Kaggle Playground

Problem Statement / Real World Implementations

In [1]:
```
# --- 1. Importing Libraries ---
import numpy as np
import pandas as pd
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
from sklearn.model_selection import StratifiedKFold, train_test_split # Import S
from sklearn.preprocessing import OrdinalEncoder, StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, r
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
import optuna

# Notebook settings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)

# --- 2. Loading Dataset ---
TRAIN_PATH = "/kaggle/input/playground-series-s5e11/train.csv"
TEST_PATH = "/kaggle/input/playground-series-s5e11/test.csv"
SUBMISSION_PATH = "/kaggle/input/playground-series-s5e11/sample_submission.csv"

train_df = pd.read_csv(TRAIN_PATH)
test_df = pd.read_csv(TEST_PATH)
submission_df = pd.read_csv(SUBMISSION_PATH)

# Save the test IDs now
test_ids = test_df['id']

# Drop 'id' from both
train_df = train_df.drop('id', axis=1)
test_df = test_df.drop('id', axis=1)


# --- 3. Feature Engineering (From V4) ---
def create_financial_features(df):
    df['monthly_income'] = df['annual_income'] / 12
    df['total_monthly_debt'] = df['debt_to_income_ratio'] * df['monthly_income']
    df['available_income'] = df['monthly_income'] - df['total_monthly_debt']
    df['loan_to_income_ratio'] = df['loan_amount'] / df['annual_income']
    df['loan_to_available_income'] = df['loan_amount'] / df['available_income']
    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    cols_to_drop = ['annual_income', 'debt_to_income_ratio']
    df = df.drop(columns=cols_to_drop)
    return df

print("Creating financial features for train_df...")
train_df = create_financial_features(train_df)
print("Creating financial features for test_df...")
test_df = create_financial_features(test_df)


# --- 4. Smart Encoding & Processing (From V4, with fillna=-1) ---
```

```python
def process_and_encode_features(df_train, df_test):
    train_target = df_train['loan_paid_back']
    df_train = df_train.drop('loan_paid_back', axis=1)

    df_train['source'] = 'train'
    df_test['source'] = 'test'
    combined_df = pd.concat([df_train, df_test], ignore_index=True)

    # Bin Credit Score
    score_bins = [300, 579, 669, 739, 799, 850]
    score_labels = ['Poor', 'Fair', 'Good', 'Very Good', 'Excellent']
    combined_df['credit_score_bin'] = pd.cut(combined_df['credit_score'],
                                             bins=score_bins,
                                             labels=score_labels,
                                             include_lowest=True)

    # Logical Ordinal Mapping
    education_map = {'Other': 0, 'High School': 1, 'Bachelor\'s': 2, 'Master\'s'
    grades = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
    subgrades = ['1', '2', '3', '4', '5']
    grade_map = {g + s: i for i, (g, s) in enumerate((g, s) for g in grades for
    combined_df['education_level_ordinal'] = combined_df['education_level'].map(
    combined_df['grade_subgrade_ordinal'] = combined_df['grade_subgrade'].map(gr

    # Ordinal Encode Remaining Categoricals
    categorical_cols = ['gender', 'marital_status', 'employment_status', 'loan_p
    encoder = OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-
    combined_df['credit_score_bin'] = combined_df['credit_score_bin'].astype(str
    combined_df[categorical_cols] = encoder.fit_transform(combined_df[categorica

    # HANDLE NaNs (The FIX: fillna(-1))
    if combined_df['education_level_ordinal'].isnull().any():
        mode_val = combined_df['education_level_ordinal'].mode()[0]
        combined_df['education_level_ordinal'] = combined_df['education_level_or

    if combined_df['loan_to_available_income'].isnull().any():
        combined_df['loan_to_available_income'] = combined_df['loan_to_available

    # Drop old columns and split back
    cols_to_drop = ['credit_score', 'education_level', 'grade_subgrade']
    combined_df = combined_df.drop(columns=cols_to_drop)

    train_processed = combined_df[combined_df['source'] == 'train'].drop('source
    test_processed = combined_df[combined_df['source'] == 'test'].drop('source',

    train_processed['loan_paid_back'] = train_target
    return train_processed, test_processed

print("Processing and encoding all features (filling NaNs with -1)...")
train_processed, test_processed = process_and_encode_features(train_df, test_df)
print("Processing complete.")


# --- 5. Prepare Data for Tuning & K-Fold ---
X = train_processed.drop("loan_paid_back", axis=1)
y = train_processed["loan_paid_back"]
X = X.select_dtypes(include=[np.number])
X_submission = test_processed.select_dtypes(include=[np.number])
X_submission = X_submission[X.columns] # Align column order NOW
```

```python
# Create the 20% validation split *only for Optuna*
X_train_opt, X_val_opt, y_train_opt, y_val_opt = train_test_split(X, y, test_siz

# Scale this 20% split for Optuna
scaler_optuna = StandardScaler()
X_train_opt_scaled = scaler_optuna.fit_transform(X_train_opt)
X_val_opt_scaled = scaler_optuna.transform(X_val_opt)
print("Scaling for Optuna complete.")


# --- 6. Hyperparameter Tuning (n_trials=50, from V4) ---
# We run this first to get the best_params we'll use in the K-Fold
def objective_xgb(trial):
    param = {
        'tree_method': 'gpu_hist', 'predictor': 'gpu_predictor', 'gpu_id': 0,
        'lambda': trial.suggest_loguniform('lambda', 1e-3, 10.0),
        'alpha': trial.suggest_loguniform('alpha', 1e-3, 10.0),
        'colsample_bytree': trial.suggest_categorical('colsample_bytree', [0.3,
        'subsample': trial.suggest_categorical('subsample', [0.5, 0.6, 0.7, 0.8,
        'learning_rate': trial.suggest_float('learning_rate', 0.005, 0.05, log=T
        'n_estimators': trial.suggest_int('n_estimators', 200, 1000, step=100),
        'max_depth': trial.suggest_int('max_depth', 3, 12),
        'min_child_weight': trial.suggest_int('min_child_weight', 1, 300),
        'random_state': 42
    }
    model = XGBRegressor(**param, verbosity=0)
    model.fit(X_train_opt_scaled, y_train_opt) # Train on 80%
    y_pred = model.predict(X_val_opt_scaled) # Validate on 20%
    mse = mean_squared_error(y_val_opt, y_pred)
    return mse

def objective_lgbm(trial):
    param = {
        'device': 'gpu', 'gpu_platform_id': 0, 'gpu_device_id': 0,
        'boosting_type': 'gbdt', 'objective': 'regression', 'metric': 'mse',
        'lambda_l1': trial.suggest_float('lambda_l1', 1e-5, 1.0, log=True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-5, 1.0, log=True),
        'num_leaves': trial.suggest_int('num_leaves', 16, 256),
        'feature_fraction': trial.suggest_float('feature_fraction', 0.5, 1.0),
        'bagging_fraction': trial.suggest_float('bagging_fraction', 0.5, 1.0),
        'bagging_freq': trial.suggest_int('bagging_freq', 1, 7),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 100),
        'learning_rate': trial.suggest_float('learning_rate', 0.005, 0.3, log=Tr
        'n_estimators': trial.suggest_int('n_estimators', 200, 1000, step=100),
        'max_depth': trial.suggest_int('max_depth', 3, 12),
        'random_state': 42, 'verbosity': -1
    }
    model = LGBMRegressor(**param)
    model.fit(X_train_opt_scaled, y_train_opt) # Train on 80%
    y_pred = model.predict(X_val_opt_scaled) # Validate on 20%
    mse = mean_squared_error(y_val_opt, y_pred)
    return mse

print("🚀 Tuning XGBRegressor (GPU)...")
study_xgb = optuna.create_study(direction='minimize')
study_xgb.optimize(objective_xgb, n_trials=50, timeout=1200) # n_trials=50
best_params_xgb = study_xgb.best_params
print(f"✅ Best XGBRegressor parameters: {best_params_xgb}")
```

```
print("\n🚀 Tuning LGBMRegressor (GPU)...")
study_lgbm = optuna.create_study(direction='minimize')
study_lgbm.optimize(objective_lgbm, n_trials=50, timeout=1200) # n_trials=50
best_params_lgbm = study_lgbm.best_params
print(f"✅ Best LGBMRegressor parameters: {best_params_lgbm}")


# --- 7. K-Fold Cross-Validation for OOF & Test Predictions ---
print("\n🚀 Starting 10-Fold Cross-Validation for robust submission...")

N_SPLITS = 10
skf = StratifiedKFold(n_splits=N_SPLITS, shuffle=True, random_state=42)

# Arrays to store out-of-fold and test predictions
oof_preds_xgb = np.zeros(len(X))
test_preds_xgb = np.zeros(len(X_submission))
oof_preds_lgbm = np.zeros(len(X))
test_preds_lgbm = np.zeros(len(X_submission))

# Initialize models with best params from Optuna
xgb_model = XGBRegressor(**best_params_xgb, tree_method='gpu_hist', predictor='g
lgbm_model = LGBMRegressor(**best_params_lgbm, device='gpu', verbosity=-1)

for fold, (train_idx, val_idx) in enumerate(skf.split(X, y)):
    print(f"--- Fold {fold+1}/{N_SPLITS} ---")

    # --- Get fold data ---
    X_train_fold, X_val_fold = X.iloc[train_idx], X.iloc[val_idx]
    y_train_fold, y_val_fold = y.iloc[train_idx], y.iloc[val_idx]

    # --- Fit Scaler INSIDE the fold (to prevent leakage) ---
    scaler_fold = StandardScaler()
    X_train_fold_scaled = scaler_fold.fit_transform(X_train_fold)
    X_val_fold_scaled = scaler_fold.transform(X_val_fold)
    X_submission_scaled_fold = scaler_fold.transform(X_submission)

    # --- Train XGB ---
    print("Training XGB...")
    xgb_model.fit(X_train_fold_scaled, y_train_fold)
    oof_preds_xgb[val_idx] = xgb_model.predict(X_val_fold_scaled)
    test_preds_xgb += xgb_model.predict(X_submission_scaled_fold) / N_SPLITS

    # --- Train LGBM ---
    print("Training LGBM...")
    lgbm_model.fit(X_train_fold_scaled, y_train_fold)
    oof_preds_lgbm[val_idx] = lgbm_model.predict(X_val_fold_scaled)
    test_preds_lgbm += lgbm_model.predict(X_submission_scaled_fold) / N_SPLITS

# --- 8. OOF Blending Grid Search (Your Request) ---
print("\n--- OOF Blending Grid Search ---")
# Get OOF AUC for each model
oof_auc_xgb = roc_auc_score(y, oof_preds_xgb)
oof_auc_lgbm = roc_auc_score(y, oof_preds_lgbm)
print(f"OOF XGB Regressor AUC: {oof_auc_xgb:.6f}")
print(f"OOF LGBM Regressor AUC: {oof_auc_lgbm:.6f}")

best_auc = 0
best_weight = 0
# Grid search for the best weight
```

```python
for weight in np.arange(0.01, 1.0, 0.01):
    blended_oof = (weight * oof_preds_lgbm) + ((1 - weight) * oof_preds_xgb)
    auc = roc_auc_score(y, blended_oof)

    if auc > best_auc:
        best_auc = auc
        best_weight = weight

print(f"\n✅ Best OOF Blend AUC: {best_auc:.6f}")
print(f"   -> Best Weight (for LGBM): {best_weight:.2f}")


# --- 9. Create Final Blended Submission ---
print("\n⏳ Generating blended submission with best weight...")
# We use the averaged test predictions
final_blended_preds = (best_weight * test_preds_lgbm) + ((1 - best_weight) * tes
final_submission_preds = np.clip(final_blended_preds, 0, 1)

submission = pd.DataFrame({
    'id': test_ids,
    'loan_paid_back': final_submission_preds
})
submission.to_csv('submission_v7_kfold_blend.csv', index=False)
print("\n✅ Submission file 'submission_v7_kfold_blend.csv' generated successful
display(submission.head())

# --- 10. Final Plot ---
plt.figure(figsize=(8, 5))
sns.histplot(submission['loan_paid_back'], bins=30, kde=True)
plt.title('Distribution of Blended Predicted Loan Payback (K-Fold)')
plt.xlabel('Loan Payback Probability')
plt.ylabel('Frequency')
plt.show()
```

```
Creating financial features for train_df...
Creating financial features for test_df...
Processing and encoding all features (filling NaNs with -1)...
Processing complete.
[I 2025-11-04 05:50:45,216] A new study created in memory with name: no-
name-222eea70-08d7-46b8-821e-469d3362901e
Scaling for Optuna complete.
🚀 Tuning XGBRegressor (GPU)...
```

```
[I 2025-11-04 05:50:47,950] Trial 0 finished with value: 0.07581028269410184 and
parameters: {'lambda': 9.293953318459463, 'alpha': 3.001490191370945, 'colsample_bytr
0.9, 'subsample': 0.6, 'learning_rate': 0.041160146611354115, 'n_estimators': 200,
'max_depth': 8, 'min_child_weight': 58}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:50:59,780] Trial 1 finished with value: 0.07639725214735095 and
parameters: {'lambda': 2.7339618650929562, 'alpha': 0.005159541604905119,
'colsample_bytree': 0.7, 'subsample': 1.0, 'learning_rate': 0.03530443391323364,
'n_estimators': 800, 'max_depth': 12, 'min_child_weight': 95}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:51:03,307] Trial 2 finished with value: 0.077058771352278 and parame
{'lambda': 0.009592692684017409, 'alpha': 0.0018864314139151312, 'colsample_bytree':
'subsample': 0.8, 'learning_rate': 0.021297115111422526, 'n_estimators': 900, 'max_de
3, 'min_child_weight': 139}. Best is trial 0 with value: 0.07581028269410184.
[I 2025-11-04 05:51:07,572] Trial 3 finished with value: 0.07679820008480952 and
parameters: {'lambda': 0.003914611440445345, 'alpha': 0.10802741744068, 'colsample_by
0.7, 'subsample': 0.5, 'learning_rate': 0.0062383926939827035, 'n_estimators': 700,
'max_depth': 6, 'min_child_weight': 300}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:51:24,847] Trial 4 finished with value: 0.07590415480967319 and
parameters: {'lambda': 0.003535800742271127, 'alpha': 0.05490380384086695,
'colsample_bytree': 0.7, 'subsample': 0.7, 'learning_rate': 0.00978609846212531,
'n_estimators': 900, 'max_depth': 12, 'min_child_weight': 115}. Best is trial 0 with
0.07581028269410184.
[I 2025-11-04 05:51:27,314] Trial 5 finished with value: 0.07793274574773401 and
parameters: {'lambda': 1.2108838423800097, 'alpha': 2.080875448109786, 'colsample_byt
0.3, 'subsample': 1.0, 'learning_rate': 0.030507580261145216, 'n_estimators': 600,
'max_depth': 3, 'min_child_weight': 52}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:51:30,919] Trial 6 finished with value: 0.07594308828013575 and
parameters: {'lambda': 0.03210283186021761, 'alpha': 0.018535327811152993,
'colsample_bytree': 0.7, 'subsample': 0.7, 'learning_rate': 0.027015891281941855,
'n_estimators': 600, 'max_depth': 6, 'min_child_weight': 208}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:51:35,420] Trial 7 finished with value: 0.07605839836387253 and
parameters: {'lambda': 0.012263163209866248, 'alpha': 0.0013127337440396489,
'colsample_bytree': 0.9, 'subsample': 1.0, 'learning_rate': 0.007785924434363311,
'n_estimators': 600, 'max_depth': 7, 'min_child_weight': 125}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:51:52,516] Trial 8 finished with value: 0.07776931153640998 and
parameters: {'lambda': 0.0040225433795186075, 'alpha': 0.10103608228573731,
'colsample_bytree': 0.5, 'subsample': 0.5, 'learning_rate': 0.005276437932470416,
'n_estimators': 700, 'max_depth': 12, 'min_child_weight': 75}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:51:57,917] Trial 9 finished with value: 0.07588974911676048 and
parameters: {'lambda': 1.0314395598939683, 'alpha': 0.013695586524419209,
'colsample_bytree': 1.0, 'subsample': 0.7, 'learning_rate': 0.013392109824906242,
'n_estimators': 400, 'max_depth': 10, 'min_child_weight': 272}. Best is trial 0 with
0.07581028269410184.
[I 2025-11-04 05:52:00,489] Trial 10 finished with value: 0.07586999802396335 and
parameters: {'lambda': 8.548164428623284, 'alpha': 7.119287267250068, 'colsample_bytr
0.9, 'subsample': 0.6, 'learning_rate': 0.04452069071259796, 'n_estimators': 200,
'max_depth': 9, 'min_child_weight': 12}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:52:03,063] Trial 11 finished with value: 0.0758712563631687 and
parameters: {'lambda': 8.979495377402662, 'alpha': 8.920479255562103, 'colsample_bytr
0.9, 'subsample': 0.6, 'learning_rate': 0.047670380416252066, 'n_estimators': 200,
'max_depth': 9, 'min_child_weight': 4}. Best is trial 0 with value: 0.075810282694101
[I 2025-11-04 05:52:05,770] Trial 12 finished with value: 0.07592882423135425 and
parameters: {'lambda': 0.2819036093604268, 'alpha': 0.9428374381808818, 'colsample_by
0.9, 'subsample': 0.6, 'learning_rate': 0.048109811765830 57, 'n_estimators': 200,
'max_depth': 9, 'min_child_weight': 11}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:52:09,468] Trial 13 finished with value: 0.07584694378858485 and
parameters: {'lambda': 8.970920679731512, 'alpha': 8.790970369422466, 'colsample_bytr
```

```
0.9, 'subsample': 0.6, 'learning_rate': 0.0184196565367339, 'n_estimators': 400,
'max_depth': 8, 'min_child_weight': 53}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:52:11,763] Trial 14 finished with value: 0.0763114721871304 and
parameters: {'lambda': 0.08037247776563788, 'alpha': 0.7000230363739832,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.01740612637600855,
'n_estimators': 400, 'max_depth': 5, 'min_child_weight': 180}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:52:15,670] Trial 15 finished with value: 0.07691216524761096 and
parameters: {'lambda': 0.3717480834423677, 'alpha': 2.6207841881661214, 'colsample_by
0.5, 'subsample': 0.6, 'learning_rate': 0.011955238869604043, 'n_estimators': 400,
'max_depth': 8, 'min_child_weight': 47}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:52:20,442] Trial 16 finished with value: 0.07840600167957135 and
parameters: {'lambda': 3.6317111358575485, 'alpha': 0.3696281051227288, 'colsample_by
0.3, 'subsample': 0.8, 'learning_rate': 0.022318384734994055, 'n_estimators': 300,
'max_depth': 10, 'min_child_weight': 54}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:52:24,186] Trial 17 finished with value: 0.07586294167088534 and
parameters: {'lambda': 0.37585593045288773, 'alpha': 3.2337358644034455,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.016631442998694013,
'n_estimators': 500, 'max_depth': 7, 'min_child_weight': 163}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:52:26,029] Trial 18 finished with value: 0.07615219437989038 and
parameters: {'lambda': 0.001133316169683875, 'alpha': 0.3492531076250817,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.03591985181153919,
'n_estimators': 300, 'max_depth': 5, 'min_child_weight': 226}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:52:28,870] Trial 19 finished with value: 0.07589258173544527 and
parameters: {'lambda': 3.2146809713085673, 'alpha': 9.647557404966893, 'colsample_byt
0.9, 'subsample': 0.6, 'learning_rate': 0.023408621485360666, 'n_estimators': 300,
'max_depth': 8, 'min_child_weight': 81}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:52:36,677] Trial 20 finished with value: 0.07597429209713996 and
parameters: {'lambda': 1.256588915369405, 'alpha': 1.4693621882381882, 'colsample_byt
1.0, 'subsample': 0.8, 'learning_rate': 0.009906695542283217, 'n_estimators': 500,
'max_depth': 10, 'min_child_weight': 40}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:52:40,424] Trial 21 finished with value: 0.07585861309357252 and
parameters: {'lambda': 0.273037207771216, 'alpha': 4.367881497332242, 'colsample_bytr
0.9, 'subsample': 0.6, 'learning_rate': 0.016615080720067305, 'n_estimators': 500,
'max_depth': 7, 'min_child_weight': 166}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:52:43,634] Trial 22 finished with value: 0.07595737142827447 and
parameters: {'lambda': 0.12094749539820177, 'alpha': 5.267990638810116, 'colsample_by
0.9, 'subsample': 0.6, 'learning_rate': 0.01897731881798058, 'n_estimators': 500,
'max_depth': 6, 'min_child_weight': 189}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:52:47,374] Trial 23 finished with value: 0.07595103258256299 and
parameters: {'lambda': 6.062947437529783, 'alpha': 3.9149002029727957, 'colsample_byt
0.9, 'subsample': 0.6, 'learning_rate': 0.012806042908243763, 'n_estimators': 400,
'max_depth': 8, 'min_child_weight': 114}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:52:49,905] Trial 24 finished with value: 0.07609366029770386 and
parameters: {'lambda': 0.8081575402421742, 'alpha': 0.45373882342501987,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.01472362553326336,
'n_estimators': 300, 'max_depth': 7, 'min_child_weight': 154}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:52:52,691] Trial 25 finished with value: 0.07729328633076375 and
parameters: {'lambda': 2.073071534612372, 'alpha': 1.3309418500244277, 'colsample_byt
0.3, 'subsample': 0.5, 'learning_rate': 0.027663877134026703, 'n_estimators': 500,
'max_depth': 5, 'min_child_weight': 247}. Best is trial 0 with value: 0.0758102826941
[I 2025-11-04 05:53:08,599] Trial 26 finished with value: 0.07591119143979691 and
parameters: {'lambda': 0.11743594645387218, 'alpha': 0.24103530488234648,
'colsample_bytree': 0.5, 'subsample': 0.6, 'learning_rate': 0.01034305470801946,
'n_estimators': 1000, 'max_depth': 11, 'min_child_weight': 93}. Best is trial 0 with
0.07581028269410184.
[I 2025-11-04 05:53:10,703] Trial 27 finished with value: 0.07582944650382371 and
```

```
parameters: {'lambda': 0.04014378680352316, 'alpha': 4.321338465952585, 'colsample_by
0.9, 'subsample': 0.6, 'learning_rate': 0.03689136351452216, 'n_estimators': 200,
'max_depth': 8, 'min_child_weight': 68}. Best is trial 0 with value: 0.07581028269410
[I 2025-11-04 05:53:13,372] Trial 28 finished with value: 0.07584693166958886 and
parameters: {'lambda': 0.03551936835146387, 'alpha': 1.6056009479539626,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.038211058252641206,
'n_estimators': 200, 'max_depth': 9, 'min_child_weight': 29}. Best is trial 0 with va
0.07581028269410184.
[I 2025-11-04 05:53:16,084] Trial 29 finished with value: 0.0759610821027541 and
parameters: {'lambda': 0.025518552067997084, 'alpha': 0.8181600222910362,
'colsample_bytree': 0.9, 'subsample': 1.0, 'learning_rate': 0.03710567288151508,
'n_estimators': 200, 'max_depth': 9, 'min_child_weight': 25}. Best is trial 0 with va
0.07581028269410184.
[I 2025-11-04 05:53:19,827] Trial 30 finished with value: 0.07723952650933293 and
parameters: {'lambda': 0.027206658703117913, 'alpha': 0.20617948904964079,
'colsample_bytree': 0.3, 'subsample': 0.5, 'learning_rate': 0.04243959643024552,
'n_estimators': 200, 'max_depth': 11, 'min_child_weight': 74}. Best is trial 0 with v
0.07581028269410184.
[I 2025-11-04 05:53:22,810] Trial 31 finished with value: 0.07579865804972824 and
parameters: {'lambda': 0.04214475553060201, 'alpha': 1.9039263115392095,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.03155727357033346,
'n_estimators': 300, 'max_depth': 8, 'min_child_weight': 30}. Best is trial 31 with v
0.07579865804972824.
[I 2025-11-04 05:53:26,510] Trial 32 finished with value: 0.07581546248769293 and
parameters: {'lambda': 0.06267122457583892, 'alpha': 1.9125039673302908,
'colsample_bytree': 0.9, 'subsample': 0.6, 'learning_rate': 0.03096697295993383,
'n_estimators': 300, 'max_depth': 9, 'min_child_weight': 26}. Best is trial 31 with v
0.07579865804972824.
[I 2025-11-04 05:53:29,442] Trial 33 finished with value: 0.07578849177547532 and
parameters: {'lambda': 0.0590637614665525, 'alpha': 2.629539230571585, 'colsample_byt
1.0, 'subsample': 0.8, 'learning_rate': 0.032276672890830076, 'n_estimators': 300,
'max_depth': 8, 'min_child_weight': 66}. Best is trial 33 with value: 0.0757884917754
[I 2025-11-04 05:53:33,644] Trial 34 finished with value: 0.0759216475768219 and
parameters: {'lambda': 0.06755182459149221, 'alpha': 2.299687357111549, 'colsample_by
1.0, 'subsample': 0.8, 'learning_rate': 0.03167381821889696, 'n_estimators': 300,
'max_depth': 10, 'min_child_weight': 104}. Best is trial 33 with value:
0.07578849177547532.
[I 2025-11-04 05:53:35,718] Trial 35 finished with value: 0.07598478032221402 and
parameters: {'lambda': 0.013671848448530281, 'alpha': 0.7053961726966866,
'colsample_bytree': 1.0, 'subsample': 0.8, 'learning_rate': 0.026461938827347174,
'n_estimators': 300, 'max_depth': 6, 'min_child_weight': 28}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:53:38,582] Trial 36 finished with value: 0.0758281787088231 and
parameters: {'lambda': 0.05953575532334945, 'alpha': 1.1505140344117828,
'colsample_bytree': 0.7, 'subsample': 0.8, 'learning_rate': 0.03095237025685611,
'n_estimators': 300, 'max_depth': 8, 'min_child_weight': 136}. Best is trial 33 with
0.07578849177547532.
[I 2025-11-04 05:53:49,348] Trial 37 finished with value: 0.07649823866780665 and
parameters: {'lambda': 0.14176312313595363, 'alpha': 0.02742949785272816,
'colsample_bytree': 1.0, 'subsample': 0.7, 'learning_rate': 0.03321876455404922,
'n_estimators': 800, 'max_depth': 11, 'min_child_weight': 92}. Best is trial 33 with
0.07578849177547532.
[I 2025-11-04 05:53:52,477] Trial 38 finished with value: 0.07585512654389054 and
parameters: {'lambda': 0.007848807518467572, 'alpha': 2.173863991306137,
'colsample_bytree': 1.0, 'subsample': 1.0, 'learning_rate': 0.028704451109321818,
'n_estimators': 400, 'max_depth': 7, 'min_child_weight': 36}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:53:56,184] Trial 39 finished with value: 0.07581188258802778 and
parameters: {'lambda': 0.006941028689173039, 'alpha': 0.17362825524271913,
'colsample_bytree': 0.7, 'subsample': 0.8, 'learning_rate': 0.02494852356538896,
```

```
'n_estimators': 300, 'max_depth': 9, 'min_child_weight': 60}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:53:59,110] Trial 40 finished with value: 0.07725092734766455 and
parameters: {'lambda': 0.0018643722037964146, 'alpha': 0.006370532840090619,
'colsample_bytree': 0.7, 'subsample': 0.8, 'learning_rate': 0.025843579649215694,
'n_estimators': 700, 'max_depth': 3, 'min_child_weight': 61}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:02,767] Trial 41 finished with value: 0.07593056394005583 and
parameters: {'lambda': 0.018543087888492068, 'alpha': 0.07013494882166266,
'colsample_bytree': 0.7, 'subsample': 0.8, 'learning_rate': 0.04174357176733185,
'n_estimators': 300, 'max_depth': 9, 'min_child_weight': 18}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:05,733] Trial 42 finished with value: 0.07604085373033768 and
parameters: {'lambda': 0.00419421257291884, 'alpha': 0.13109295904192603,
'colsample_bytree': 0.7, 'subsample': 0.8, 'learning_rate': 0.025149057405744298,
'n_estimators': 200, 'max_depth': 9, 'min_child_weight': 4}. Best is trial 33 with va
0.07578849177547532.
[I 2025-11-04 05:54:10,201] Trial 43 finished with value: 0.07589595474450317 and
parameters: {'lambda': 0.007869352182380072, 'alpha': 1.7243522923151082,
'colsample_bytree': 0.7, 'subsample': 0.7, 'learning_rate': 0.02049865092427281,
'n_estimators': 300, 'max_depth': 10, 'min_child_weight': 85}. Best is trial 33 with
0.07578849177547532.
[I 2025-11-04 05:54:13,815] Trial 44 finished with value: 0.07586812976502903 and
parameters: {'lambda': 0.19665252767242905, 'alpha': 0.0031039285312874027,
'colsample_bytree': 0.5, 'subsample': 0.8, 'learning_rate': 0.040663541359820955,
'n_estimators': 400, 'max_depth': 8, 'min_child_weight': 40}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:15,632] Trial 45 finished with value: 0.07592193079854721 and
parameters: {'lambda': 0.046041493449869964, 'alpha': 0.03883007800897363,
'colsample_bytree': 1.0, 'subsample': 0.5, 'learning_rate': 0.03330975136617626,
'n_estimators': 200, 'max_depth': 7, 'min_child_weight': 63}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:19,840] Trial 46 finished with value: 0.07583629352338757 and
parameters: {'lambda': 0.5521103589080509, 'alpha': 5.8874308650778, 'colsample_bytre
0.7, 'subsample': 1.0, 'learning_rate': 0.047269973348589865, 'n_estimators': 400,
'max_depth': 9, 'min_child_weight': 103}. Best is trial 33 with value: 0.075788491775
[I 2025-11-04 05:54:25,362] Trial 47 finished with value: 0.0758437530395798 and
parameters: {'lambda': 0.005265037455041876, 'alpha': 2.897404766530089,
'colsample_bytree': 1.0, 'subsample': 0.8, 'learning_rate': 0.029525385304036828,
'n_estimators': 600, 'max_depth': 8, 'min_child_weight': 20}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:29,295] Trial 48 finished with value: 0.09567957940376398 and
parameters: {'lambda': 0.018879265936111604, 'alpha': 0.5022157334697028,
'colsample_bytree': 0.3, 'subsample': 0.7, 'learning_rate': 0.007568950168075951,
'n_estimators': 300, 'max_depth': 9, 'min_child_weight': 47}. Best is trial 33 with v
0.07578849177547532.
[I 2025-11-04 05:54:32,568] Trial 49 finished with value: 0.07595048315338805 and
parameters: {'lambda': 0.0027696962402860075, 'alpha': 0.2034783671889947,
'colsample_bytree': 0.9, 'subsample': 0.8, 'learning_rate': 0.022652146870211196,
'n_estimators': 200, 'max_depth': 10, 'min_child_weight': 129}. Best is trial 33 with
value: 0.07578849177547532.
[I 2025-11-04 05:54:32,570] A new study created in memory with name: no-name-7b2549b2
a268-4757-917d-93551d012a95
```

✅ Best XGBRegressor parameters: {'lambda': 0.0590637614665525, 'alpha': 2.629539230
'colsample_bytree': 1.0, 'subsample': 0.8, 'learning_rate': 0.032276672890830076,
'n_estimators': 300, 'max_depth': 8, 'min_child_weight': 66}

🚀 Tuning LGBMRegressor (GPU)...

```
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
1 warning generated.
[I 2025-11-04 05:54:45,388] Trial 0 finished with value: 0.07619856078959096 and
parameters: {'lambda_l1': 0.007030465188174908, 'lambda_l2': 0.0006581048646428316,
'num_leaves': 144, 'feature_fraction': 0.5266316815819165, 'bagging_fraction':
0.9213697071124759, 'bagging_freq': 1, 'min_child_samples': 61, 'learning_rate':
0.033248979198644404, 'n_estimators': 200, 'max_depth': 9}. Best is trial 0 with valu
0.07619856078959096.
[I 2025-11-04 05:55:06,026] Trial 1 finished with value: 0.08297633529333712 and
parameters: {'lambda_l1': 0.10444765546422762, 'lambda_l2': 0.0039838502405088695,
'num_leaves': 113, 'feature_fraction': 0.825504854405926, 'bagging_fraction':
0.8474886272032789, 'bagging_freq': 5, 'min_child_samples': 13, 'learning_rate':
0.2995417634807788, 'n_estimators': 700, 'max_depth': 11}. Best is trial 0 with value
0.07619856078959096.
[I 2025-11-04 05:55:22,786] Trial 2 finished with value: 0.07624820963236877 and
parameters: {'lambda_l1': 0.006801355235208913, 'lambda_l2': 5.288223513170339e-05,
'num_leaves': 99, 'feature_fraction': 0.9718000188097597, 'bagging_fraction':
0.9079158876343449, 'bagging_freq': 7, 'min_child_samples': 24, 'learning_rate':
0.031254183634088634, 'n_estimators': 700, 'max_depth': 4}. Best is trial 0 with valu
0.07619856078959096.
[I 2025-11-04 05:55:43,373] Trial 3 finished with value: 0.0760341839305434 and paran
{'lambda_l1': 0.0001653138146694767, 'lambda_l2': 0.6694689968875535, 'num_leaves': 1
'feature_fraction': 0.5895624561459303, 'bagging_fraction': 0.7725179197683131,
'bagging_freq': 2, 'min_child_samples': 77, 'learning_rate': 0.06260002703075547,
'n_estimators': 600, 'max_depth': 5}. Best is trial 3 with value: 0.0760341839305434.
[I 2025-11-04 05:55:48,710] Trial 4 finished with value: 0.0770667767388295 and paran
{'lambda_l1': 0.5138858994189014, 'lambda_l2': 0.005849601831341976, 'num_leaves': 23
'feature_fraction': 0.8120933857331546, 'bagging_fraction': 0.6359613505965713,
'bagging_freq': 2, 'min_child_samples': 63, 'learning_rate': 0.13290366768080092,
```

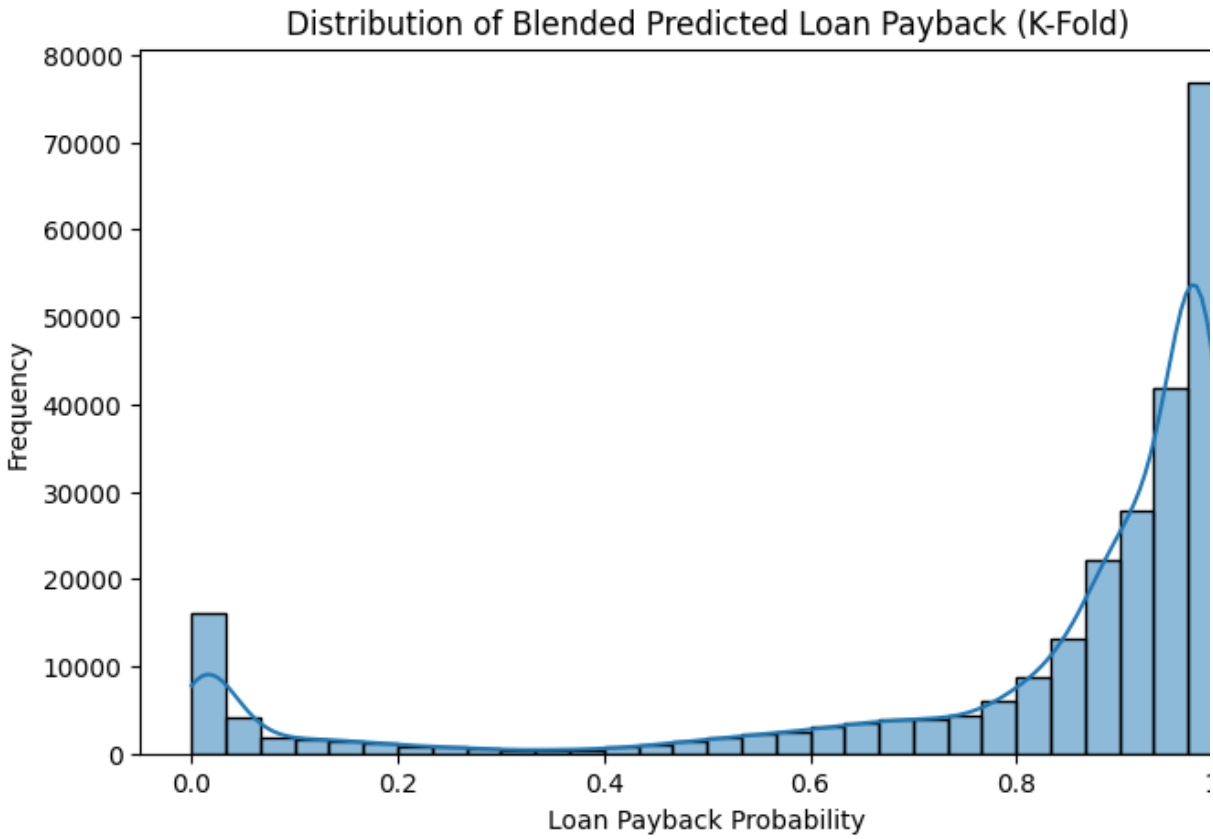
```
'n_estimators': 200, 'max_depth': 3}. Best is trial 3 with value: 0.0760341839305434.
[I 2025-11-04 05:56:18,717] Trial 5 finished with value: 0.07581688670996149 and
parameters: {'lambda_l1': 6.432443301130878e-05, 'lambda_l2': 0.2422491918497322,
'num_leaves': 203, 'feature_fraction': 0.7934450301476234, 'bagging_fraction':
0.6712179457852541, 'bagging_freq': 6, 'min_child_samples': 57, 'learning_rate':
0.015268971306524486, 'n_estimators': 500, 'max_depth': 9}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:56:25,994] Trial 6 finished with value: 0.08055551035024425 and
parameters: {'lambda_l1': 0.5180808512650436, 'lambda_l2': 0.0006451268020397896,
'num_leaves': 125, 'feature_fraction': 0.8894792580605522, 'bagging_fraction':
0.9695513783189307, 'bagging_freq': 5, 'min_child_samples': 80, 'learning_rate':
0.009942529989036821, 'n_estimators': 300, 'max_depth': 3}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:56:46,923] Trial 7 finished with value: 0.07920433333865375 and
parameters: {'lambda_l1': 0.04403920000631578, 'lambda_l2': 0.8973069331399319,
'num_leaves': 240, 'feature_fraction': 0.6233421874133424, 'bagging_fraction':
0.6724505699439914, 'bagging_freq': 4, 'min_child_samples': 71, 'learning_rate':
0.00849576258312716, 'n_estimators': 300, 'max_depth': 12}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:57:00,540] Trial 8 finished with value: 0.07607272729947627 and
parameters: {'lambda_l1': 0.02471742174904267, 'lambda_l2': 0.011733474766235428,
'num_leaves': 73, 'feature_fraction': 0.9612220969719796, 'bagging_fraction':
0.8127389053568403, 'bagging_freq': 5, 'min_child_samples': 36, 'learning_rate':
0.16086593458012793, 'n_estimators': 600, 'max_depth': 4}. Best is trial 5 with value
0.07581688670996149.
[I 2025-11-04 05:57:21,271] Trial 9 finished with value: 0.07734569544493893 and
parameters: {'lambda_l1': 0.0010318879110336001, 'lambda_l2': 9.464720553888432e-05,
'num_leaves': 54, 'feature_fraction': 0.666000033605566, 'bagging_fraction':
0.5175793120117966, 'bagging_freq': 1, 'min_child_samples': 89, 'learning_rate':
0.005048389748617261, 'n_estimators': 800, 'max_depth': 5}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:58:09,586] Trial 10 finished with value: 0.0758302690096769 and
parameters: {'lambda_l1': 1.4813249393556876e-05, 'lambda_l2': 0.07255819787476929,
'num_leaves': 189, 'feature_fraction': 0.7459569164861799, 'bagging_fraction':
0.516252279453441, 'bagging_freq': 7, 'min_child_samples': 45, 'learning_rate':
0.01469968720883134, 'n_estimators': 1000, 'max_depth': 8}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:58:57,731] Trial 11 finished with value: 0.07586291184935591 and
parameters: {'lambda_l1': 2.376695559813349e-05, 'lambda_l2': 0.06263546940269317,
'num_leaves': 188, 'feature_fraction': 0.7348343287984969, 'bagging_fraction':
0.5003201599767069, 'bagging_freq': 7, 'min_child_samples': 43, 'learning_rate':
0.01646615886893902, 'n_estimators': 1000, 'max_depth': 8}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 05:59:59,191] Trial 12 finished with value: 0.07592255434594943 and
parameters: {'lambda_l1': 1.3187925183397903e-05, 'lambda_l2': 0.09407459737699685,
'num_leaves': 197, 'feature_fraction': 0.747053768771999, 'bagging_fraction':
0.6122985812421822, 'bagging_freq': 6, 'min_child_samples': 48, 'learning_rate':
0.01849666658728204, 'n_estimators': 1000, 'max_depth': 10}. Best is trial 5 with val
0.07581688670996149.
[I 2025-11-04 06:00:13,454] Trial 13 finished with value: 0.07616393134670534 and
parameters: {'lambda_l1': 0.00016775247877969969, 'lambda_l2': 0.06976450435618947,
'num_leaves': 21, 'feature_fraction': 0.8298977718533178, 'bagging_fraction':
0.710757291715795, 'bagging_freq': 7, 'min_child_samples': 29, 'learning_rate':
0.018542821360925627, 'n_estimators': 500, 'max_depth': 7}. Best is trial 5 with valu
0.07581688670996149.
[I 2025-11-04 06:00:28,795] Trial 14 finished with value: 0.07604142125369763 and
parameters: {'lambda_l1': 6.348829258005792e-05, 'lambda_l2': 0.1695527083903794,
'num_leaves': 161, 'feature_fraction': 0.6870049394460855, 'bagging_fraction':
0.563881636503866, 'bagging_freq': 6, 'min_child_samples': 53, 'learning_rate':
0.053926313413008665, 'n_estimators': 400, 'max_depth': 7}. Best is trial 5 with valu
```

```
0.07581688670996149.
[I 2025-11-04 06:01:22,026] Trial 15 finished with value: 0.07579051525572111 and
parameters: {'lambda_l1': 0.0007012463598366529, 'lambda_l2': 0.02098683225691074,
'num_leaves': 202, 'feature_fraction': 0.8768724727385411, 'bagging_fraction':
0.584380330157251, 'bagging_freq': 6, 'min_child_samples': 96, 'learning_rate':
0.010139111634202886, 'n_estimators': 900, 'max_depth': 9}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:02:19,341] Trial 16 finished with value: 0.07597315177201917 and
parameters: {'lambda_l1': 0.0009209129081901704, 'lambda_l2': 0.026448011209989457,
'num_leaves': 218, 'feature_fraction': 0.8663266565141845, 'bagging_fraction':
0.7219371374049558, 'bagging_freq': 4, 'min_child_samples': 96, 'learning_rate':
0.005053445544685155, 'n_estimators': 800, 'max_depth': 10}. Best is trial 15 with va
0.07579051525572111.
[I 2025-11-04 06:02:49,725] Trial 17 finished with value: 0.07583796806745981 and
parameters: {'lambda_l1': 0.0010554021573794452, 'lambda_l2': 0.0012743785816041228,
'num_leaves': 209, 'feature_fraction': 0.9224305814005653, 'bagging_fraction':
0.6051824631812123, 'bagging_freq': 6, 'min_child_samples': 100, 'learning_rate':
0.011232307241937446, 'n_estimators': 500, 'max_depth': 9}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:03:24,494] Trial 18 finished with value: 0.07585708359779844 and
parameters: {'lambda_l1': 0.00020829013243735397, 'lambda_l2': 1.1160208489883794e-05
'num_leaves': 246, 'feature_fraction': 0.813058806387244, 'bagging_fraction':
0.6544205271302578, 'bagging_freq': 3, 'min_child_samples': 67, 'learning_rate':
0.02600737674499606, 'n_estimators': 900, 'max_depth': 6}. Best is trial 15 with valu
0.07579051525572111.
[I 2025-11-04 06:03:53,040] Trial 19 finished with value: 0.07602318754457024 and
parameters: {'lambda_l1': 0.0004752094718356869, 'lambda_l2': 0.19926638343374395,
'num_leaves': 153, 'feature_fraction': 0.9188091094659171, 'bagging_fraction':
0.5706168493873365, 'bagging_freq': 4, 'min_child_samples': 85, 'learning_rate':
0.00752033112089622, 'n_estimators': 500, 'max_depth': 12}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:04:36,595] Trial 20 finished with value: 0.07728610100727944 and
parameters: {'lambda_l1': 4.919021201747721e-05, 'lambda_l2': 0.01721941303936059,
'num_leaves': 256, 'feature_fraction': 0.7876228656215017, 'bagging_fraction':
0.6999343892964164, 'bagging_freq': 6, 'min_child_samples': 11, 'learning_rate':
0.05816633221640164, 'n_estimators': 700, 'max_depth': 10}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:05:22,422] Trial 21 finished with value: 0.07579161982304947 and
parameters: {'lambda_l1': 3.810065958738492e-05, 'lambda_l2': 0.27449708613285023,
'num_leaves': 177, 'feature_fraction': 0.7110344992161641, 'bagging_fraction':
0.551961043100667, 'bagging_freq': 7, 'min_child_samples': 42, 'learning_rate':
0.013377158238422213, 'n_estimators': 900, 'max_depth': 8}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:06:13,149] Trial 22 finished with value: 0.07580123268143395 and
parameters: {'lambda_l1': 5.801922441665145e-05, 'lambda_l2': 0.26852110045121597,
'num_leaves': 174, 'feature_fraction': 0.7036596825466599, 'bagging_fraction':
0.5578656278455262, 'bagging_freq': 6, 'min_child_samples': 34, 'learning_rate':
0.01215042861740263, 'n_estimators': 900, 'max_depth': 9}. Best is trial 15 with valu
0.07579051525572111.
[I 2025-11-04 06:06:58,721] Trial 23 finished with value: 0.07591945030638433 and
parameters: {'lambda_l1': 0.000384417731059242, 'lambda_l2': 0.29252451911803556,
'num_leaves': 168, 'feature_fraction': 0.683650806293741, 'bagging_fraction':
0.5729036385563974, 'bagging_freq': 5, 'min_child_samples': 32, 'learning_rate':
0.023492337338808143, 'n_estimators': 900, 'max_depth': 8}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:07:38,540] Trial 24 finished with value: 0.07586057995046305 and
parameters: {'lambda_l1': 4.1237315728336215e-05, 'lambda_l2': 0.03222063161495835,
'num_leaves': 222, 'feature_fraction': 0.6226091337231356, 'bagging_fraction':
0.55329627943119, 'bagging_freq': 7, 'min_child_samples': 23, 'learning_rate':
0.011176335223591719, 'n_estimators': 900, 'max_depth': 7}. Best is trial 15 with val
```

```
0.07579051525572111.
[I 2025-11-04 06:08:21,389] Trial 25 finished with value: 0.07590803982317286 and
parameters: {'lambda_l1': 0.002657162502049993, 'lambda_l2': 0.49320720230722687,
'num_leaves': 136, 'feature_fraction': 0.6958780386089681, 'bagging_fraction':
0.6145365229063263, 'bagging_freq': 6, 'min_child_samples': 38, 'learning_rate':
0.006885154657703108, 'n_estimators': 800, 'max_depth': 11}. Best is trial 15 with va
0.07579051525572111.
[I 2025-11-04 06:09:12,942] Trial 26 finished with value: 0.07579435791241844 and
parameters: {'lambda_l1': 9.446726752276928e-05, 'lambda_l2': 0.00988172301743576,
'num_leaves': 176, 'feature_fraction': 0.7197751905625953, 'bagging_fraction':
0.5508052841601867, 'bagging_freq': 5, 'min_child_samples': 52, 'learning_rate':
0.012109176392562648, 'n_estimators': 900, 'max_depth': 9}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:09:42,841] Trial 27 finished with value: 0.07644766549933615 and
parameters: {'lambda_l1': 0.002588434042093448, 'lambda_l2': 0.005895008847609486,
'num_leaves': 178, 'feature_fraction': 0.6247196783117512, 'bagging_fraction':
0.5406108179437312, 'bagging_freq': 5, 'min_child_samples': 51, 'learning_rate':
0.006385490577557858, 'n_estimators': 800, 'max_depth': 6}. Best is trial 15 with val
0.07579051525572111.
[I 2025-11-04 06:10:42,493] Trial 28 finished with value: 0.07578312225172398 and
parameters: {'lambda_l1': 0.00010477299490708462, 'lambda_l2': 0.001683991135918689,
'num_leaves': 152, 'feature_fraction': 0.7736216825706469, 'bagging_fraction':
0.5986775279291205, 'bagging_freq': 3, 'min_child_samples': 70, 'learning_rate':
0.008894682337619 34, 'n_estimators': 1000, 'max_depth': 11}. Best is trial 28 with va
0.07578312225172398.
[I 2025-11-04 06:11:40,286] Trial 29 finished with value: 0.07595983950335267 and
parameters: {'lambda_l1': 0.0003357989075825222, 'lambda_l2': 0.0015590706141402056,
'num_leaves': 144, 'feature_fraction': 0.5285016982994512, 'bagging_fraction':
0.6010844345244366, 'bagging_freq': 3, 'min_child_samples': 92, 'learning_rate':
0.02348449979482323, 'n_estimators': 1000, 'max_depth': 11}. Best is trial 28 with va
0.07578312225172398.
[I 2025-11-04 06:12:29,342] Trial 30 finished with value: 0.07604493331015397 and
parameters: {'lambda_l1': 0.007012188645779636, 'lambda_l2': 0.0005628319308015488,
'num_leaves': 109, 'feature_fraction': 0.8586970316882998, 'bagging_fraction':
0.7617638644887628, 'bagging_freq': 3, 'min_child_samples': 74, 'learning_rate':
0.0328824657921986, 'n_estimators': 1000, 'max_depth': 10}. Best is trial 28 with val
0.07578312225172398.
[I 2025-11-04 06:13:24,965] Trial 31 finished with value: 0.07577843599974726 and
parameters: {'lambda_l1': 9.825672076489441e-05, 'lambda_l2': 0.0020369223966167417,
'num_leaves': 158, 'feature_fraction': 0.7772955367321573, 'bagging_fraction':
0.5320452477914508, 'bagging_freq': 2, 'min_child_samples': 64, 'learning_rate':
0.007833621248950956, 'n_estimators': 900, 'max_depth': 9}. Best is trial 31 with val
0.07577843599974726.
[I 2025-11-04 06:14:18,114] Trial 32 finished with value: 0.07580800208074738 and
parameters: {'lambda_l1': 2.000360580886949e-05, 'lambda_l2': 0.0018752619027780134,
'num_leaves': 131, 'feature_fraction': 0.7683365496885769, 'bagging_fraction':
0.592849624459086, 'bagging_freq': 2, 'min_child_samples': 66, 'learning_rate':
0.008198593952758371, 'n_estimators': 900, 'max_depth': 11}. Best is trial 31 with va
0.07577843599974726.
[I 2025-11-04 06:14:49,821] Trial 33 finished with value: 0.07579407241055634 and
parameters: {'lambda_l1': 0.00011599698850856678, 'lambda_l2': 0.0026173751899320296,
'num_leaves': 149, 'feature_fraction': 0.8532310622287435, 'bagging_fraction':
0.6403497962078055, 'bagging_freq': 1, 'min_child_samples': 59, 'learning_rate':
0.009041661282313828, 'n_estimators': 700, 'max_depth': 8}. Best is trial 31 with val
0.07577843599974726.
```

```
✅ Best LGBMRegressor parameters: {'lambda_l1': 9.825672076489441e-05, 'lambda_l2':
0.0020369223966167417, 'num_leaves': 158, 'feature_fraction': 0.7772955367321573,
'bagging_fraction': 0.5320452477914508, 'bagging_freq': 2, 'min_child_samples': 64,
'learning_rate': 0.007833621248950956, 'n_estimators': 900, 'max_depth': 9}

🚀 Starting 10-Fold Cross-Validation for robust submission...
--- Fold 1/10 ---
Training XGB...
Training LGBM...
--- Fold 2/10 ---
Training XGB...
Training LGBM...
--- Fold 3/10 ---
Training XGB...
Training LGBM...
--- Fold 4/10 ---
Training XGB...
Training LGBM...
--- Fold 5/10 ---
Training XGB...
Training LGBM...
--- Fold 6/10 ---
Training XGB...
Training LGBM...
--- Fold 7/10 ---
Training XGB...
Training LGBM...
--- Fold 8/10 ---
Training XGB...
Training LGBM...
--- Fold 9/10 ---
Training XGB...
Training LGBM...
--- Fold 10/10 ---
Training XGB...
Training LGBM...

--- OOF Blending Grid Search ---
OOF XGB Regressor AUC: 0.910354
OOF LGBM Regressor AUC: 0.910451

✅ Best OOF Blend AUC: 0.910555
   -> Best Weight (for LGBM): 0.58

💡 Generating blended submission with best weight...

✅ Submission file 'submission_v7_kfold_blend.csv' generated successfully!
```

	id	loan_paid_back
0	593994	0.946277
1	593995	0.957180
2	593996	0.507415
3	593997	0.963145
4	593998	0.981638


Distribution of Blended Predicted Loan Payback (K-Fold)
Frequency
80000
70000
60000
50000
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8
Loan Payback Probability