CREDIT RISK MANAGEMENT BASED ON DECISION TREE MODEL

GROUP NAME: JULIA, HAJIME, MIZUKI, YUKI

RESEARCH PAPER SUMMARY

- **Paper:** Credit Risk Management Based on Decision Tree Model
- **Authors:** Yutian Gan, Han Luo, Wei Wei
- **Focus:** Predicting credit card default using Decision Tree classifier
- **Dataset:** Default of Credit Card Clients (UCI Repository)

PROBLEM DEFINITION

- **Challenge:** Accurately predicting credit card defaults
- **Importance:**
  - Reduces financial risk exposure
  - Supports informed lending decisions
- **Objective:**
  - Develop an interpretable, accurate classification model

DATASET DESCRIPTION

INFORMATIONS

- **Source:** UCI ML Repository (via Kaggle)
- **Period:** April–Sept 2005 (Taiwan credit card clients)
- **Size:** 30,000 records, 25 variables
- **Target:** default payment next month
- **Features:** Demographics, Credit Info, Payment History, Bills, Payments

TABLES

- **Demographics** (SEX, EDUCATION, MARRIAGE, AGE)
- **Credit Information**(LIMIT_BAL)
- **Payment history** (PAY_1–PAY_6)
- **Bill amounts** (BILL_AMT1–6)
- **Payment amounts** (PAY_AMT1–6)

APPLYING THE PAPER'S METHODOLOGY TO A SIMILAR DATASET

Dataset: HMEQ (Home Equity) Data

Overview:

- Baseline and credit performance information for 5,960 recent home equity loans.
- Used to assess credit risk and automate decision-making for loan approval.

Objective:

- Binary Classification: Predict whether a borrower will default on their loan.
- Minimize financial loss by accurately flagging high-risk loans.

APPLYING THE PAPER'S METHODOLOGY TO A SIMILAR DATASET

Data Volume:

- Rows: 5,960 instances
- Columns: 13 variables (1 Target + 12 Features)

Target Variable: BAD

- 1 = Client defaulted
- 0 = Loan repaid

Key Features:

- Financial: Loan Amount (LOAN), Value of Property (VALUE), Debt-to-Income Ratio (DEBTINC).
- Personal: Years on Job (YOJ), Job Type (JOB).

DATA PREPROCESSING

Missing Values:

- Created binary flags to capture missingness patterns.
- Imputation:
  - Categorical: Filled with "Unknown" (new category).
  - Numerical: Filled with Median or 0.

Outlier:

- Capped variables at the 99th percentile.

Other Preprocessing:

- One-Hot Encoding and Standard Scaling are applied.

MODEL BUILDING

DECISION TREE

```
# Parameters for grid search
param_grid = {
    'max_depth': [3, 5, 7, 10, 15, 20, None],
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 5, 10],
    'criterion': ['gini', 'entropy'],
    'class_weight': [None, 'balanced'] + [{0: 1, 1: w} for w in [1, 2, 3, 5, 10]]
}

model = DecisionTreeClassifier(random_state=42)
grid_search = GridSearchCV(
    model,
    param_grid,
    cv=10,
    scoring='f1',
    n_jobs=-1,
    verbose=1
)
grid_search.fit(X_train_scaled, y_train)

best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test_scaled)
y_pred_proba = best_model.predict_proba(X_test_scaled)[:, 1]
```

PERFORMANCE COMPARISON

ORIGINAL DATASET

- Accuracy: 78.1%
- Precision: 50.4%
- Recall: 55.0%
- F1-score: 52.6%
- ROC-AUC: 74.8%

NEW DATASET

- Accuracy: 87.6%
- Precision: 68.9%
- Recall: 68.9%
- F1-score: 68.9%
- ROC-AUC: 79.0%

MISSING VALUES

ORIGINAL DATASET

```
ID                          0
LIMIT_BAL                   0
SEX                         0
EDUCATION                   0
MARRIAGE                    0
AGE                         0
PAY_0                       0
PAY_2                       0
PAY_3                       0
PAY_4                       0
PAY_5                       0
PAY_6                       0
BILL_AMT1                   0
BILL_AMT2                   0
BILL_AMT3                   0
BILL_AMT4                   0
BILL_AMT5                   0
BILL_AMT6                   0
PAY_AMT1                    0
PAY_AMT2                    0
PAY_AMT3                    0
PAY_AMT4                    0
PAY_AMT5                    0
PAY_AMT6                    0
default.payment.next.month  0
dtype: int64
```

NEW DATASET

	0
BAD	0
LOAN	0
MORTDUE	518
VALUE	112
REASON	252
JOB	279
YOJ	515
DEROG	708
DELINQ	580
CLAGE	308
NINQ	510
CLNO	222
DEBTINC	1267

dtype: int64

DEFAULT RATE BY COLUMN IN THE NEW DATASET

	row with missing values	row without missing values
YOJ	12.62%	20.64%
DEROG	12.29%	20.98%
DELINQ	12.41%	20.76%
CLAGE	25.32%	19.66%
NINQ	14.71%	20.44%
CLNO	23.87%	19.80%
DEBTINC	62.04%	8.59%

CONTRIBUTION 2

ADVANCED HYPERPARAMETER TUNING

ORIGINAL PARAMETER

- max_depth
- min_samples_leaf
- min_samples_spliit
- criterion
- class_weight

ADDED PARAMETER

- ccp_alpha
- max_features
- min_inpurity_decrease
- splitter

BEFORE

- Accuracy: 0.781
- Precision: 0.504
- Recall: 0.550
- F1-score: 0.526
- ROC-AUC: 0.748

AFTER

- Accuracy: 0.772 (-0.9%)
- Precision: 0.488 (-1.6%)
- Recall: 0.578 (+2.8%)
- F1-score: 0.529 (+0.3%)
- ROC-AUC: 0.759 (+1.1%)

MODEL COMPARISON SETUP

- Models compared: **Decision Tree vs. XGBoost**
- Same dataset and same preprocessing for both models
- Same train/test split (**70% / 30%**, stratified)
- Class imbalance handled in both models
  - Decision Tree: class weights
  - XGBoost: scale_pos_weight

PERFORMANCE RESULTS

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Decision Tree	0.781	0.504	0.550	0.526	0.748
XGBoost	0.793	0.530	0.560	0.545	0.777

- **Observation:**

XGBoost performs slightly better across all metrics

CONFUSION MATRIX & INTERPRETATION

- XGBoost identifies more defaulters
- Slight reduction in false positives and false negatives

Confusion Matrix (TP – Defaulters):

- Decision Tree: 1087
- XGBoost: 1108

Conclusion:

XGBoost provides more accurate and balanced predictions than a single Decision Tree on the same dataset.

MODEL COMPARISON SETUP

- Models compared: **Decision Tree vs. LightBGM**
- Same set up to ensure fair comparison
- Class imbalance handled in both models
  - Decision Tree: class weights
  - LightBGM: scale_pos_weight
- Hyperparameter tuning via GridSearchCV (10-fold CV)

PERFORMANCE RESULTS

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Decision Tree	0.781	0.504	0.550	0.526	0.748
LightBGM	0.789	0.521	0.582	0.550	0.779
Improvements	+1.0%	+3.4%	+5.8%	+4.6%	+4.1%

- Observation: LightGBM is the better model for this credit risk task, particularly because it reduces the chance of missing actual defaulters (i.e. lower false negative.)

CONFUSION MATRIX & INTERPRETATION

True Positives (Defaults Caught):

- Decision Tree: 1087
- LightGBM: 1150 → +63 improvement

False Negatives (Defaults Missed):

- Decision Tree: 890
- LightGBM: 827 → 7% reduction

Conclusion:

LightGBM is better at detecting true defaulters and reduces the risk of missing high-risk customers, offering better protection against financial losses.

CHALLENGES

1. Methodology Adaptation

- Applying the reference paper's work to a completely different dataset (HMEQ) proved difficult.
- The pipeline had to be significantly modified to handle HMEQ-specific issues (e.g., heavy missing data) that differed from the original study.

2. Advanced Algorithm Adoption

- Self-learning and implementing XGBoost & LightGBM from scratch.
- Mastering complex hyperparameter tuning, which requires a deeper understanding of gradient boosting compared to simple Decision Trees.

FUTURE SCOPE

1. Validation on Modern Data

- Testing on recent data is crucial to ensure the model works under modern economic conditions.

2. Deep Learning Integration

- Implement Deep Learning models to capture complex, non-linear patterns that standard Tree-based models might overlook, aiming for higher accuracy.

3. Advanced Feature Engineering

- Create domain-specific metrics, such as "Credit Utilization Ratio" (Balance / Limit).
- Explore interactions between variables to detect risk segments that single variables miss.

HAPPY HOLIDAYS!