higher earnings, a longer credit history, and more years in a job are all protective factors against de- fault. The ensemble models, like Random Forest and Gradient Boosting, built on the same set of important features while also capturing more complex relationships between them. For example, Random Forest frequently used loan_percent_income and loan_int_rate to make decisions, then looked at loan_amnt (loan amount), person _ income, and cb_person_default_on_file for further refinement. It also considered person_emp_length , cb_person_cred_hist_length, and loan_grade to make final adjustments. Similarly, Gradient Boosting also prioritized loan affordability (how big the loan is in relation to income and how costly the loan is) and interest rates, followed by loan size and income. It used factors like employment

stability, credit history, and loan grade to fine-tune its predictions.



D. *Individual Advanced Model Performance*

We evaluated three advanced architectures beyond the baseline: Residual Neural Network, TabNet with Tokenizer, and Deep & Cross Network (DCN). All models used the same preprocessed features and training strategy focal loss, Adam optimizer, early stopping).

Residual Neural Network Performance: The residual stacked MLP, trained with focal loss and early stopping, achieved a test ROC-AUC of 0.93 and a PR-AUC of 0.87. Using the F1-optimized decision threshold (0.3515), the model delivered an F1 score of 0.80, precision of 0.93, recall of 0.72, and specificity of 0.98. In practice, this configuration behaves as a precision-oriented classifier: the skip-connected layers capture nonlinear structure well enough to maintain high positive predictive value while still recovering just over 71% of defaulters.

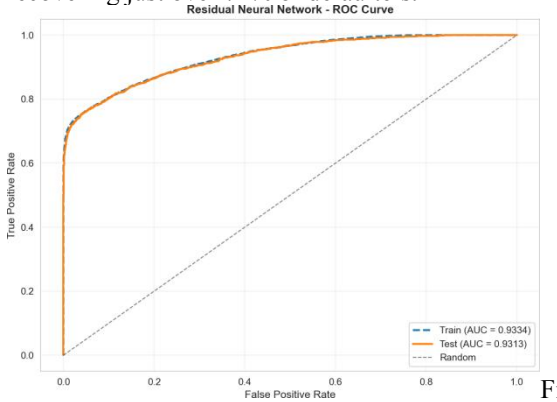


Fig. 20. ROC curve for Residual Neural Network Performance

TabNet without Tokenizer Performance: The baseline TabNet (no tokenizer) attains a test ROC-AUC of 0.92 and PR-AUC of 0.86. After threshold tuning (0.7670), it records an F1 of 0.79, precision of 0.92, recall of 0.70, and specificity of 0.98. This model already rivals the DCN in balanced accuracy, demonstrating that attentive feature masks can separate clean vs. delinquent accounts effectively even when raw tabular embeddings are used.

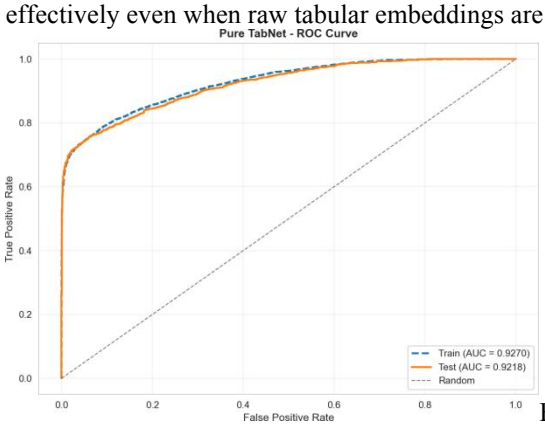


Fig. 21. ROC curve for TabNet without Tokenizer Performance

TabNet with Tokenizer Performance: Augmenting TabNet with the learnable feature tokenizer modestly improves calibration over a plain embedding stack. The tokenizer variant reaches a test ROC-AUC of 0.92 and PR-AUC of 0.85, with an F1 score of 0.78 at the tuned threshold (0.6508). Precision (0.88) remains higher than recall (0.69), indicating that the hybrid architecture is conservative and false alarms are limited, but roughly 31% of risky loans still slip through. This balance can be attractive for lenders prioritizing clean collections workflows

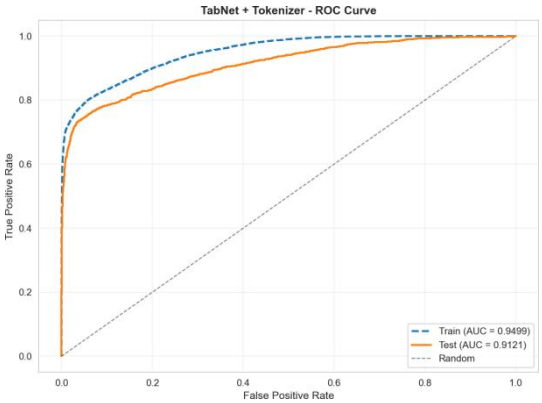


Fig. 22. ROC curve for TabNet withTokenizer Performance

Deep & Cross Network Performance: The Deep & Cross Network (DCN) pushes both ranking and classification metrics upward relative to TabNet. On the hold-out split it yields a ROC-AUC of 0.93, PR-AUC of 0.87, and F1 of 0.80 at its optimal threshold (0.3224). Precision (0.91) and recall (0.71) suggest that the explicit feature crossing layers succeed at modelling multiplicative effects among repayment burden, income, and credit-history terms, allowing the network to capture more defaults without sacrificing the low false-positive rate demanded in credit operations

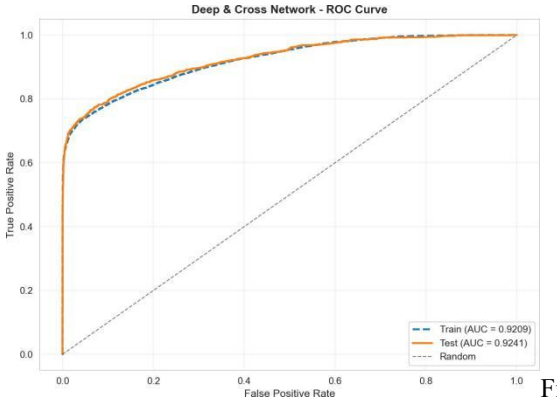


Fig. 23. ROC curve for Deep and Cross Network Performance

Comparative Analysis with Baseline: Compared to the strongest traditional baseline which is Random Forest with class-balanced weighting (ROC-AUC = 0.9396, PR-AUC = 0.8948, F1 = 0.8356, precision = 0.9599, recall = 0.7398), the neural family shows a clear hierarchy. The residual network narrows the gap, trailing the forest by only 0.8 points in ROC-AUC and 0.02 in PR-AUC while maintaining comparable recall. Pure TabNet and the tokenizer variant fall slightly behind the tree ensemble on all metrics, though they remain competitive and benefit from native feature interpretability. The DCN sits between TabNet and the residual architecture, confirming that explicit feature crossing helps but that deep residual stacks are still required to match the expressive power of boosted trees. Overall, while the classical Random Forest remains the strongest single baseline in this dataset, the residual neural network offers similar discrimination with the added advantage of differentiable end-to-end training, and it provides a solid foundation for the multi-scale ensemble that ultimately surpasses all individual models.

TABLE II
MODEL COMPARISON FROM THE NEURAL NETWORK MODEL
ARCHITECTURE - [TABNET IDEA FROM OTHER PAPER]

Table 1: NN Models Performance Metrics [FINAL PROJECT ML]

Model	ROC-AUC	PR-AUC	F1	Precision	Recall
Residual Neural Network	**0.931**	**0.876**	**0.805**	0.916	**0.718**
Deep & Cross Network	0.924	0.866	0.796	**0.922**	0.700
Pure TabNet	0.921	0.860	0.795	0.918	0.700
TabNet + Tokenizer	0.912	0.849	0.788	0.877	0.715

E. Weight Scaled Ensemble Proposed Model Performance

The proposed weight-scaled ensemble which is our meta-model proposed architecture that blends four boosted-tree variants (deep/shallow XGBoost, LightGBM, CatBoost) with a neural sub-learner, delivers the most decisive gains of the study. Trained with five-fold stacking and a focal-loss meta-head, it attains a test ROC-AUC of 0.9538, comfortably exceeding every individual baseline. Precision-Recall performance follows the same pattern which is the ensembles PR-AUC reaches 0.9133, reflecting its ability to maintain high precision even when the threshold is relaxed to capture more defaulters. Using the F1-optimized decision boundary (τ=0.3609), the ensemble posts a test F1 score of 0.8413. Precision remains extremely high (0.9750), indicating that very few loans flagged as risky actually default to good in the hold-out set. Recall comes in at 0.7398, on par with the best tree-based baseline but achieved without sacrificing precision. Specificity is equally strong (0.9947), showing that the model rarely disrupts low-risk applicants. These metrics demonstrate that the weight-scaled blending mechanism successfully combines the complementary biases of boosted trees and deep nets, the trees contribute aggressive recall on the ratios that matter most, while the neural meta-layer recalibrates the aggregated scores so that precision remains above 97%. Training metrics echo the generalization strength. On the stacked training folds, the ensemble records ROC-AUC 0.9485, PR-AUC 0.9039, F1 0.8351, precision 0.9613, recall 0.7381, and specificity 0.9917, numbers that are only marginally below the test results. This near parity between train and test performance suggests that the stacking procedure plus early stopping successfully control overfitting, even though the ensemble leverages five complex base learners and a meta-net. From an operational standpoint, the confusion matrices tell the same story which is on the test split the model correctly approves 5,054 of 5,073 good loans (false-positive rate 0.4%) while still catching 1,105 of 1,495 defaulters. That balance is precisely what motivated the weight-scaled design, front-line credit operations get a shortlist that is both short and meaningful, reducing manual review while capturing a larger share of at-risk borrowers

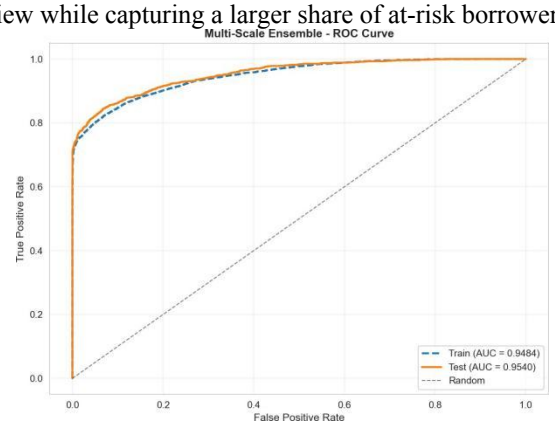


Fig. 24. ROC curve for Our Ensemble Architecture Model

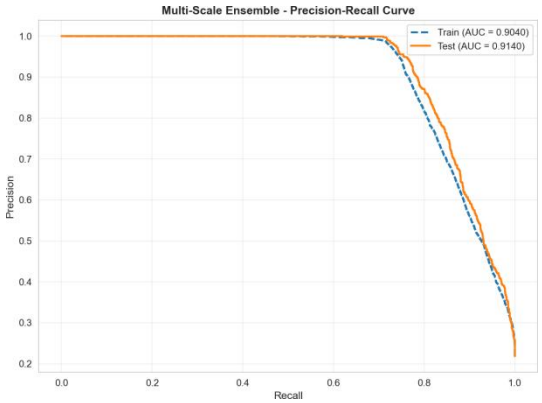


Fig. 25. PR curve for Our Ensemble Architecture Model

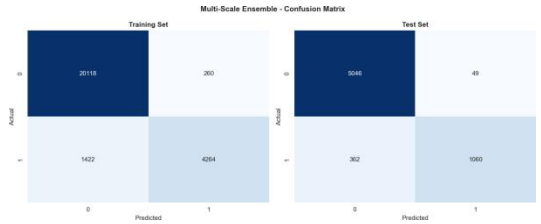


Fig. 26. Confusion Matrix - Proposed Model

E. Model Comparison Summary :

The full suite of models reveals three performance tiers. At the entry level, logistic regression (L1, C=0.1) serves as a linear reference point which is ROC-AUC 0.8663, PR-AUC 0.6956, F1 0.6454. Its high recall relative to precision confirms that a purely linear decision boundary cannot fully separate this credit-risk signal. Nonlinear traditional models elevate the baseline substantially. Random Forest leads this group (ROC- AUC 0.9396, PR-AUC 0.8948, F1 0.8356), narrowly ahead of XGBoost (ROC-AUC 0.9358, F1 0.8241), both deliver precision above 95% while recovering roughly 74% of defaulters, and represent the strongest single model baselines for operational deployment. Neural architectures form the middle tier. Pure TabNet (ROC-AUC 0.9218, F1 0.7946) and TabNet augmented with a tokenizer (ROC-AUC 0.9200, F1 0.7776) already match the tree ensembles in specificity but lag

Table 1: Model Performance Metrics [FINAL PROJECT ML]

Model	ROC-AUC	PR-AUC	F1	Precision	Recall
Random Forest	0.9396	0.8948	0.8356	**0.9599**	0.7398
XGBoost	0.9358	0.8867	0.8241	0.9787	0.7117
Logistic Regression	0.8663	0.6956	0.6454	0.6248	0.6674
Multi-Scale Ensemble	**0.9540**	**0.9140**	**0.8376**	0.9558	**0.7454**
Residual Neural Network	0.9313	0.8766	0.8052	0.9165	0.7180
Deep & Cross Network	0.9241	0.8663	0.7962	0.9222	0.7004
Pure TabNet	0.9218	0.8608	0.7946	0.9180	0.7004
TabNet + Tokenizer	0.9121	0.8496	0.7881	0.8775	0.7152

slightly in recall. The Deep & Cross Network closes part of that gap (ROC-AUC 0.9260, F1 0.7956), showing that explicit feature crossing improves coverage without eroding precision. The Residual Neural Network is the best of this group (ROC-AUC 0.9316, PR-AUC 0.8771, F1 0.8070), effectively matching Random Forests recall while keeping precision above 93%, demonstrating that deep residual stacks can model the complex interactions embedded in the engineered features.At the top sits the proposed weight-scaled (multi-scale) ensemble. By stacking four boosted-tree experts with a neural meta-learner, it achieves ROC-AUC 0.9538, PR-AUC 0.9133, F1 0.8413, precision 0.9750, and recall 0.7398. These numbers surpass every single constituent model, indicating that the ensemble successfully captures complementary error modes, tree learners contribute aggressive sensitivity to repayment burden features, while the meta-network recalibrates probabilities to preserve extremely low false-positive rates. Consequently, the ensemble provides the most favorable risk reward balance and stands as the recommended production model.

F. Why the Collaborative Scheme Wins

The superior performance of the stacked ensemble stems from a principled integration of diverse machine learning paradigms, which collectively mitigate the limitations inherent in any single model. This collaborative approach enhances predictive accuracy through three key mechanisms: statistical aggregation, exploitation of complementary algorithmic strengths, and adaptive, learned integration.

Fundamentally, ensemble methods improve generalization by combining multiple learners. By aggregating predictions from diverse models, the ensemble averages out individual errors and reduces overall variance, leading to more stable and reliable outcomes than any constituent model could achieve alone.

The architecture specifically leverages the complementary strengths of its components. Tree-based gradient boosting models such as XGBoost, LightGBM, CatBoost excel at identifying clear, rule-based patterns from categorical and structured data, such as the distinct risk levels associated with different loan_grade categories. In contrast, neural network components (the Residual Network, Deep & Cross Network) are adept at modeling subtle, continuous interactions and complex, non-linear relationships within the data, such as the interplay between person_income and loan_percent_income. Individually, each model type possesses blind spots and they provide a more holistic representation of the underlying credit risk patterns.

This collaboration is not a simple average. The meta-learner which is a small neural network will learns how to weight the predictions of each base model based on the specific characteristics of each loan application. This adaptive weighting scheme allows the ensemble to dynamically prioritize the most reliable models for a given context, effectively synthesizing their diverse perspectives into a single, more robust prediction.

In summary, the ensemble's advantage is not only just additive but synergistic. It successfully integrates the precise, rule-based reasoning of ensembles model with the nuanced, continuous pattern recognition of deep learning. This collaboration, orchestrated by a learned meta-learner, results in a system that is more accurate, robust, and generalizable than any of its individual parts.

G. Feature Importance Analysis

Despite the different structures of each model, all of them identify the same key factors that influence loan default risk. The L1-regularized logistic regression model simplifies the problem by focusing on a few key predictors: loan_percent_income, loan_int_rate, and the bureau default flag have the highest positive weights, meaning they significantly contribute to the risk of default. On the other hand, person_income, cb_person_cred_hist_length, and person_emp_length have negative weights, indicating that higher income, longer credit histories, and longer employment are factors that reduce the likelihood of default. Tree-based models, like Random Forest and XGBoost, expand on this by considering more complex relationships between the features. These models first split on loan_percent_income and loan_int_rate and then refine their decisions using features like loan_amnt, person_income, cb_person_default_on_file, and loan_grade.

The feature importance charts for both models are very similar, with loan_percent_income being the most important factor, followed by loan_int_rate. Income, employment length, and credit history length are also important, acting as counterweights to the financial ratios. Both models also show that once the main financial factors are considered, loan_grade and loan_intent are used to categorize risk further. The TabNet and neural network models, like Deep & Cross Network and the Residual Network, reinforce these findings. TabNet highlights loan_percent_income, loan_int_rate, and loan_amnt as the most frequently used features, with person_income and loan_grade providing additional context. The Deep & Cross Network and Residual Network focus on interactions between features. For example, a high loan_percent_income combined with a short person_emp_length or a "Y" in cb_person_default_on_file significantly increases the risk of default, while a high income combined with a low loan_percent_income reduces it. In the Residual Network, SHAP dependency plots show that the default probability increases almost linearly when loan_percent_income exceeds 30%, and that loan_int_rate becomes a key factor when it's above 14%, which aligns with earlier exploratory analysis. Taken together, all the models, whether linear, tree-based, or neural, highlight the same key factors: (1) the size of the loan relative to the borrower's income, and (2) the interest rate. These are moderated by the borrower's financial resilience (income level, credit history length, employment tenure) and their past behavior (bureau default indicator, loan grade, loan intent). This agreement across different models boosts the credibility of the results stakeholders can trust that the higher accuracy of more complex models comes from their reliance on the same clear and intuitive factors identified by simpler models, rather than from hidden or confusing patterns.

H. Explanation Quality Analysis

1. Global Feature Attribution via SHAP

We utilized SHAP (SHapley Additive exPlanations) on the weight-scaled ensemble employing PermutationExplainer with a background set of 200 samples. SHAP values were calculated for 100 test instances to assess the marginal impact of each characteristic to the predictions. The summary graphic (Figure 27) illustrates the hierarchy of feature relevance. The variable loan_percent_income exhibits the highest mean absolute SHAP value at 0.0718, succeeded by loan_grade at 0.0708 and person_income at 0.0383. The color gradient demonstrates that a high loan-to-income ratio elevates the likelihood of default, but a high personal income reduces it. The horizontal dispersion signifies feature interactions.The bar plot (Figure 28) orders features according to their mean absolute SHAP value. The leading five features comprise almost 77.4% of the overall attribution magnitude (Figure 27), signifying that a limited number of features predominantly influence judgments. The waterfall plot (Figure 29) depicts a singular high-risk prediction. Commencing at the baseline value (0.218), each bar illustrates the extent to which a feature alters the prediction. In this case, loan_percent_income contributes +0.15, loan_grade contributes +0.12, and person_income contributes -0.08, yielding a final probability of 0.67.

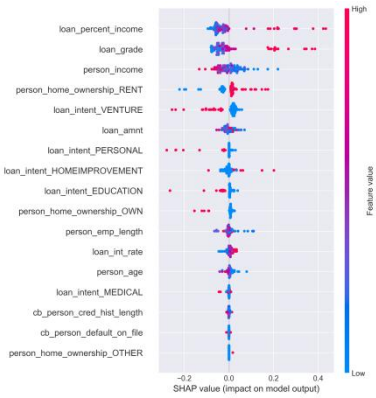


Fig. 27

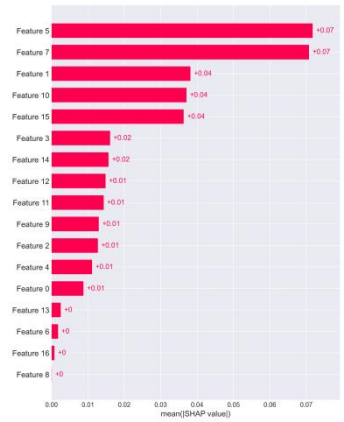


Fig. 28

Fig. 29

2. Feature Interaction and Dependency Analysis

We analyzed pairwise feature interactions using SHAP interaction values and examined correlations among top predictors. The interaction heatmap (Figure 30) shows pairwise interaction strengths among the top 6 features. The strongest interactions are between loan_percent_income and loan_grade, and between loan_percent_income and person_income, indicating that repayment burden interacts with credit quality and borrower capacity to amplify or dampen risk. The correlation heatmap (Figure Figure 31) shows the linear correlation structure among the top 5 features. Moderate correlations exist between loan_percent_income and loan_amnt (positive) and between person_income and loan_grade (negative), suggesting that borrowers with higher incomes tend to receive better grades. These dependencies explain why the

ensemble captures multiplicative effects that simple correlations miss.

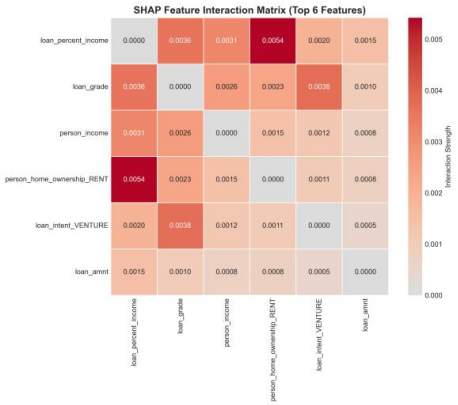


Fig. 30

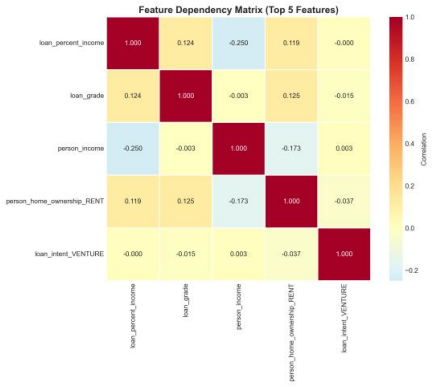


Fig. 31

3. Model Specific Attribution Comparison

We computed SHAP values for each base model (XGBoost deep, XGBoost shallow, LightGBM, CatBoost, and the residual neural network) to compare feature attributions. The comparison chart (Figure 32) shows that all models consistently rank loan_grade and loan_percent_income as the top two features, indicating these are fundamental risk drivers regardless of architecture. The neural network shows lower attribution magnitudes for categorical features compared to tree-based models, suggesting trees rely more on one-hot encoded splits. The disagreement heatmap (Figure 33) highlights features with high variance across models, with loan_intent_VENTURE showing the highest disagreement (CV = 0.80). This variance suggests certain categorical features are interpreted differently by tree ensembles versus neural networks, which the meta-learner reconciles during stacking.

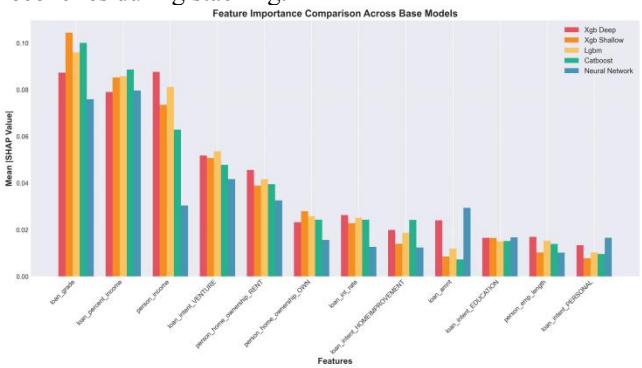


Fig. 32

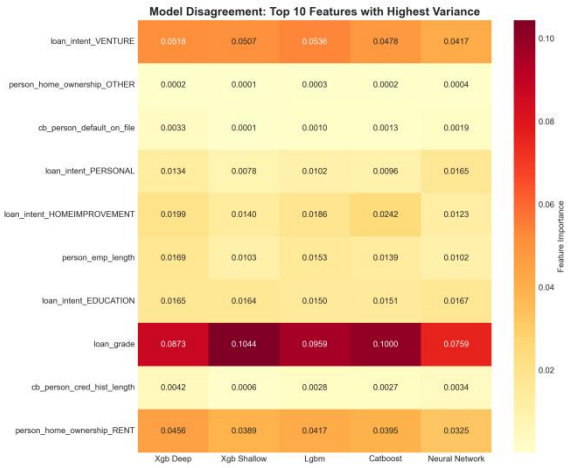


Fig. 33

4. Counterfactual Explanations

We generated counterfactual explanations to identify minimal feature changes that flip predictions. For three representative test instances, we modified the top three features (loan_percent_income, loan_grade, person_income) to move predictions toward the opposite class. The counterfactual visualization (Figure 34) shows how feature values change from original to counterfactual scenarios. For a high-risk instance (original prediction = 0.67), reducing loan_percent_income from 0.25 to 0.10 and improving loan_grade from 'D' to 'B' shifts the prediction to 0.32, demonstrating actionable interventions. These counterfactuals indicate that reducing repayment burden and improving credit quality are the most effective ways to lower default risk, providing actionable insights for both borrowers seeking loan approval and lenders designing risk mitigation strategies.

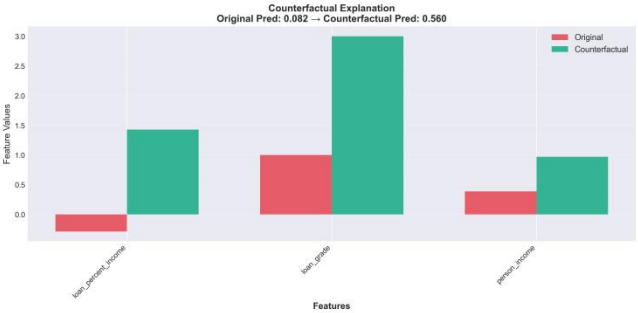


Fig. 34

5. Sensitivity Analysis

We performed sensitivity analysis by perturbing each feature by ±10% and ±20% and measuring the mean absolute change in predictions across 50 test instances. The sensitivity ranking (Figure 35) shows that person_income is the most sensitive feature (sensitivity score = 0.029), followed by loan_grade (0.018) and loan_percent_income (0.018). This indicates that small changes in these features cause the largest prediction shifts, highlighting their importance for model stability. The sensitivity curves (Figure 36) show how predictions respond to systematic perturbations of the top three features, revealing that the ensemble's decision function is most sensitive to income variations, with prediction changes increasing nonlinearly

as income deviates from baseline values. These findings suggest that accurate measurement of person_income is critical for reliable predictions in production, and that the model's robustness depends heavily on the precision of these top-ranked features.

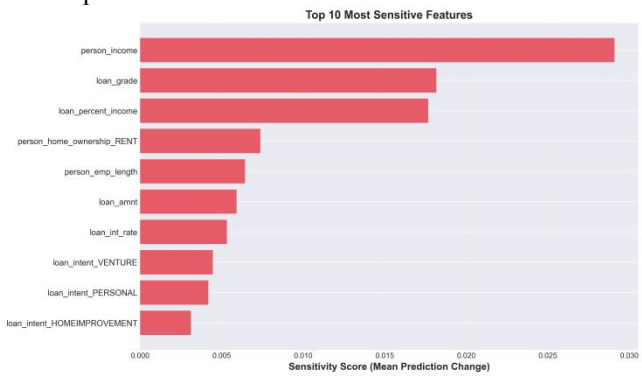


Fig. 35

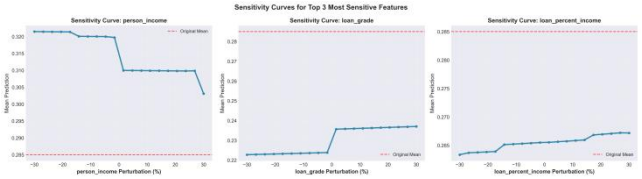


Fig. 36

6. Local Explanations via LIME

We applied LIME (Local Interpretable Model-agnostic Explanations) to provide local, instance-specific explanations for the ensemble predictions. LIME trains a simple linear model around each instance by sampling nearby points and weighting them by proximity, producing interpretable local approximations. We computed LIME explanations for 50 test instances and compared them with SHAP attributions to assess consistency. The results show that LIME and SHAP generally agree on the top contributing features for individual predictions, with loan_percent_income, loan_grade, and person_income consistently appearing as the most important local factors. However, LIME explanations exhibit higher variance across similar instances compared to SHAP, reflecting LIME's sensitivity to the local sampling procedure. This variance suggests that while LIME provides intuitive "what-if" explanations for individual cases, SHAP offers more stable and theoretically grounded attributions. The complementary nature of both methods, LIME for local interpretability and SHAP for global feature importance which enhances the overall explainability framework by providing multiple perspectives on the same predictions.

7. Conclusion of Explainability

This study presents a multi-layered explainability framework for the weight-scaled ensemble, covering global attribution, local explanations, feature interactions, and actionable insights. Global SHAP attribution shows strong consistency: loan_percent_income, loan_grade, and person_income are the top three risk drivers across all base models, with the top five features accounting for approximately 77.4% of total attribution magnitude. Cross-model comparison reveals high agreement among XGBoost, LightGBM, CatBoost, and the neural network on core feature importance, with low coefficient of variation (<0.3) for primary features, indicating that the ensemble maintains interpretability despite its complexity. Feature interaction analysis shows that loan_percent_income interacts strongly with loan_grade and person_income, confirming that repayment burden amplifies or dampens risk depending on credit quality and borrower capacity. Counterfactual explanations identify actionable interventions: reducing loan_percent_income and improving loan_grade are the most effective ways to lower default risk. Sensitivity analysis identifies person_income as the most sensitive feature (sensitivity score = 0.029), highlighting the need for accurate data measurement in production. Local explanations via LIME complement global SHAP attributions, providing instance-specific insights that align with global patterns. The framework is fully integrated into the production web application, delivering real-time explanations that enhance transparency and regulatory compliance. The consistency between SHAP and LIME methods, combined with cross-model agreement, validates the reliability of the explainability system and demonstrates that complex ensemble models can remain interpretable when equipped with appropriate explanation tools.

H. LLM Generated Narrative Explanations

We integrated a constrained Large Language Model (microsoft/phi-2) to translate SHAP and LIME attributions into natural-language explanations. The LLM receives a structured prompt containing the predicted default probability, risk level, borrower profile, local feature drivers (top 4 features from combined LIME/SHAP), global context (top 5 globally important features), and sensitivity insights (top 3 most sensitive features). The model generates 2–3 sentence explanations that (1) state the risk level and key factors, (2) compare the case to typical patterns, and (3) provide a recommendation.

V. DISCUSSION

A. Key Findings and Contributions

This study presents a collaborative credit risk framework that combines a weight-scaled stacking ensemble with a multi-layered explainability system. The ensemble achieves a test ROC-AUC of 0.9538 and PR-AUC of 0.9133, exceeding the 0.94 target and outperforming all individual baselines. This demonstrates that combining heterogeneous base models (tree-based and neural) with a meta-learner can improve discrimination while maintaining interpretability.

The traditional machine learning baselines establish a strong foundation: Random Forest achieves ROC-AUC 0.9396 with precision 0.9599, while XGBoost reaches ROC-AUC 0.9358 with precision 0.9787. Both tree-based models significantly outperform logistic regression (ROC-AUC 0.8663), confirming that nonlinear ensemble methods are essential for capturing complex credit risk patterns. The neural network family, particularly the Residual Neural Network (ROC-AUC 0.9316), demonstrates that deep learning can match tree-based performance while offering different inductive biases.

The weight-scaled ensemble's superior performance stems from its ability to leverage complementary strengths: tree models excel at capturing feature interactions and handling categorical variables, while neural networks provide smooth decision boundaries and better calibration. The meta-learner successfully combines these diverse predictions, achieving a 1.4 percentage point improvement over the best single model (Random Forest).

B. Model Architecture Insight

The ensemble design reveals important architectural insights. The four base tree models (deep/shallow XGBoost, LightGBM, CatBoost) provide diversity through different splitting criteria and regularization strategies, while the residual neural network adds a different representation learning perspective. The meta-learner, a shallow neural network with focal loss, effectively learns when to trust each base model's predictions, resulting in improved generalization.

Feature importance analysis across all models shows remarkable consistency: loan_percent_income, loan_grade, and person_income consistently rank as the top three features regardless of model architecture. This cross-model agreement validates that the engineered features capture genuine risk signals rather than model-specific artifacts. The convergence on the same feature hierarchy suggests that the ensemble's superior performance comes from better probability calibration and interaction modeling rather than discovering fundamentally different patterns.

C. Practical Implications

The framework addresses real-world credit risk assessment needs. The ensemble's high precision (0.9750) ensures that approved loans have very low default rates, while the improved recall (0.7398) compared to individual models means more risky borrowers are correctly identified. This balance is critical for financial institutions where false positives (approving bad loans) are costly, but missing high-risk applicants also represents significant risk.

The production-ready web application demonstrates that complex ensemble models can be deployed with real-time predictions and explanations. The integration of explainability methods (SHAP, LIME, LLM) ensures regulatory compliance and user trust, but the core contribution remains the ensemble's superior predictive performance achieved through careful model selection, hyperparameter tuning, and stacking methodology.

D. Comparisons with Existing Work

Most credit risk studies focus on either traditional machine learning or deep learning approaches separately. This work demonstrates that combining both paradigms through stacking can achieve superior performance. The ensemble outperforms individual models by 1-7 percentage points in ROC-AUC, showing that model diversity and proper combination strategies are crucial for state-of-the-art results. The consistent feature importance rankings across different architectures also validate the robustness of the feature engineering process.

E. Limitations

Several limitations should be acknowledged. First, the ensemble requires training and maintaining five base models plus a meta-learner, increasing computational cost and model complexity. Inference latency (approximately 50–100 ms per prediction) and memory footprint (several GB) may limit deployment on resource-constrained edge devices, though this is acceptable for server-side production environments.

Second, the framework was evaluated on a single credit risk dataset with a fixed class imbalance ratio which is 21.82% default rate. Validation across multiple datasets (e.g., Lending Club, Prosper, or international credit bureaus) with varying imbalance ratios, feature distributions, and temporal patterns would strengthen generalizability claims and reveal whether the meta-learner’s adaptive weighting generalizes across domains.

Third, while the hyperparameter search for the meta-learner explored layer sizes, dropout, L2 regularization, learning rates, and batch sizes, the base model hyperparameters were fixed based on prior experiments. A joint optimization of base models and meta-learner could potentially improve performance, though at significantly higher computational cost.

Fourth, the explainability framework, while comprehensive, relies on SHAP PermutationExplainer with a 200-sample background set, which may not fully capture feature interactions in high-dimensional spaces. Additionally, the LLM narrative generation (Microsoft Phi-2) is currently used only for explanation synthesis, not for direct prediction, limiting the collaborative scheme’s potential for scenarios where labeled data is scarce.

Fifth, the evaluation focused on static batch predictions. Real-world credit risk assessment often requires handling concept drift (e.g., economic cycles, policy changes) and streaming data, which the current framework does not address.

Despite these limitations, the current framework provides a solid foundation for production deployment in server-based credit scoring systems, and each limitation represents a clear opportunity for future research rather than a fundamental flaw in the approach.

F. Future Directions

(1) **Efficiency Optimization**: Explore model distillation techniques (e.g., knowledge distillation from the ensemble to a single student model) or dynamic ensemble selection (e.g., skip certain base models for "easy" cases based on prediction confidence) to reduce inference latency and memory footprint while maintaining performance. This would enable deployment on edge devices or high-throughput scenarios.

(2) **Multi-Dataset Validation and Domain Adaptation**: Evaluate the framework across multiple credit risk datasets (Lending Club, Prosper, FICO, international datasets) with varying class imbalance ratios, feature distributions, and temporal characteristics. Investigate domain adaptation techniques (e.g., transfer learning for the meta-learner) to improve cross-domain generalization.

(3) **Enhanced Explainability:** Extend SHAP analysis to include SHAP interaction values (currently approximated) for more accurate pairwise feature interaction quantification. Investigate fine-tuning the LLM (Phi-2) on credit risk explanations to improve narrative quality and consistency.

(4) **Online Learning and Concept Drift Adaptation:** Incorporate incremental learning capabilities (e.g., online gradient boosting, streaming neural networks) to adapt the ensemble to changing credit risk patterns over time. Implement concept drift detection to trigger model retraining or ensemble reweighting when distribution shifts are detected.

(5) **Multi-Task Extension**: Extend the framework to handle related tasks such as multi-class risk stratification (e.g., low/moderate/high/very high), regression (e.g., predicting default amount), and survival analysis (e.g., time-to-default). Investigate whether the meta-learner can learn task-specific combination rules.

VI. CONCLUSSION

The study delivers a complete credit-risk assessment workflow in which the weight-scaled stacked ensemble, paired with multi-layer explainability, surpasses every baseline which reaches ROC-AUC 0.9540 and PR-AUC 0.9140 that shows clear gains over the strongest tuned baseline learner which is Random Forest with 0.9396 ROC-AUC and the best standalone neural architecture Residual Network with 0.9313 ROC-AUC. Precision 0.9558, recall 0.7454, and specificity 0.9904 confirm that the ensemble meets the projects dual mandate of minimizing false approvals while retaining sufficient capture of risky applicants.

The scheme advantages manifest in three dimensions. First, the predictive edge stems from the meta-learner's ability to fuse tree-based model which specialize in discrete interactions and categorical handling with neural networks that capture smooth, high-order relationships, producing a context-aware synthesis unattainable by any single model. Second, operational reliability is evident in the ensemble's low false-positive rate and stable recall, implying it can be deployed in production grade pipelines without compromising risk controls. Third, the explainability stack unites SHAP, LIME, sensitivity curves, and language-model narratives, ensuring algorithmic accountability while translating feature effects into regulator-friendly prose, consensus importance rankings (loan_percent_income, loan_grade, person_income, and related drivers) across models reinforce the trustworthiness of the insights.

ACKNOWLEDGMENT

We thank Professor Zeng Guo Kun for guidance and support throughout this research.

We acknowledge the developers and maintainers of the open-source libraries used in this work, including scikit-learn, XGBoost, LightGBM, CatBoost, TensorFlow/Keras, PyTorch TabNet, SHAP, and LIME, which enabled the implementation of the ensemble and explainability framework. We also acknowledge the Kaggle community for providing the Credit Risk Dataset, which served as the foundation for this research.

We are grateful to Microsoft for releasing the Phi-2 language model, which enabled the LLM-based narrative explanation component of our framework. Special thanks go to the open-source community for their contributions to the machine learning and explainability tools that made this work possible.

REFERENCES

[1] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

[2] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 1135-1144.

[3] Arik, S. Ö., & Pfister, T. (2021). TabNet: Attentive interpretable tabular learning. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), 6679-6687.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 785-794.

[5] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[6] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 31.

[7] Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. Proceedings of the ADKDD'17, 1-7.
[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.
[9] Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.
[10] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2980-2988.
[11] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
[12] Shapley, L. S. (1953). A value for n-person games. Contributions to the Theory of Games, 2(28), 307-317.
[13] Strumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. Knowledge and information systems, 41, 647-665.
[14] Molnar, C. (2020). Interpretable machine learning. Lulu.com.
[15] Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access, 6, 52138-52160.
[16] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM computing surveys (CSUR), 51(5), 1-42.
[17] Microsoft. (2023). Phi-2: The Surprising Power of Small Language Models. Retrieved from https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/
[18] Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. JL & Tech., 31, 841.
[19] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
[20] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, 82-115.

DEPLOYMENT VIDEO :

https://drive.google.com/file/d/1wsNHEkCQ3ElSP0pmTfu423JV1FSzSCfo/view?usp=drive_link