Business Case with Machine Learning

Supervised Learning: Classification Algorithms

Artificial Intelligence II: Prof. Marc Torrens

Noor Ahmad Raza

esade
RAMON LLULL UNIVERSITY

Agenda

1 **Overview**
1.1. Project Overview

2 **Data Preparation**
2.1. Data Cleaning
2.2. Feature Engineering
2.3. Feature Selection

3 **Model Creation**
2.1. Initial Random Forest Model Training and Evaluation
2.2 Feature Importance Analysis
2.3 Handling Class Imbalance with SMOTE
2.4 Retrain Random Forest After Balancing

4 **Model Interpretation**
4.1. Model Evaluation
4.2. Confusion Matrix Interpretation
4.3. Cost-Sensitive Threshold Optimization

5 **Model Implementation**

Overview

1.1. Dataset Overview

1.1. Project Overview

Business Objective: To build a predictive model that classifies loan applications as safe or risky and evaluates the financial implications of these predictions to guide investment decisions

Methodology

Data Preparation → Model Creation → Model Interpretation → Model Implementation

Outcome: A cost-optimized loan approval model that supports data-driven investment strategies, balancing risk and return

Data Preparation

2.1. Data Cleaning

2.2 Feature Engineering

2.3 Feature Selection

2.1. Data Cleaning

Column Filtering: Removing Low-Value & Leaky Features → Missing Values Analysis → Handling Missing Values → Imputing Remaining Numerical Features

2.2. Feature Engineering

Date Processing: We transformed the loan issuance date into meaningful temporal features to capture macro trends and seasonality → Define Target Variable and Separate Open vs Closed Loans: Filtered the dataset into closed loans (for model training) and open loans (for later prediction)

2.3. Feature Selection

Identifying Categorical Columns → Categorical Encoding for Model Training → Feature Selection → Correlation Analysis of Numerical Features → Final Numerical Feature Selection

Model Creation

3.1. Initial Random Forest Model Training and Evaluation

3.2. Feature Importance Analysis

3.3. Handling Class Imbalance with SMOTE

3.4. Retrain Random Forest After Balancing

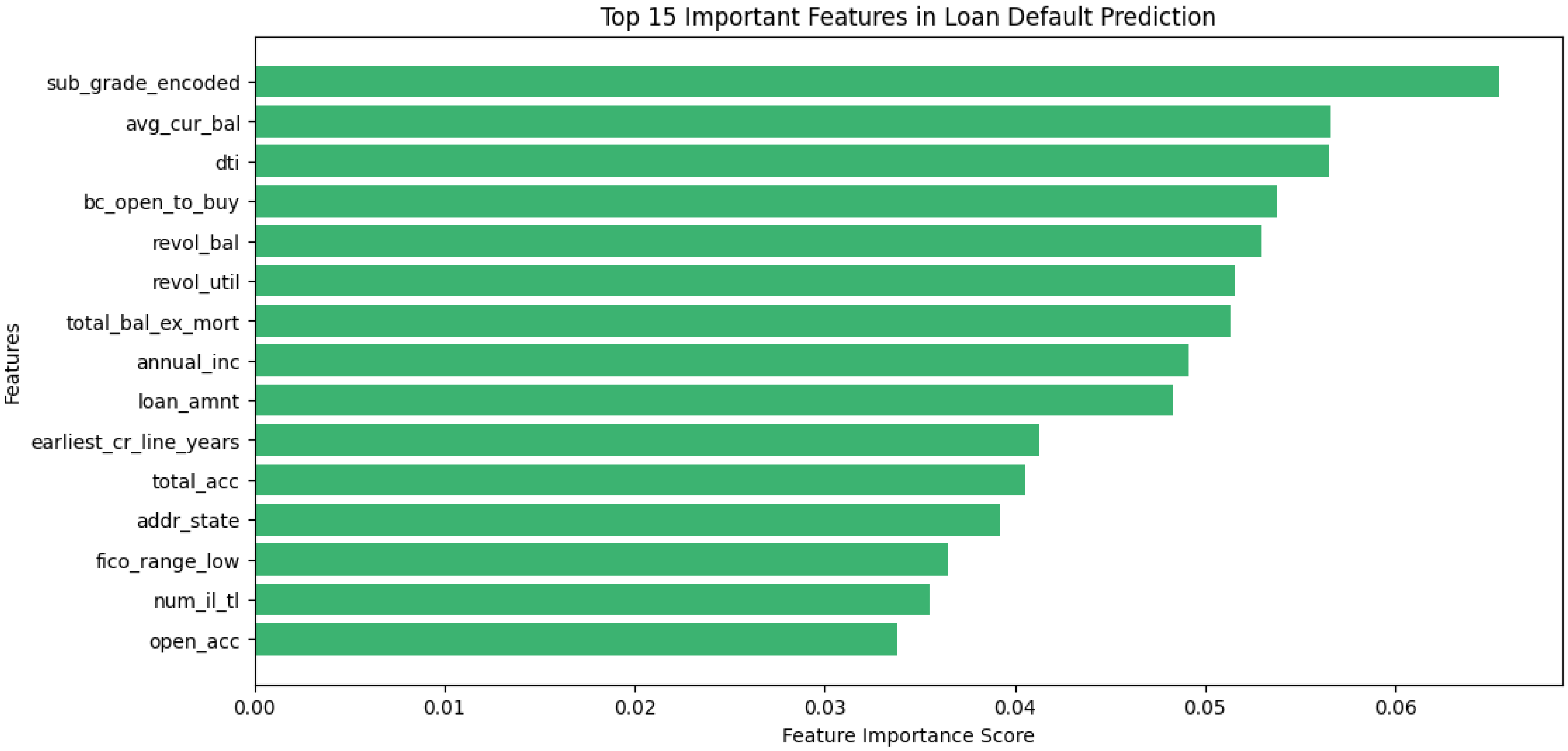
3.1. Initial Random Forest Model Training and Evaluation

We trained a **Random Forest Classifier** using the selected and pre-processed features

```
Random Forest Classifier Performance:
Accuracy: 0.7913352447897398

Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.09      0.16     27575
           1       0.80      0.98      0.88    102169

    accuracy                           0.79    129744
   macro avg       0.68      0.54      0.52    129744
weighted avg       0.75      0.79      0.73    129744
```

3.2. Feature Importance Analysis

After training the Random Forest model, we analysed which features contributed the most to the prediction.



3.3. Handling Class Imbalance with SMOTE

a) We checked to see if there is any Class Imbalance

```
Class Distribution in y_train:
loan_status_binary
1    0.787464
0    0.212536
Name: proportion, dtype: float64
```

b) To address class imbalance in the training set, we apply SMOTE

```
Balanced Class Distribution After SMOTE:
loan_status_binary
1    0.5
0    0.5
Name: proportion, dtype: float64
```

3.4. Retrain Random Forest After Balancing

We retrain the Random Forest model on the SMOTE-balanced training dataset

```
Random Forest Results After SMOTE Balancing:
Accuracy: 0.7867184609692934

Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.17      0.25     27575
           1       0.81      0.95      0.88    102169

    accuracy                           0.79    129744
   macro avg       0.65      0.56      0.56    129744
weighted avg       0.74      0.79      0.74    129744
```

Model Interpretation

4.1. Model Evaluation

4.2. Confusion Matrix Interpretation

4.3. Cost-Sensitive Threshold Optimization

4.1. Model Evaluation – Classification Report

Random Forest Classification Report

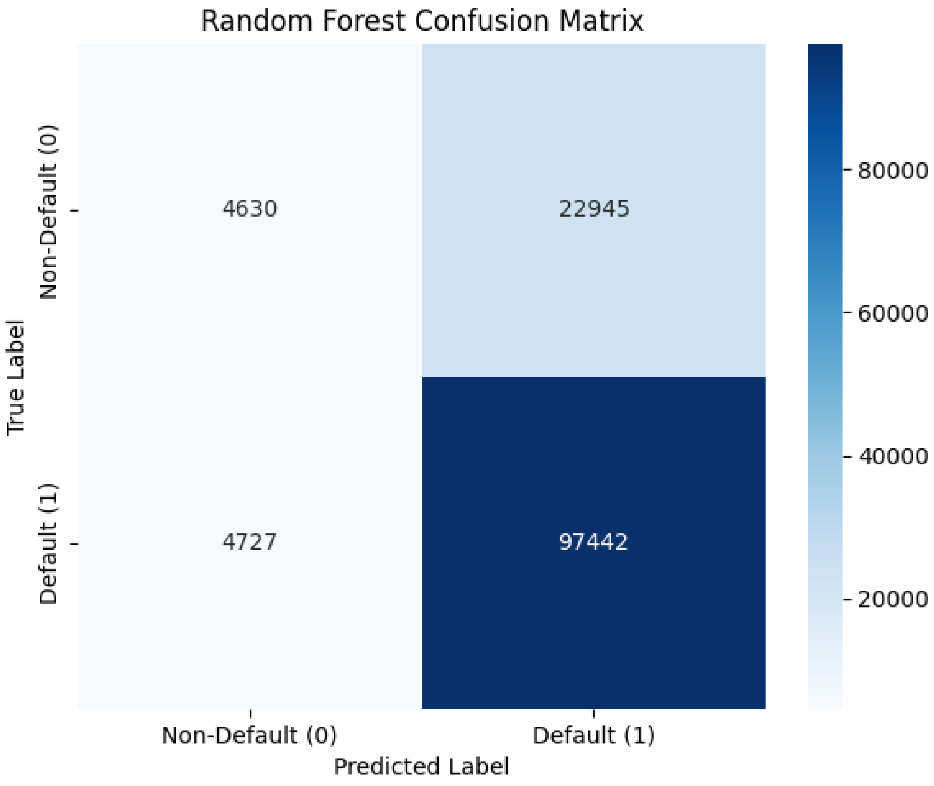
```
Random Forest Results After SMOTE Balancing:
Accuracy: 0.7867184609692934

Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.17      0.25     27575
           1       0.81      0.95      0.88    102169

    accuracy                           0.79    129744
   macro avg       0.65      0.56      0.56    129744
weighted avg       0.74      0.79      0.74    129744
```

XGBoost Classification Report

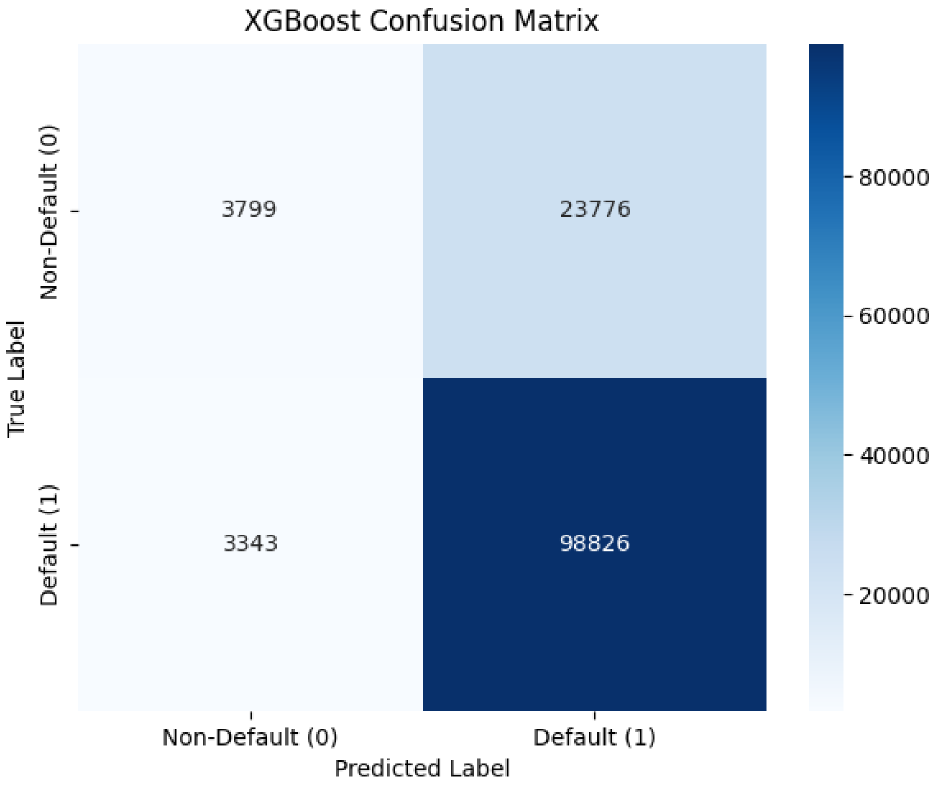
```
XGBoost Results After SMOTE Balancing:
Accuracy: 0.7909807004562831

Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.14      0.22     27575
           1       0.81      0.97      0.88    102169

    accuracy                           0.79    129744
   macro avg       0.67      0.55      0.55    129744
weighted avg       0.75      0.79      0.74    129744
```

4.1. Model Evaluation – Confusion Matrix

Random Forest Confusion Matrix



XGBoost Confusion Matrix



4.2. Confusion Matrix Interpretation

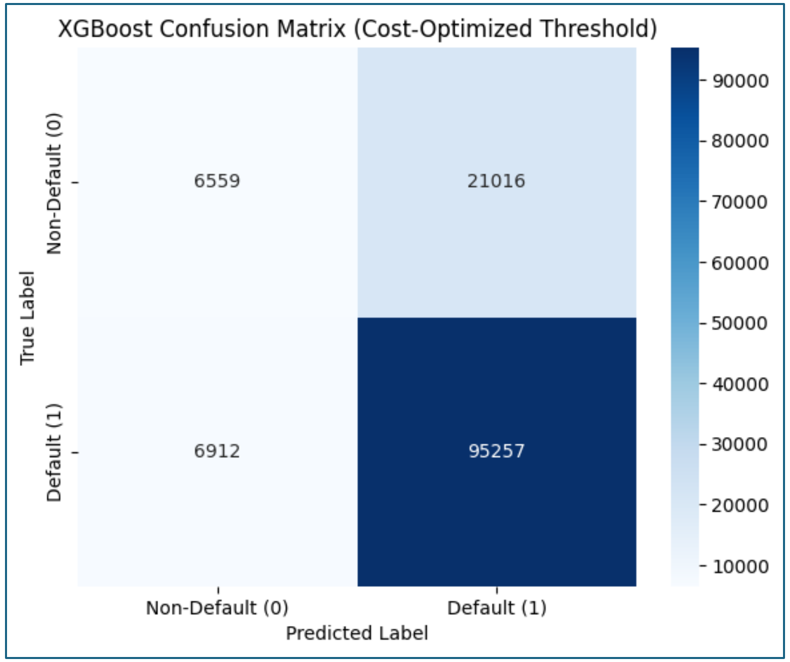
```
Business Impact Analysis — Random Forest
True Positives (TP): 97442 → Correctly identified defaulters
False Positives (FP): 22945 → Lost profitable customers
False Negatives (FN): 4727 → Approved loans that defaulted
True Negatives (TN): 4630 → Correctly identified safe borrowers

Business Impact Analysis — XGBoost
True Positives (TP): 98826 → Correctly identified defaulters
False Positives (FP): 23776 → Lost profitable customers
False Negatives (FN): 3343 → Approved loans that defaulted
True Negatives (TN): 3799 → Correctly identified safe borrowers
```

4.2. Cost-Sensitive Threshold Optimization

To balance the cost of false positives (FP) and false negatives (FN), we explore different decision thresholds.
The goal is to minimize the total business cost:

- FP Cost: Losing a profitable customer (missed revenue)
- FN Cost: Approving a loan that defaults

Process:
Predicted probabilities are extracted using `xgb_model.predict_proba()`.
We evaluate multiple thresholds (ranging from 0.1 to 0.9) to minimize the financial costs associated with prediction errors.
The optimal threshold is selected to minimize total costs (FP + FN)



Model Implementation

5.1. Estimating Business Impact on Open Loans

5.1. Estimating Business Impact on Open Loans

Objective: Use the trained XGBoost model on open loans to predict default risks and estimate the financial impact

Process Overview

•**Feature Consistency**

→ Applied same preprocessing steps from training data to open loans (label encoding, feature engineering).

→ Ensured matching feature columns between training and open loan datasets.

•**Prediction**

→ Used the trained model to estimate **default probabilities**.

→ Applied **cost-optimized threshold** = 0.557

Metric	Value	Interpretation
False Positives (FP)	21,106	Lost profitable customers
False Negatives (FN)	6,559	Approved loans that ended in default
Total Cost	$380.83M	Combined loss from FP and FN cases

Assumption: Cost per False Positive (FP**)** = $15,000 & Cost per False Negative (FN) = $10,000

Key Takeaway: Applying the model with an optimal threshold gives a realistic estimate of **financial risk** from new loan applicants, guiding better investment decisions

Thank You!