Credit Scoring: Creditworthiness Prediction

This project explores a synthetic credit scoring dataset from Kaggle.
The goal is to predict customer creditworthiness based on basic demographic and financial information.

Data

The dataset contains 12 input features and a binary target:

- Numerical: Age, Income, Debt, Credit_Score, Loan_Amount, Loan_Term, Num_Credit_Cards
- Categorical: Gender, Education, Payment_History, Employment_Status, Residence_Type, Marital_Status
- Target: Creditworthiness (1 – creditworthy, 0 – non-creditworthy)

The classes are moderately imbalanced: about 70% creditworthy vs 30% non-creditworthy.

Workflow

1. **Exploratory Data Analysis (EDA) + Logistic Regression baseline** (`notebooks/eda.ipynb`)
    - Inspected feature distributions and basic correlations.
    - Checked class balance and discussed why accuracy is a misleading metric for imbalanced data.
    - Built a baseline logistic regression model using a `Pipeline`:
        - `ColumnTransformer` with `StandardScaler` for numerical features and `OneHotEncoder` for categorical features.
        - Tried logistic regression with and without `class_weight="balanced"`.
    - Result: accuracy ≈ 0.70, but ROC-AUC ≈ 0.5 and recall for class 0 is essentially 0 (the model almost always predicts class 1).

2. **Tree-based baseline: RandomForestClassifier** (`notebooks/models_baseline.ipynb`)
    - Reused the same preprocessing pipeline (scaling + one-hot encoding).
    - Trained `RandomForestClassifier` with class weighting to handle imbalance.
    - Evaluated accuracy, ROC-AUC and classification report.
    - Observation: RandomForest behaved similarly to logistic regression:
        - accuracy remained high (~0.70),
        - ROC-AUC stayed close to 0.5,
        - the model still struggled to correctly identify non-creditworthy clients (class 0).

3. **LightGBM + feature engineering + feature importance** (`notebooks/model_lightGBM.ipynb`)
    - Added domain-motivated features:
        - `Debt_to_Income = Debt / Income`
        - `Credit_Utilization = Debt / Loan_Amount` (with protection against division by zero)

- Extended the numeric feature list and trained `LGBMClassifier` inside the same preprocessing pipeline.
- Used LightGBM feature importances to understand which variables drive the model:
  - Most important: Credit_Score, Loan_Amount, Income, Credit_Utilization, Debt, Age, Debt_to_Income.
- Built a reduced model:
  - kept only the most informative numerical features and a subset of categorical ones (e.g. Payment_History, Employment_Status, Residence_Type, Marital_Status),
  - removed less informative or sensitive features such as Gender.
- Result of the reduced LightGBM model:
  - accuracy dropped to ~0.60,
  - recall for class 0 improved to around 0.30 (the model finally detects some non-creditworthy customers),
  - ROC-AUC still around 0.5, indicating that global ranking quality remains limited.

Key takeaways

- Different algorithms (logistic regression, RandomForest, LightGBM), class weighting and simple feature engineering all converge to ROC-AUC ≈ 0.5 on this synthetic dataset.
- Models that optimize only accuracy end up predicting almost exclusively the majority class, which is unacceptable for credit risk problems.
- LightGBM with engineered features and a reduced set of predictors trades accuracy for better minority-class recall, which is more realistic in a risk-management context.
- The main limitation appears to be the weak signal in the synthetic dataset rather than the choice of algorithm.
  Recognizing such limitations and documenting them clearly is an important part of responsible model development.

Files

- `notebooks/eda.ipynb` – exploratory data analysis and a first baseline model (logistic regression with class weighting).
- `notebooks/models_baseline.ipynb` – tree-based baseline model (`RandomForestClassifier`) using the same preprocessing pipeline.
- `notebooks/model_lightGBM.ipynb` – gradient boosting model (LightGBM) with engineered features, feature importance analysis and reduced feature set experiments.