Lending Club Problem

Executive Summary

This report analyzes a machine learning pipeline implementing LightGBM for binary classification of loan defaults using Lending Club dataset. The model employs undersampling for class imbalance,feature engineering, and hyperparameter tuning via grid search to predict loan status outcomes.

Problem Definition

Business Problem: Predict loan default risk to minimize financial losses and optimize lending decisions.
Technical Problem: Binary classification task where:
Target Variable: loan_status (0 = No default, 1 = Default)
Objective: Classify loans as likely to default or not default
Success Metric: F1-score (balances precision and recall for imbalanced datasets)

Dataset Overview

Data Sources:
Training Data: /content/train_lending_club.csv
Test Data: /content/test_lending_club.csv
Data Quality Issues Addressed:
- Outlier Removal: Applied Z-score filtering (threshold < 3) on numeric columns
- Missing Values: Dropped rows with missing target values
- Data Leakage Prevention: Feature engineering focused on information available at loan origination

Feature Engineering & Selection

Original Feature Processing:
FICO Score Consolidation:
df1['fico_score'] = (df1['fico_range_low'] + df1['fico_range_high']) / 2
- Combined FICO range into single representative score
- Dropped original fico_range_low and fico_range_high
Selected Features (15 Total):
Categorical Features (5):
- sub_grade: Loan sub-grade rating
- term: Loan term duration
- home_ownership: Borrower's home ownership status
- verification_status: Income verification status
- initial_list_status: Initial listing status

Numerical Features (10):
- fico_score (engineered)
- annual_inc- mort_acc
- time_to_earliest_cr_line
- emp_length
- revol_bal
- revol_util
- dti
- int_rate
- loan_amnt

Advanced Feature Engineering

Ratio-Based Features:
- debt_to_income: loan_amnt / annual_inc
- available_revol_credit: revol_bal / (revol_util + 1e-6)

Feature Replacement Strategy:
- Replaced raw financial amounts with meaningful ratios
- Dropped original components: loan_amnt, annual_inc, revol_bal, revol_util

Approach 1:

Features: 15 selected features including 5 categorical (sub_grade, term, home_ownership, verification_status, initial_list_status) and 10 numerical with engineered ratios like debt-to-income.

Sampling: Used **undersampling** to balance classes by reducing majority class size to match minority class, avoiding computational overhead.

Model: **LightGBM** with gradient boosting, 3-fold stratified cross-validation, and grid search across 32 hyperparameter combinations (num_leaves, max_depth, learning_rate, n_estimators, scale_pos_weight).

Evaluation:

```
Optimal Threshold for max F1: 0.16348804101683737

===== Test Results =====
Test Accuracy: 0.8681188372664438
Test AUC: 0.8354937993049932

Classification Report:
              precision    recall  f1-score   support

         0.0       0.70      0.28      0.40     14748
         1.0       0.88      0.98      0.93     78859

    accuracy                           0.87     93607
   macro avg       0.79      0.63      0.67     93607
weighted avg       0.85      0.87      0.84     93607
```

Approach-2((LightGBM, Random Forest, Logistic Regression) + Logistic Regression**)**

Sampling: Same **undersampling** approach to balance classes by matching majority class size to minority class.
Models: **Stacking ensemble** with 3 base learners (LightGBM, Random Forest, Logistic Regression) + Logistic Regression as meta-learner, all wrapped in scikit-learn pipeline.
Evaluation:

```
===== Test Results =====
Accuracy:  0.8695
AUC:       0.8339
F1 Score:  0.9262
Precision: 0.8848
Recall:    0.9716
```

Approach -3

Sampling: No explicit sampling - relies on individual model class balancing (class_weight="balanced", scale_pos_weight) instead of dataset-level undersampling.

Models: 3-layer stacking - AdaBoost (with Decision Trees), LightGBM, XGBoost as base learners + Logistic Regression meta-learner, using 5-fold cross-validation with out-of-fold predictions.

Evaluation:

```
===== Test Results =====
Accuracy: 0.854006644802205
AUC: 0.7806550566043255

Classification Report:
              precision    recall  f1-score   support

         0.0       0.61      0.20      0.30     14748
         1.0       0.87      0.98      0.92     78859

    accuracy                           0.85     93607
   macro avg       0.74      0.59      0.61     93607
weighted avg       0.83      0.85      0.82     93607
```

Approach 4 (adaboost+lgbm)

Problem: Same loan default prediction using **AdaBoost with LightGBM as base estimator** - combining adaptive boosting with gradient boosting for enhanced sequential learning.
Features: Same 15 features with identical engineering, but uses **OneHotEncoder** + **StandardScaler** preprocessing pipeline with median imputation for missing values.
Sampling: **Undersampling with replacement** (unlike previous models that used without replacement) to balance classes by matching majority to minority class size.
Models: **Hybrid AdaBoost-LightGBM** - AdaBoost (50 estimators, 0.1 learning rate) using LightGBM classifiers (50 estimators each) as weak learners, wrapped in sklearn Pipeline with imputation.

Evaluation:

```
Accuracy: 0.8671253218242225
AUC: 0.8293856634899958

Classification Report:
              precision    recall  f1-score   support

         0.0       0.68      0.30      0.42     14748
         1.0       0.88      0.97      0.93     78859

    accuracy                           0.87     93607
   macro avg       0.78      0.64      0.67     93607
weighted avg       0.85      0.87      0.84     93607
```

Approach 5 (Balanced_Bagging)

Link:

https://medium.com/@nageshmashette32/balanced-bagging-classifier-bagging-for-imbalanced-classification-dfba66c44c14

Problem: Same loan default prediction using **custom balanced bagging ensemble** - training multiple LightGBM models on different balanced subsets and averaging predictions.

Features: Same 15 features with identical engineering, but uses **OneHotEncoder** for categorical variables and combines numerical + encoded features into single array.

Sampling: **Custom oversampling -** for each of 5 models, upsamples minority class to match majority class size (opposite of undersampling), creating diverse balanced training sets.

Models: **Ensemble of 5 LightGBM models** (100 estimators each) trained on different balanced samples with class weighting (minority class weight=5), using **soft voting** (average probabilities) for final predictions.

Evaluation: Same threshold optimization and metrics, but leverages **ensemble diversity** from multiple models trained on different balanced subsets to improve prediction stability and performance.

```
Optimal Threshold: 0.48981957303541357
Accuracy: 0.8693580608287842
AUC: 0.8358671099857073

Classification Report:
              precision    recall  f1-score   support

         0.0       0.69      0.31      0.43     14748
         1.0       0.88      0.97      0.93     78859

    accuracy                           0.87     93607
   macro avg       0.79      0.64      0.68     93607
weighted avg       0.85      0.87      0.85     93607
```

Approach 6 (ADASYN+Stacking)

Problem: Same loan default prediction using **ADASYN oversampling + stacking ensemble** - combines synthetic minority oversampling with multi-algorithm ensemble for handling class imbalance.

Features: Same 15 features with identical engineering, but uses **ColumnTransformer** with OneHotEncoder for categorical and StandardScaler for numerical features in integrated preprocessing pipeline.

Sampling: **ADASYN oversampling** (Adaptive Synthetic Sampling) - generates synthetic minority class samples based on data density distribution, more sophisticated than basic SMOTE.

Models: **3-algorithm stacking ensemble** - LightGBM, XGBoost, and AdaBoost as base learners with Logistic Regression meta-learner, all wrapped in **imbalanced-learn Pipeline** with built-in ADASYN.

Evaluation: Integrated pipeline approach (preprocessing → ADASYN → stacking) with threshold optimization, measuring standard metrics while leveraging both synthetic data generation and ensemble learning for improved minority class detection.

```
  warnings.warn(

===== Test Results =====
Optimal Threshold: 0.2787
Accuracy:  0.8662
AUC:       0.8285
F1 Score:  0.9247
Precision: 0.8796
Recall:    0.9745
```

Approach 7(custom(loss function)+Stacking.)

Problem: Advanced loan default prediction using **custom weighted loss functions + stacking ensemble** - mathematically penalizes minority class errors 5x more without any data sampling.

Features: Same 15 features but uses **ColumnTransformer pipeline** with OneHotEncoder + StandardScaler, emphasizing mathematical preprocessing over data manipulation approaches.

Sampling: **No sampling** - instead uses **custom loss functions** that weight minority class errors 5x higher through modified gradients/hessians in LightGBM and XGBoost training.

Models: **Cost-sensitive stacking** - Custom weighted BCE for LightGBM, custom weighted log-loss for XGBoost, standard AdaBoost, combined with balanced Logistic Regression meta-learner.

Evaluation: **Comprehensive threshold analysis** with precision-recall curves and multi-metric visualization across 100 threshold points, representing the most mathematically sophisticated approach in the series.

```
===== Test Results =====
Optimal Threshold (F1 max): 0.1139
Accuracy:  0.8949
AUC:       0.7969
F1 Score:  0.9151
Precision: 0.8491
Recall:    0.9923
```