CORNERSTONE
COMMUNITY COLLEGE

DS 301 - Machine Learning
Final Project Report

Hajime - 2024013622
Julia - 2024013814
Mizuki - 2025015002
Yuki - 2024014195

Summary
The primary objective of this project is to reproduce and validate the methodology presented in the research paper "Credit Risk Management Based on Decision Tree Model" by Yutian Gan et al. The project focuses on predicting credit card defaults using the "Default of Credit Card Clients" dataset from the UCI Machine Learning Repository. We aimed to replicate the authors' data preprocessing steps, specifically the feature engineering of demographic variables, and their Decision Tree classification model. Furthermore, we extended the scope of the study by experimenting with algorithms not covered in the paper and performing a comparison of the results using similar datasets.

Motivation
Credit risk management is a critical challenge for financial institutions, where accurately predicting defaults can significantly reduce financial exposure and support informed lending decisions. We selected this specific research paper because it utilizes Decision Trees, a "white-box" model that offers high interpretability compared to more complex algorithms. Our primary motivation was to verify whether the proposed methodology—combining interpretable models with specific preprocessing strategies—can be reproduced and meaningfully applied to credit risk prediction.

Research Paper Details

- **Title:** Credit Risk Management Based on Decision Tree Model
- **Authors:** Yutian Gan, Han Luo, Wei Wei
- **Objective:** To develop an accurate and interpretable classification model for predicting the probability of credit card default.
- **Methodology:** The paper proposes using a Decision Tree classifier. A key aspect of their methodology involves detailed analysis of demographic features and payment history to identify risk factors.
- **Key Findings:** The paper suggests that specific demographic combinations (e.g., age groups and marital status) have varying impacts on default risk, and that a properly tuned Decision Tree can effectively capture these non-linear relationships.

Dataset Details

We utilized the dataset specified in the research paper, obtained from the UCI Machine Learning Repository.

- **Dataset Name:** Default of Credit Card Clients
- **Source:** Taiwan credit card clients (April - September 2005).
- **Volume:** The dataset consists of 30,000 records and 25 variables.
- **Target Variable:** default payment next month (Binary: 1 for default, 0 for non-default).
- **Features:**
    - **Demographics:** SEX, EDUCATION, MARRIAGE, AGE.
    - **Financial History:** LIMIT_BAL (Credit Limit).
    - **Repayment Status:** PAY_1 through PAY_6 (repayment status from September to April).
    - **Bill & Payment Amounts:** BILL_AMT1-6 and PAY_AMT1-6.

Data Preprocessing and Feature Engineering

Following the steps outlined in the paper and our initial analysis, we performed the following preprocessing steps:

1. **Combined Features:** We created a new categorical feature GENDER_MARRIAGE by combining SEX and MARRIAGE. This resulted in 6 initial categories to capture interaction effects between gender and marital status.
2. **Data Cleaning:** We identified that the category for "Divorced Women" had an extremely small sample size. To avoid noise and potential overfitting due to class imbalance within this subgroup, we excluded these records from the dataset.
3. **Encoding:** Categorical variables were encoded to be suitable for multiple models.
4. **Handling Class Imbalance:** We observed a default rate of 22.1%, indicating an imbalanced dataset. We addressed this by implementing class weights in the model training phase to prevent the model from biasing towards the majority class (non-default).

Steps Reproduced from the Paper

We successfully reproduced the core workflow described in the research paper:

1. **Model Implementation:** We built a Decision Tree Classifier using scikit-learn.
2. **Hyperparameter Tuning:** We reproduced the optimization process using Grid Search. We tuned parameters including max_depth, min_samples_split, min_samples_leaf, and criterion (Gini vs. Entropy).
3. **Model Evaluation:** We tested various tree depths. Consistent with general decision tree behavior, we found that a moderate depth provided the best balance between accuracy and generalization.

Contributions

1. Applying the research paper's methodology to a similar dataset

We used the HMEQ dataset, created models similar to those in a research paper, and interpreted the results.

A. **Similar Dataset Project Overview**

This project aims to build and evaluate classification models to predict whether customers will default (non-performing loan) using a mortgage loan dataset (hmeq.csv). The variable targeted for prediction is "BAD" (a binary variable indicating the presence or absence of default).

B. **Data Preprocessing**

Due to the presence of numerous missing values and outliers in the dataset, the following procedures were implemented to maximize model performance.

a. **Missing Value Identification and Flag Creation**

i. **Process**: Flag columns (e.g., YOJ_MISSING) were created for numerical variables that contained missing values.

ii. **Reason for Selection**: The fact that data is missing may itself be critical information for predicting customer default risk (e.g., insufficient information for loan screening). This flagging was done to allow the model to utilize this information.

b. **Missing Value Imputation**

Categorical Variables (REASON, JOB):

i. **Process**: Missing values were imputed with a new category, **"Unknown"**.

ii. **Reason for Selection**: Instead of simply ignoring missing values, treating them as an independent category allows the model to learn the specific tendency of the cases where the data is missing.

Numerical Variables (LOAN, MORTDUE, VALUE, CLAGE, CLNO, DEBTINC):

i. **Process**: Missing values were imputed with the **Median**.

ii. **Reason for Selection**: Unlike the mean, the median is **robust** against the influence of extreme outliers present in the data. This prevents the compensation from significantly distorting the data distribution and stabilizes subsequent model learning.

Other Numerical Variables (YOJ, DEROG, DELINQ, NINQ):

i. **Process**: Missing values were imputed with **0**.

ii. **Reason for Selection**: Based on domain knowledge, it is highly likely that missing values in these variables—such as "Years on Job (YOJ)" or "Number of Past Delinquencies (DELINQ)"—can be interpreted as meaning "zero" (e.g., missing YOJ might mean 0 years of employment, or missing DELINQ might mean no past delinquencies).

c. **Outlier Treatment**

i. **Process**: **99th percentile clipping (Winsorization)** was applied to numerical variables such as LOAN, MORTDUE, VALUE, YOJ, CLAGE, NINQ, and CLNO to adjust extreme outliers.

ii. **Reason for Selection**: Linear-based models like Logistic Regression and SVM, as well as StandardScaler, are highly sensitive to extreme outliers. Clipping helps to suppress the impact of a small number of extreme data points that could destabilize model training, thereby

aiming to improve the model's generalization performance without drastically changing the overall data distribution.

d. **Other Preprocessing**
   i. **Encoding and Scaling**: One-Hot Encoding (OHE) was applied to categorical variables, and StandardScaler was used to standardize all numerical features.

C. **Model Building and Evaluation**

The data was split into a training set (70%) and a test set (30%), and the following three classification models were built:

- Decision Tree
- Logistic Regression
- Support Vector Machine (SVM)

For the Decision Tree model specifically, the following methodology was applied:

a. **Definition of Tuning Range (param_grid)**
   A wide range of values for the main hyperparameters to be explored was defined.
   i. **Parameters to be Explored**:
      - max_depth (Maximum depth)
      - min_samples_split (Minimum samples required to split a node)
      - min_samples_leaf (Minimum samples required to be at a leaf node)
      - criterion (Splitting criterion: Gini impurity or entropy)
      - class_weight (Weighting for handling class imbalance)

b. **Execution of Tuning Method (GridSearchCV)**
   An exhaustive search (Grid Search) was performed within the defined parameter range.
      - **Model**: DecisionTreeClassifier (Decision Tree Classifier)
      - **Evaluation Method**: **10-fold Cross-Validation** (cv=10) was used to robustly evaluate the performance of each parameter set.

- **Optimization Metric**: By setting scoring='f1', the parameter combination that yields the highest **F1 score** is selected as the optimum solution (this is especially effective when dealing with class imbalance).

2. Advanced hyperparameter tuning for the Decision Tree model

We performed further hyperparameter tuning to optimize the model's generalization performance and prevent overfitting. We utilized GridSearchCV to explore the best combination.

Added parameter:

- ccp_alpha: Controls the complexity of the tree by applying Minimal Cost-Complexity Pruning to remove irrelevant branches and prevent overfitting.
- max_features: Determines the maximum number of features considered when looking for the best split at each node, introducing randomness to reduce model variance.
- min_inpurity_decrease: Sets a threshold where a node will only split if it induces a decrease in impurity greater than or equal to this specific value.
- splitter: Specifies the strategy used to choose the split at each node, allowing a choice between the "best" split and a "random" split to diversify the model.

Performance comparison:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Original	0.781	0.504	0.550	0.526	0.748
Tuned	0.772	0.488	0.578	0.529	0.759

Observations

The hyperparameter tuning, optimized for the F1-score via GridSearchCV, successfully yielded a higher F1-score and significantly improved Recall to 0.578. While there was a minor trade-off in Accuracy and Precision, the increased Recall indicates the model is now more effective at capturing actual default cases, which is critical for risk management. Furthermore, the improvement in ROC-AUC to 0.759 demonstrates that the tuned model has better overall separability between classes. Therefore, the optimization achieved its goal of enhancing the model's sensitivity to defaults.

3. Model Comparison: Decision Tree vs. XGBoost

A. Experimental Setup

Both models were trained and evaluated on the same preprocessed dataset derived from the UCI Credit Card Default Dataset.

The dataset was split into 70% training and 30% testing using stratified sampling to preserve the original class distribution (default rate ≈ 22.1%).

Identical preprocessing steps were applied to both models, including:

- Outlier capping using percentile-based thresholds
- Cleaning of categorical variables (EDUCATION, MARRIAGE)
- Feature engineering with the combined Gender–Marriage variable
- Handling class imbalance through class weighting (Decision Tree) and scale_pos_weight (XGBoost)

This ensured a fair and consistent comparison.

B. Performance Comparison

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

C. Confusion Matrix Analysis

Decision Tree

```
[[5885 1069]
 [ 890 1087]]
```

XGBoost

```
[[5971  983]
 [ 869 1108]]
```

Compared to the Decision Tree, XGBoost:

Correctly identified more default cases (1108 vs. 1087)

Reduced false positives (983 vs. 1069)

Slightly reduced false negatives, improving default detection

D. Discussion of Results

i. Improved Overall Performance

XGBoost achieved higher accuracy (0.793 vs. 0.781) and a notably higher ROC-AUC (0.777 vs. 0.748), indicating stronger discriminative power between defaulters and non-defaulters.

ii. Better Handling of Class Imbalance

With the use of the `scale_pos_weight` parameter, XGBoost demonstrated superior performance in identifying minority-class samples (defaulters). This is reflected in higher recall (0.560 vs. 0.550) and F1-score (0.545 vs. 0.526), which are critical metrics in credit risk prediction.

iii. Reduction of Misclassification Risk

In credit risk management, failing to identify a defaulter can lead to financial losses. XGBoost reduced false negatives compared to the Decision Tree, thereby improving risk control effectiveness.

iiii. Model Complexity vs. Interpretability

While Decision Trees offer high interpretability, they are prone to overfitting and limited in capturing complex non-linear relationships. XGBoost, as an ensemble boosting method, combines multiple weak learners and better captures interactions among repayment behavior, credit utilization, and demographic features, leading to improved predictive performance.

E. Conclusion on XGBoost Model Contribution

The introduction of XGBoost as an additional classification model clearly improved model performance compared to the baseline Decision Tree.

The gains in F1-score and ROC-AUC demonstrate that XGBoost provides a more balanced and reliable prediction of credit default risk, particularly in the presence of imbalanced data.

Therefore, this contribution successfully enhances the original modeling approach and offers a more robust solution for real-world credit risk assessment scenarios.

4. Model Comparison: Decision Tree vs. lightGBM

A. Experimental Setup

To ensure the consistent setup among experiments, the same dataset, training test split, and the identical preprocessing are used. This ensured a fair and consistent comparison.

Hyperparameter tuning for LightGBM was performed using GridSearchCV with 10-fold cross-validation over 324 parameter combinations. The best parameters found were:

- learning_rate=0.01
- max_depth=7
- n_estimators=200
- num_leaves=31
- min_child_samples=20.

B. Performance Comparison

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%

C. Confusion Matrix Analysis

Decision Tree

[[5885 1069]
[890 1087]]

XGBoost

[[5896 1058]
[827 1150]]

Compared to the Decision Tree, LightBGM:

Correctly identified more default cases (1150 vs. 1087)
Reduced false positives (1058 vs. 1069)
Significantly reduced false negatives (827 vs. 890), improving default detection by 7.1%

D. Discussion of Results

i. Improved Overall Performance

LightGBM achieved higher accuracy (0.789 vs. 0.781) and a notably higher ROC-AUC (0.779 vs. 0.748), representing a 4.1% improvement in discriminative power between defaulters and non-defaulters.

ii. Better Handling of Class Imbalance

With the use of the `scale_pos_weight` parameter, LightGBM demonstrated superior performance in identifying minority-class samples (defaulters). This is reflected in higher recall (0.582 vs. 0.550, a 5.8% improvement) and F1-score (0.550 vs. 0.526, a 4.5% improvement), which are critical metrics in credit risk prediction.

iii. Reduction of Misclassification Risk

In credit risk management, failing to identify a defaulter can lead to financial losses. LightGBM reduced false negatives from 890 to 827 compared to the Decision Tree, thereby improving risk control effectiveness by correctly identifying 63 additional defaulters.

iiii. Feature Importance Insights

LightGBM's feature importance analysis revealed that PAY_0 (repayment status in September) is by far the most predictive feature, consistent with the paper's findings that repayment history is the strongest indicator of default risk. Other important features include PAY_AMT2 (payment amount), LIMIT_BAL (credit limit), and BILL_AMT1 (bill statement amount).

E. Conclusion on LightGBM Model Contribution

The introduction of LightGBM as an additional classification model clearly improved model performance compared to the baseline Decision Tree.

The gains in F1-score (+4.5%) and ROC-AUC (+4.1%) demonstrate that LightGBM provides a more balanced and reliable prediction of credit default risk, particularly in the presence of imbalanced data.

LightGBM also offers computational efficiency advantages as a gradient boosting framework, making it suitable for larger-scale credit risk assessment applications.

Therefore, this contribution successfully enhances the original modeling approach and offers a robust solution for real-world credit risk assessment scenarios.

Significant Improvements

1. **Validation on a Similar Dataset**
   We applied our methodology to a different dataset (HMEQ mortgage data) to see if it works in other situations. By using similar steps, we proved that the approach in the paper is effective not just for the original data, but for other credit risk problems as well.

2. **Improved Risk Detection (Hyperparameter Tuning)**
   We adjusted the Decision Tree settings to focus on "Recall" (the ability to find defaults). This successfully improved the model's sensitivity. As a result, the model became much better at catching high-risk customers compared to the original version. This is very important for reducing financial losses.

3. **Better Performance with Ensemble Models**
   We introduced advanced models like XGBoost and LightGBM. These ensemble methods showed higher overall performance and stability compared to the single Decision Tree. They were able to predict defaults more accurately, proving that modern algorithms are more powerful for this task.

Challenges

1. **Balancing Precision and Recall**
   A big challenge was the trade-off between Accuracy and Recall. When we tried to catch more defaults (increase Recall), the model also raised more false alarms (lower Precision). Finding the right balance where the model is both useful and accurate was difficult.

2. **Handling Data Differences**
   When we tested the similar dataset (HMEQ), it had different problems, such as many missing values. Deciding how to fix these missing parts without changing the data's meaning was a complex task. It was hard to keep the preprocessing steps consistent across different datasets.

3. **Learning and Implementing Advanced Models**
   We introduced advanced models like XGBoost and LightGBM, which were not covered in our class. Since we had to learn these algorithms from scratch, implementing them correctly was a big challenge. Also, tuning their parameters is much more complex than a simple Decision Tree, requiring us to understand how gradient boosting works to get the best results.

Conclusion

This project successfully reproduced the methodology presented by Gan et al., confirming that the Decision Tree is a valid baseline for understanding credit risk. Beyond simple reproduction, the study was extended by applying the model to a similar mortgage dataset (HMEQ) and introducing advanced algorithms like XGBoost and LightGBM.

Our experiments revealed a clear distinction: while the Decision Tree from the original paper offers superior interpretability, the advanced ensemble models demonstrated significantly higher predictive power and stability. Specifically, the hyperparameter tuning applied to the Decision Tree improved its ability to detect defaults, but the boosting algorithms provided the best overall performance. We conclude that while the paper's methodology is sound for explaining risk factors, modern credit scoring systems should leverage ensemble methods to maximize detection accuracy.

Future Scope

1. Using Newer Data

The dataset we used is from 2005, which is quite old. Economic conditions change over time. In the future, we should test our model with recent credit card data (for example, from 2024 or 2025) to make sure it works for today's customers.

2. Trying Deep Learning

Next time, we should try Neural Networks (Deep Learning). These models might find complex patterns that Decision Trees miss, potentially improving the prediction accuracy even more.

3. Advanced Feature Engineering

The current model relies mainly on the demographic and payment history variables provided in the dataset. Future work could involve creating new, more complex features. For example, calculating the "Credit Utilization Ratio" (Bill Amount divided by Credit Limit) or analyzing the specific trends in payment delays over time could reveal deeper risk patterns that simple variables miss.