XG Boost with Time Series

🔍 Model Overview

This model is built to **predict loan repayment status** (loan_status) using advanced preprocessing, feature engineering, and **XGBoost** with **GPU acceleration** for fast and scalable learning. It uses **Time Series Cross-Validation (TSCV)** to mimic real-world, forward-moving data, which is critical in financial forecasting.

🧱 Preprocessing and Feature Engineering

1. **Base Feature Engineering**:
    - Introduces new ratios like income_per_age, debt_income_ratio, and loan_age_ratio that reflect borrowers' financial behavior more deeply than raw variables.
    - Log transformations (log_person_income, log_loan_amnt) reduce skewness in high-income or large-loan cases.
    - Interactions (e.g., income_loan_interaction) help the model capture non-linear patterns.
2. **Polynomial Features**:
    - Adds squared terms (like loan_amnt_squared) to help XGBoost capture quadratic relationships.
3. **Binning**:
    - Uses quantile-based (qcut) and fixed-bin (cut) techniques to bucket income, age, loan amount, and employment length for potential threshold effects.
4. **Behavioral & Credit History Indicators**:
    - Binary flags (has_short_credit, has_long_credit) encode the quality of employment history.
    - credit_risk_score multiplies loan-to-income ratio with default history, creating a proxy for financial stress.
5. **Categorical Encoding**:

- One-hot encoding is applied to categorical features and new combinations like grade_intent (loan grade + intent), improving model flexibility.

⚙️ Data Preparation

- **Imputation**: Missing values are filled with mean values using SimpleImputer.
- **Scaling**: All features are standardized with StandardScaler to ensure uniform input distributions.
- **Dimensionality Reduction** (optional): PCA reduces dimensionality to 10 components, helping with model simplicity and potentially mitigating overfitting.

⚖️ Class Imbalance Handling

- **ADASYN** (Adaptive Synthetic Sampling) oversamples the minority class (e.g., defaults) by creating synthetic points in sparse regions, making the model more sensitive to underrepresented cases.

🧪 Model Training with TimeSeriesSplit

- **5-fold TimeSeriesSplit** ensures that each fold uses earlier data to predict later outcomes — crucial for avoiding *look-ahead bias* in credit risk.
- Uses **XGBoost (GPU accelerated)**:
  - Objective: binary:logistic for classification
  - Boosting rounds: 100
  - scale_pos_weight is dynamically adjusted per fold to tackle class imbalance during training.

📊 Evaluation Metrics

The model evaluates performance using:

- **Accuracy**: How often predictions match true values.
- **Precision**: Ability to correctly identify defaulters.

- **Recall**: How well the model captures actual defaulters (key for risk mitigation).
- **F1-score**: Harmonic mean of precision and recall.
- **AUC (Area Under ROC Curve)**: Ability to distinguish between classes.

✅ **Observed Performance Across Folds:**

Metric	Fold Range	Average	Insights
Accuracy	89.9% – 92.0%	~90.9%	Very high; shows strong overall predictive ability.
AUC	0.855 – 0.930	~0.893	Excellent class separation capability.
Precision	~0.91 (avg)	~0.91	Confident in positive predictions (defaults).
Recall	~0.89 (avg)	~0.89	Captures most true defaults, though there's room for improvement.
F1-score	~0.90 (avg)	~0.90	Balanced performance on both precision and recall.

Note: Some folds may have NaN AUC if one class is missing in test data — this is a known limitation with imbalanced splits.

📈 **Visual Insights**

1. **Metric Trend Charts** across folds show **consistent performance** with minor fluctuations.
2. **Feature Importance** (by frequency in trees):
    - Likely top features:
        - person_income
        - loan_grade_*
        - loan_int_rate (not shown in code but likely part of original features)
        - credit_risk_score
        - Engineered ratios like loan_to_income and income_per_age

These features are intuitive and align with domain knowledge — income, loan intent, and credit history are strong risk indicators.

💾 Model Export

- Final trained model is saved as xgboost_credit_model_advanced.pkl for future use — ideal for integration into dashboards, APIs, or batch prediction pipelines.

📌 Recommendations for Improvement

1. **Hyperparameter Tuning**:
    - Use GridSearchCV or Optuna to optimize parameters like learning_rate, max_depth, and subsample.
2. **Ensemble Voting**:
    - Combine with other classifiers (like LightGBM, logistic regression) in a stacked or voting ensemble for robustness.
3. **Explainability Tools**:
    - Add **SHAP values** to visualize individual prediction drivers — very useful in financial and regulatory settings.
4. **Temporal Feature Engineering**:
    - If available, use date/time info to create features like seasonality, month effect, or recent payment history.
5. **Threshold Optimization**:
    - Instead of default 0.5 cutoff, choose threshold maximizing F1 or minimizing misclassification cost.

📚 Summary

This model combines **thoughtful feature engineering**, **robust validation**, **balanced class handling**, and **high computational efficiency**. It performs strongly across all key metrics and provides a solid foundation for deployment in a real-world credit risk scoring system.

Let me know if you'd like:

- A short version for a presentation/report,
- A notebook markdown summary,
- Or analysis comparing this with another model (e.g., logistic regression or random forest).