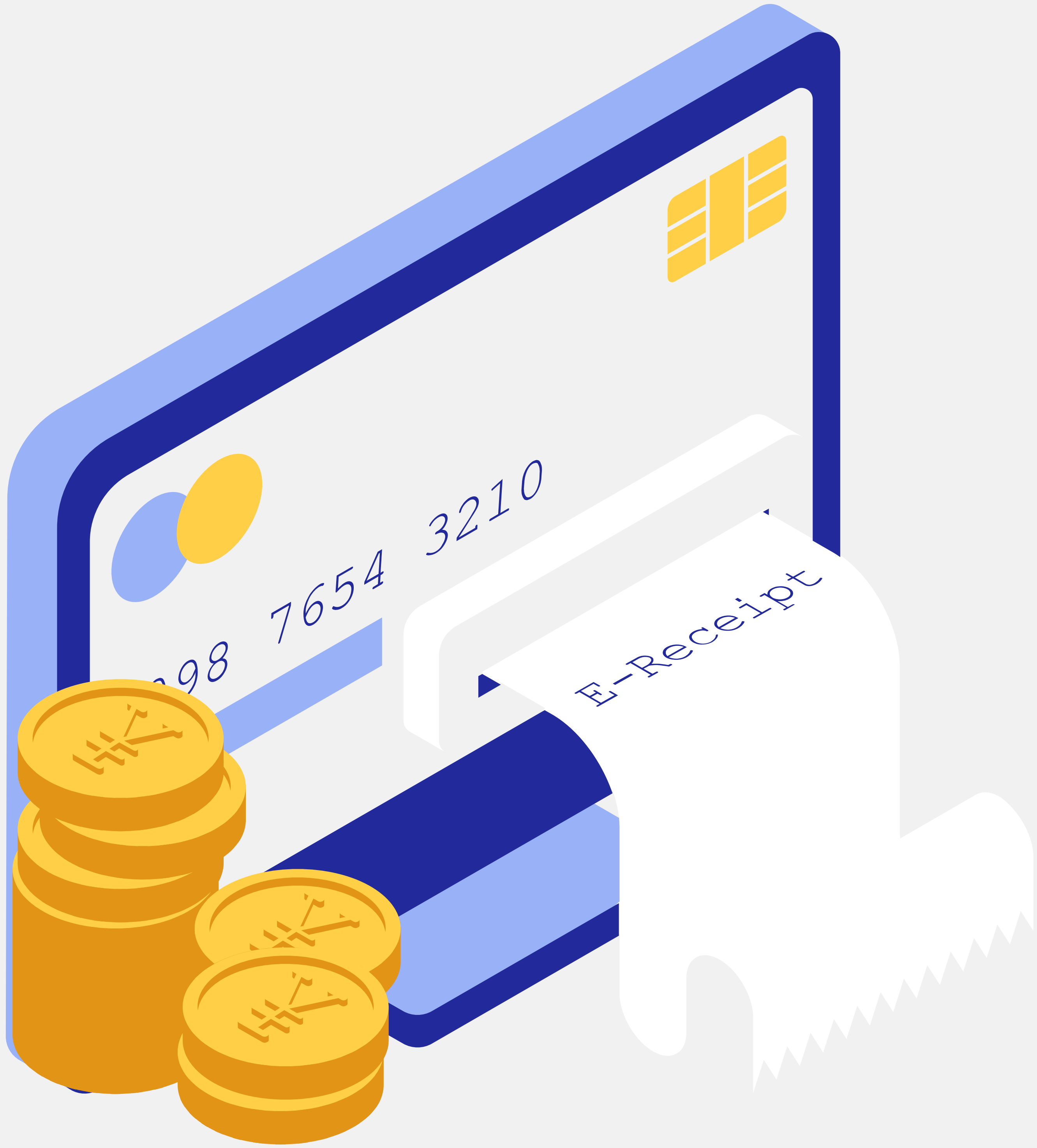
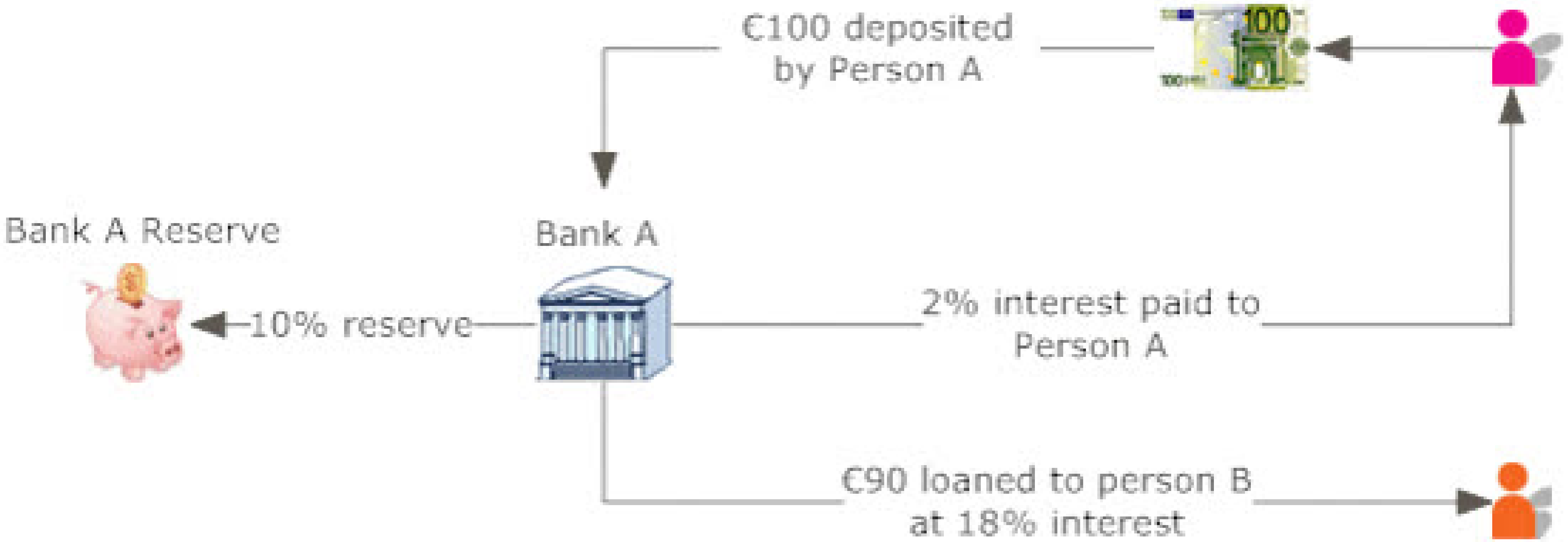


Enhancing Loan Risk Prediction:

A Machine Learning Approach for Minimizing Default Risk

What are "Bank Loans"?

A **bank loan** is a financial arrangement where a bank **lends money** to an individual or a business with the agreement that it will be repaid over time, usually **with interest.**


€100 deposited
by Person A
Bank A Reserve
Bank A
10% reserve
2% interest paid to
Person A
€90 loaned to person B
at 18% interest

Why is this a problem in banking?

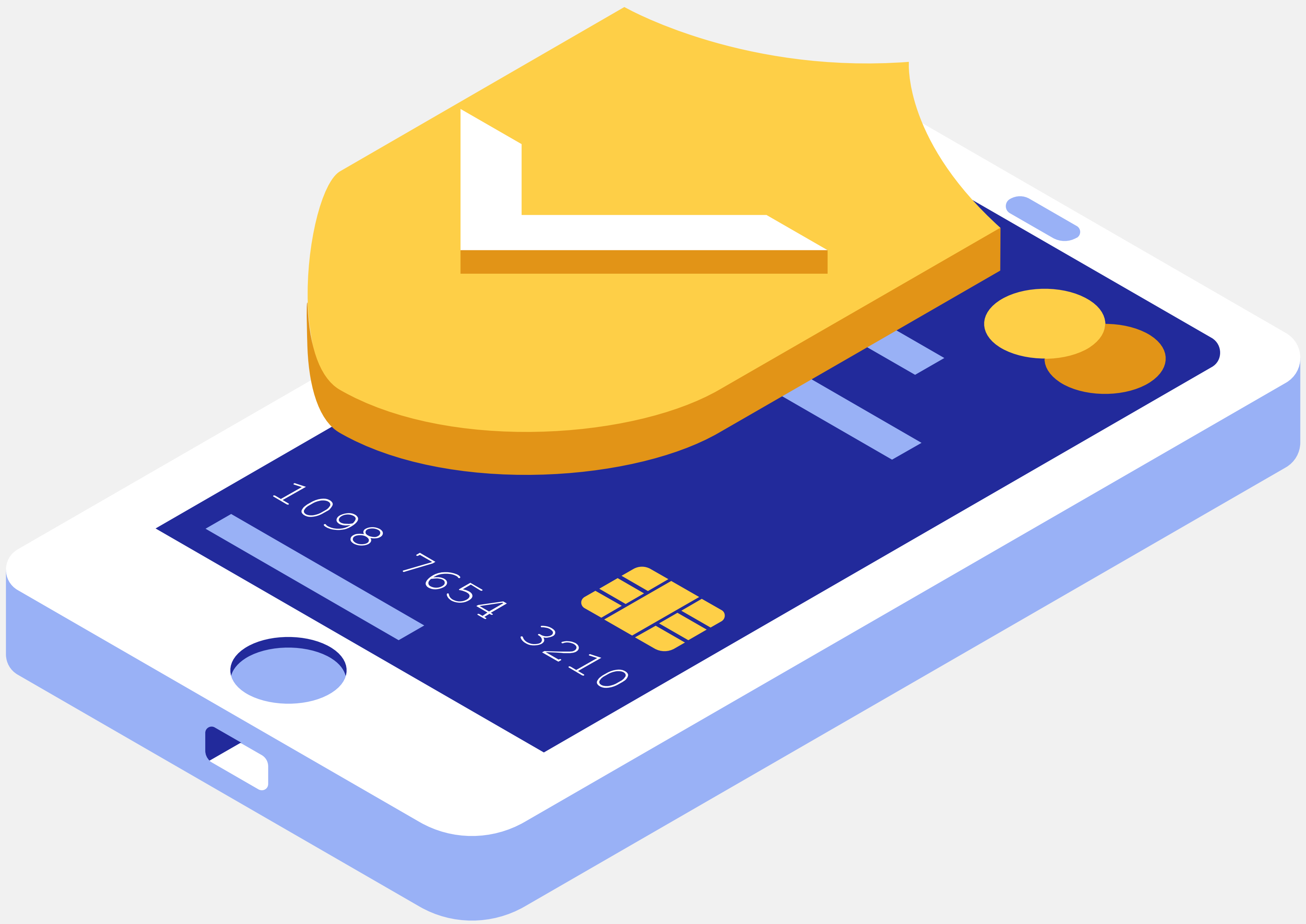
When customers fail to repay their loans, it creates **significant risks** for banks and financial institutions:

- Financial Losses
- Increased Non-Performing Loans (NPLs)
- Higher Interest Rates for Other Customers
- Lower Credit Availability
- Regulatory & Reputation Risk

How Can Machine Learning Help?

By accurately predicting loan risk, **machine learning models** help banks:

- Identify high-risk borrowers **early**
- Minimize **default rates**
- Improve lending decisions
- Optimize interest rates based on risk



Assessment *Without* Machine Learning

- Banks have a limited number of analysts, making it difficult to process large volumes of loan applications efficiently.
- Manual reviews require significant time to analyze each application individually.
- Loan officers may have biases or inconsistent evaluations.

Assessment *With* Machine Learning

- Machine learning can process thousands of applications simultaneously, reducing delays and improving efficiency.
- ML models can analyze patterns and risk factors instantly, leading to faster loan approvals.
- ML algorithms rely on historical data and remove human bias, ensuring fairer decisions.

Goals, Objectives & Key Business Metrics

Goal	Objective	Business Metric
The primary goal is to **optimize the loan approval decision-making process**, making it more accurate, efficient, and data-driven. This includes the ability to assess credit risk profiles effectively and minimize the likelihood of loan defaults.	• Develop a Machine Learning model **as the first filter** to predict whether a customer is likely to default on their loan. • Identify key factors that influence loan default risk. • Provide business recommendations based on insights and findings from data analysis.	**Default Rate:** This metric measures the proportion of customers who default on their loans compared to the total approved loans. It serves as an indicator of the model's ability to mitigate credit risk.

Dataset Overview

Understanding the data we are working with is one of the most important aspects of any analysis.

Column	Description	Type
ID	Customer ID	int64
Income	Income of the user	int64
Age	Age of the user	int64
Experinece	Professional experience of the user in years	int64
Married/Single	Whether married or single	object
House Ownership	Owned or rented or neither	object
Car Ownership	Does the person own a car	object
Profession	Profession	object
City	City of residence	object
State	State of residence	object
Current Job Years	Years of experience in the current job	int64
Current House Years	Number of years in the current residence	int64
Risk Flag	Labels: 1 = Default; 0 = Not Default	int64

This dataset contains **252,000 entries with 13 columns**, representing **customer profiles of an Indian Bank** for bank loan applications.

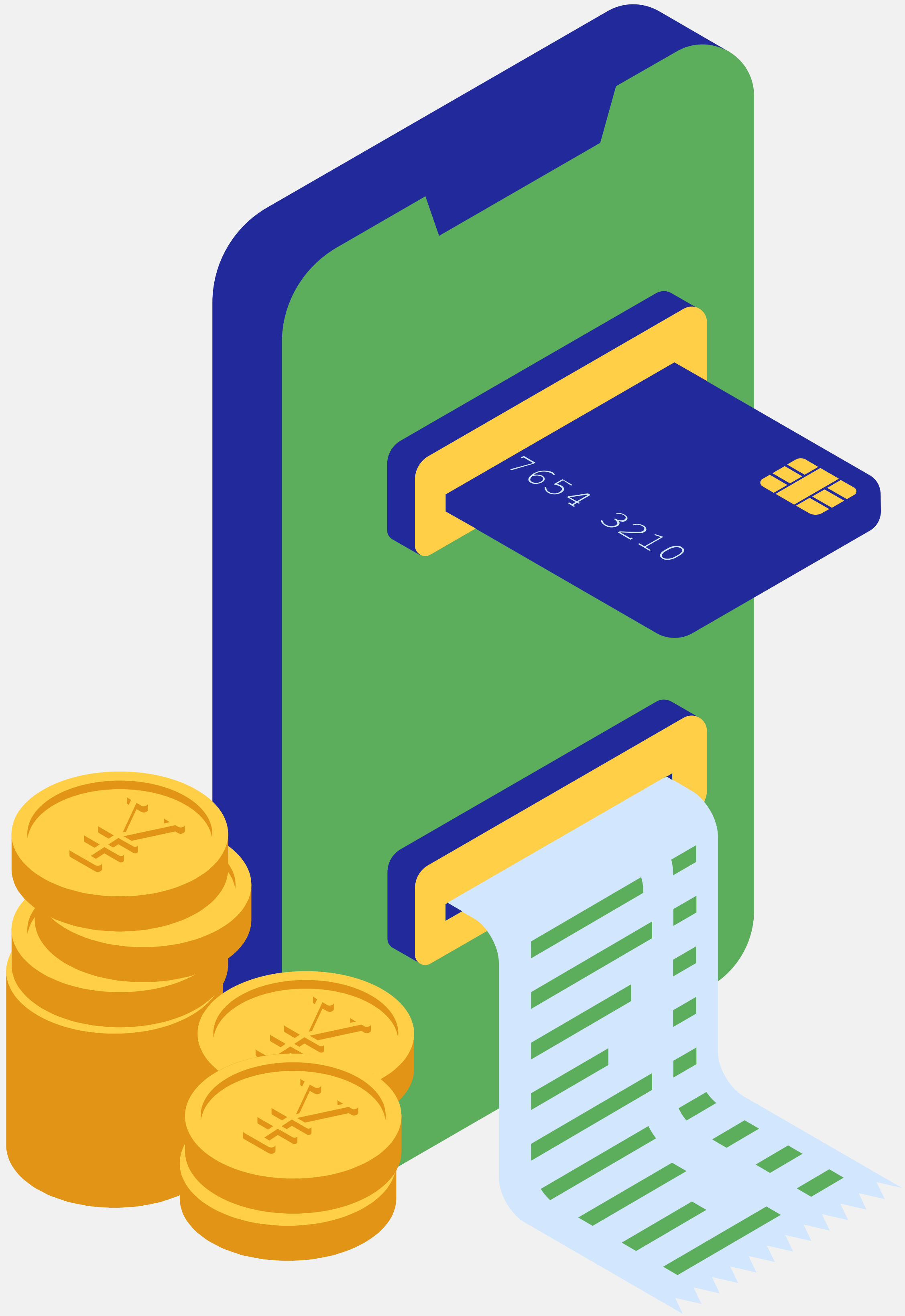
The features include **demographic and financial** attributes such as income, age, years of experience, marital status, house and car ownership, profession, city, state, years at the current job, and years in the current residence.

The target variable, **"Risk_Flag,"** indicates whether a customer is classified as high-risk (1) or low-risk (0) for loan default

Loan Amount?

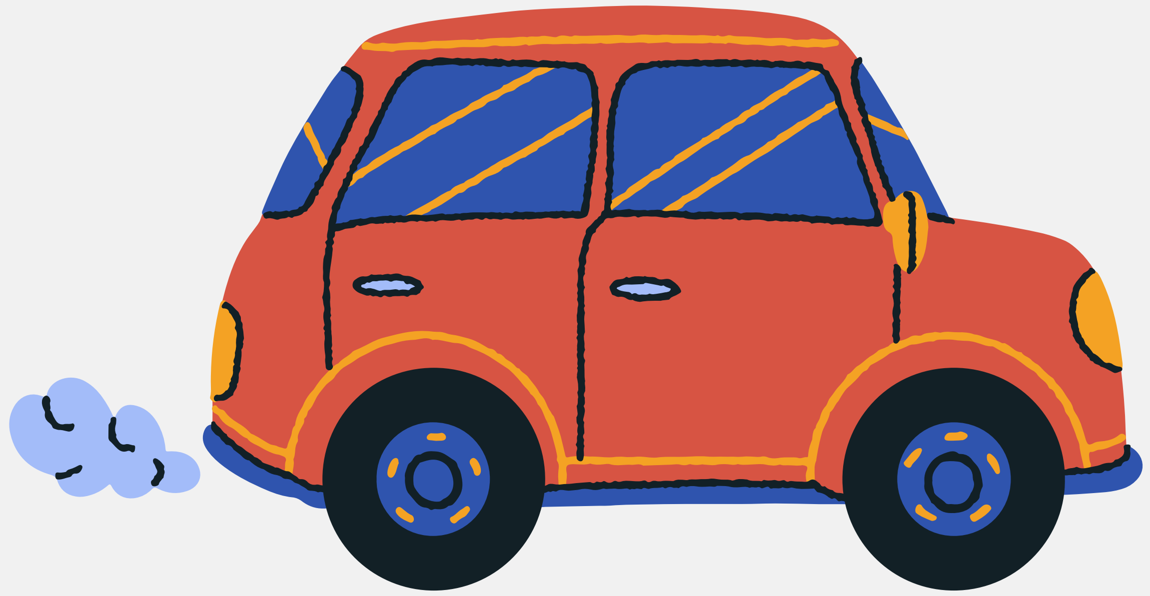
One of the key features for a Loan Prediction should be the Loan Amount

But, we did not have any information regarding the Loan Amount in this dataset



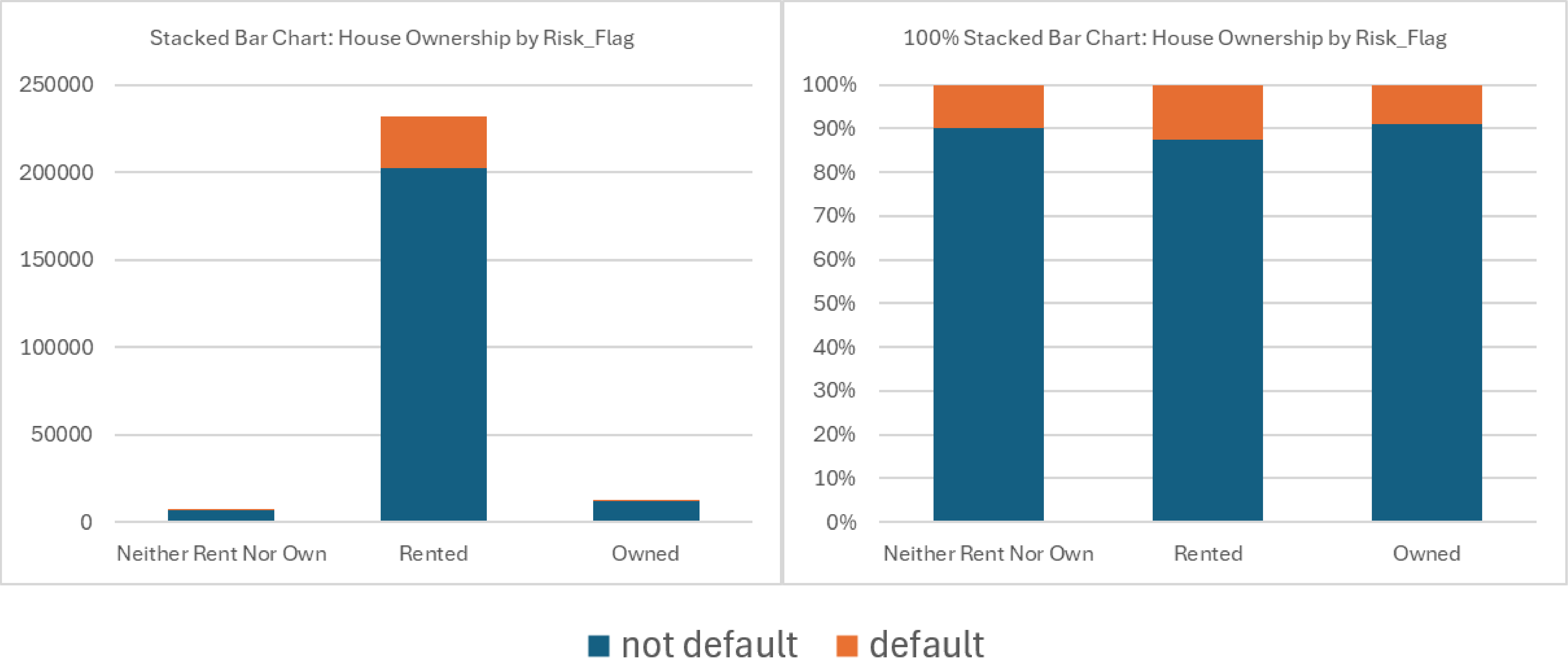
source: *https://docs.iza.org/dp13887.pdf* [1]

Two main reason to loan:

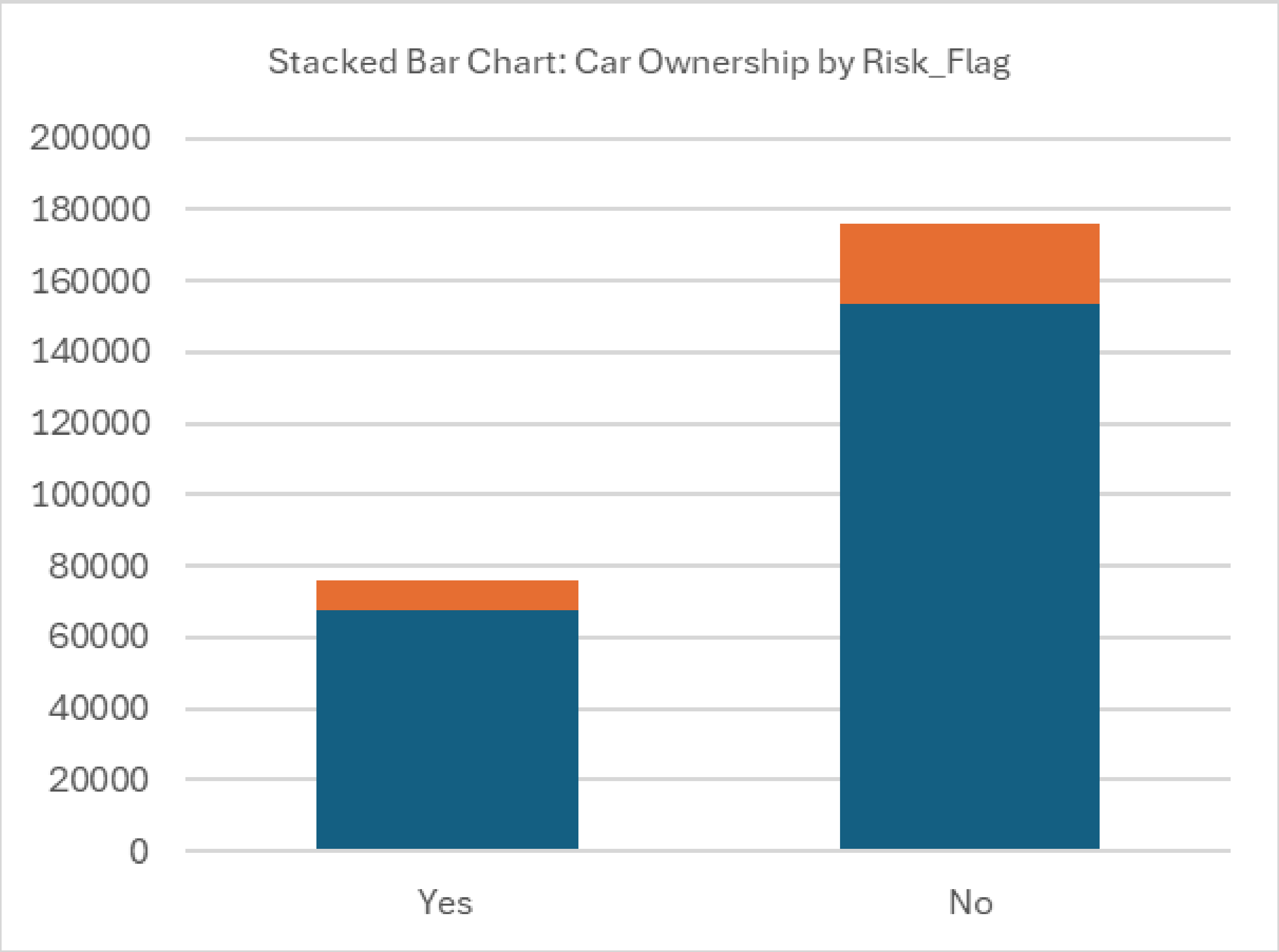


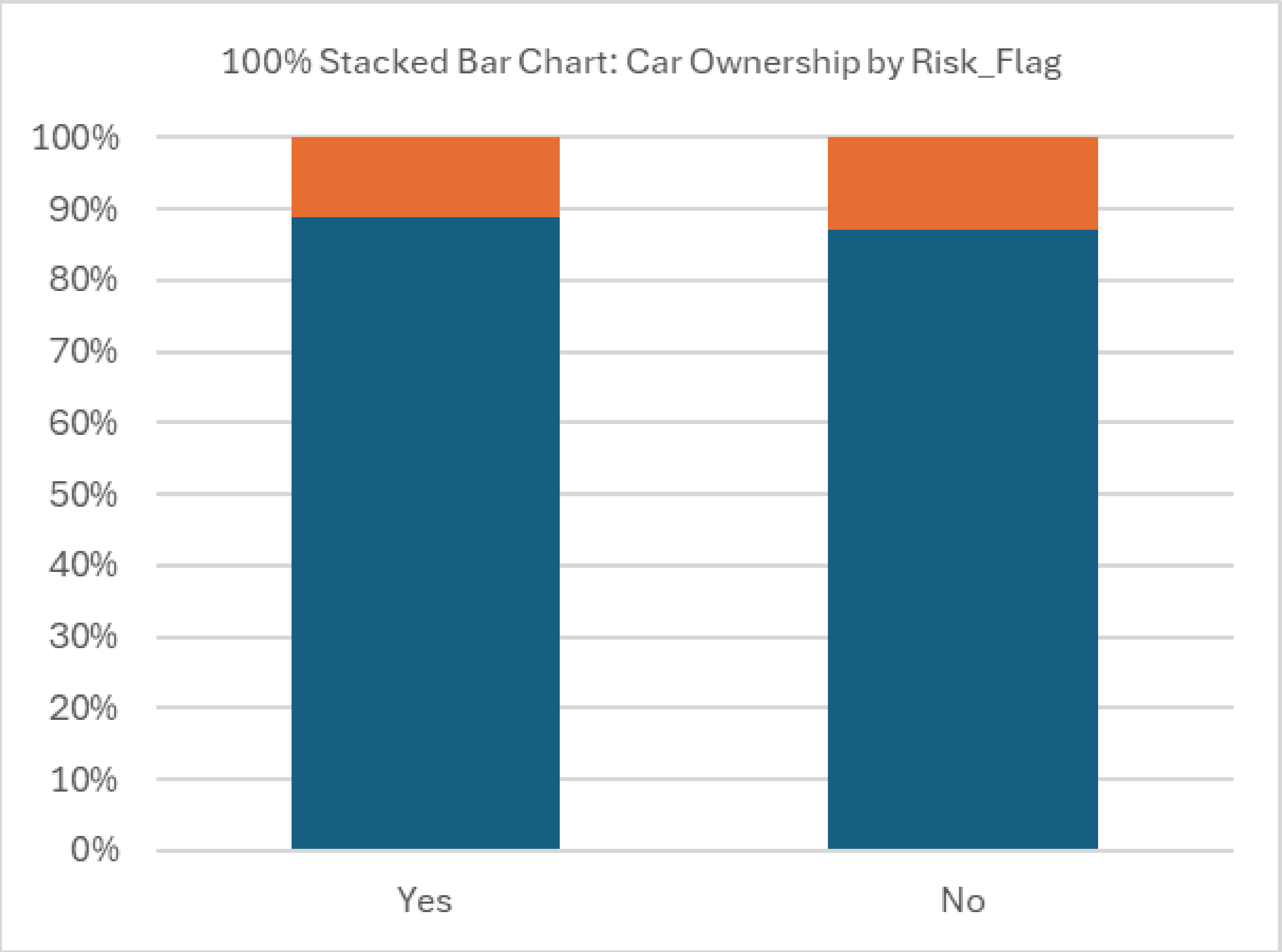
source: https://www.axisbank.com/progress-with-us-articles/money-matters/borrow/top-10-reasons-for-personal-loans

How does 'House ownership' affect default rate?

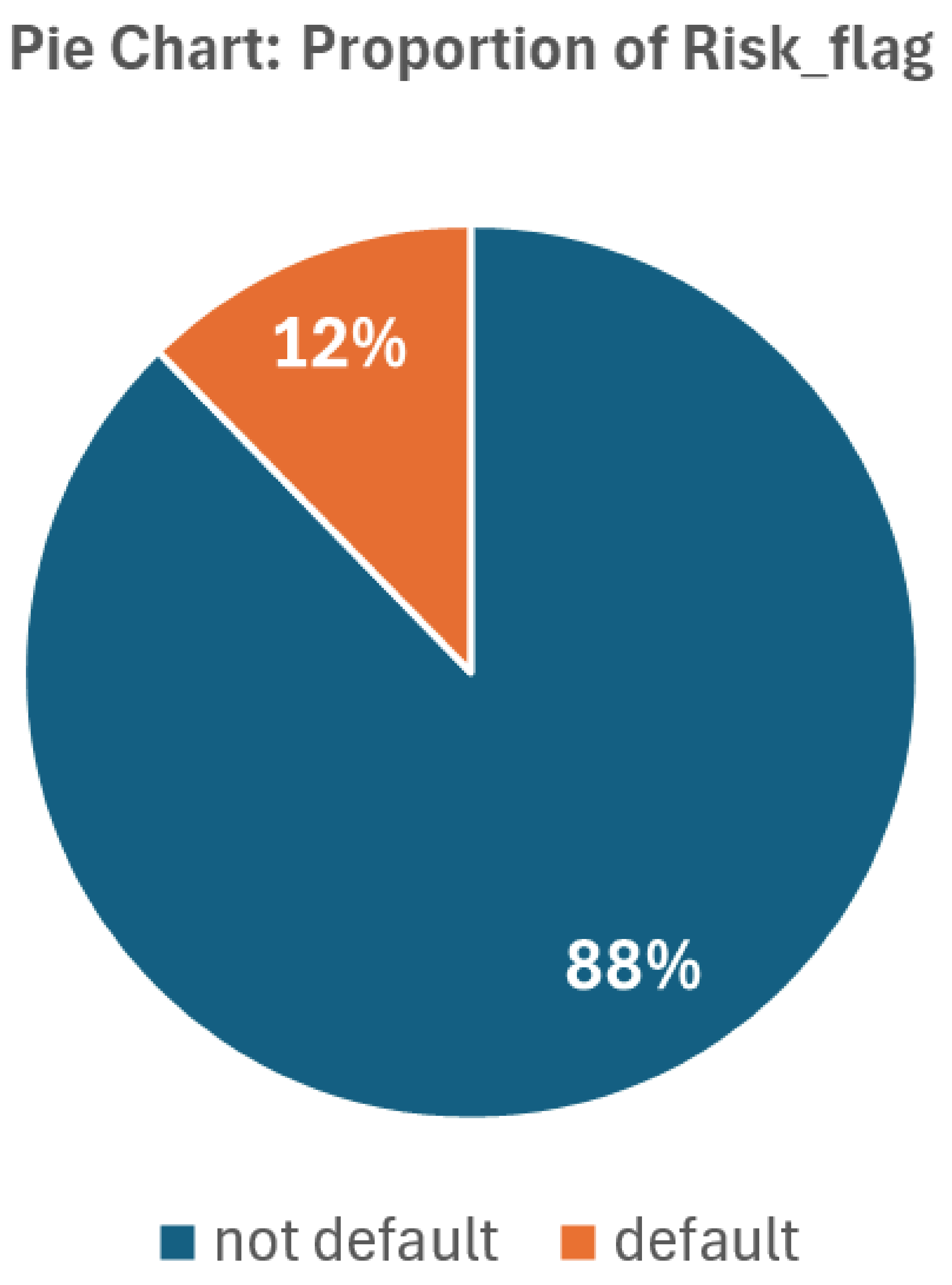


How does ‘Car ownership’ affect default rate?





■ not default ■ default

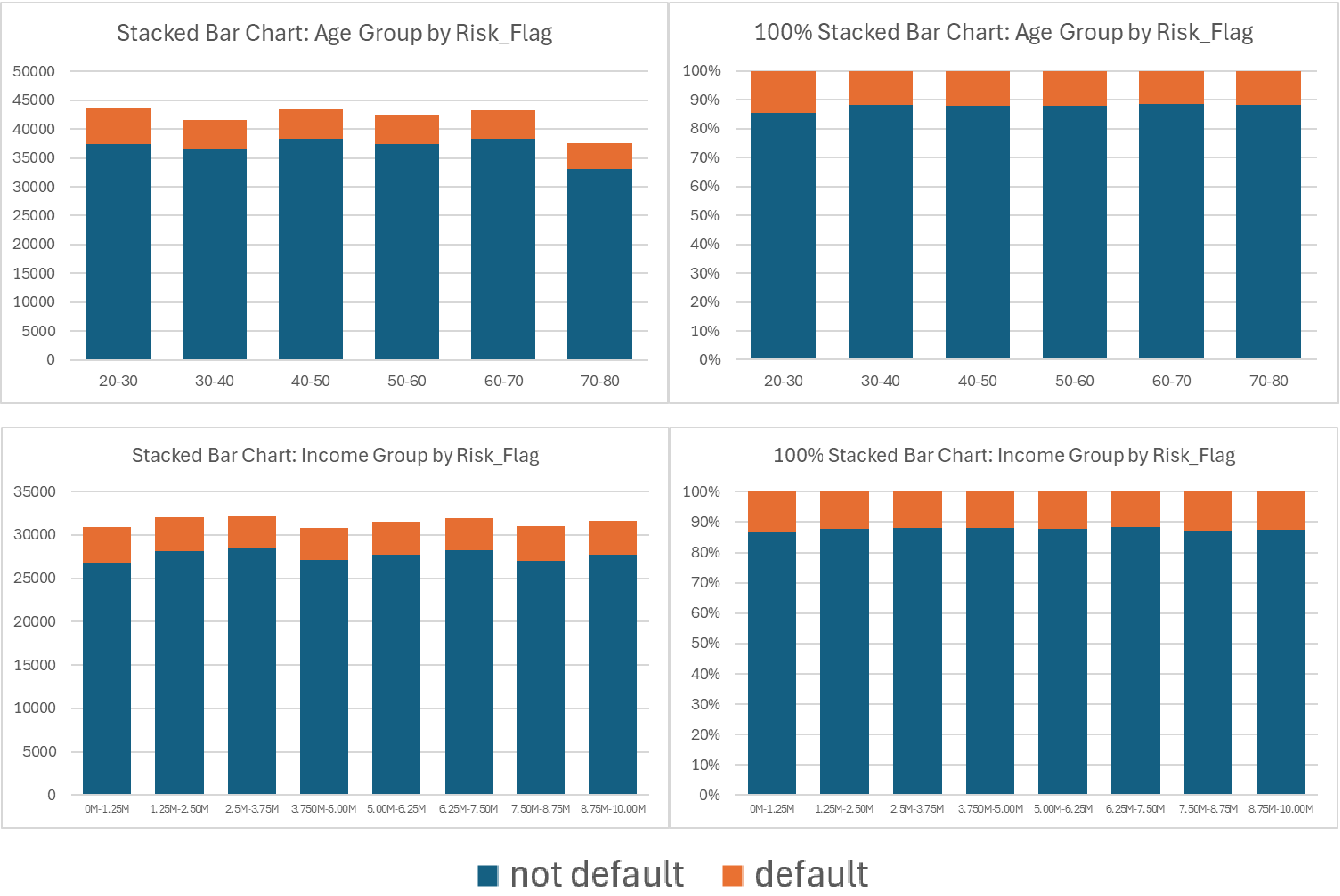


Current Default Rate

12%

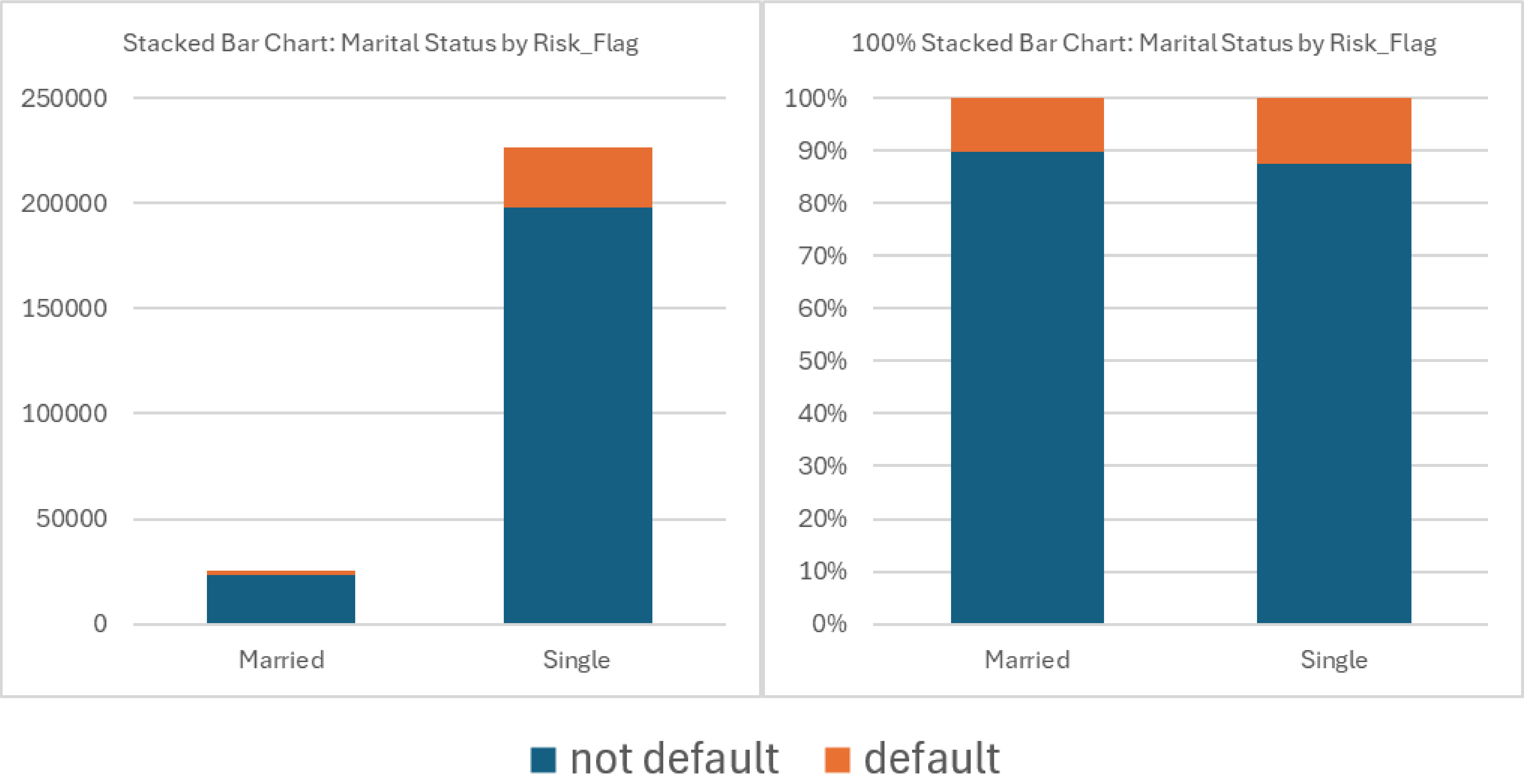
This is very alarming considering ‘normal’ default rate is below 2%.

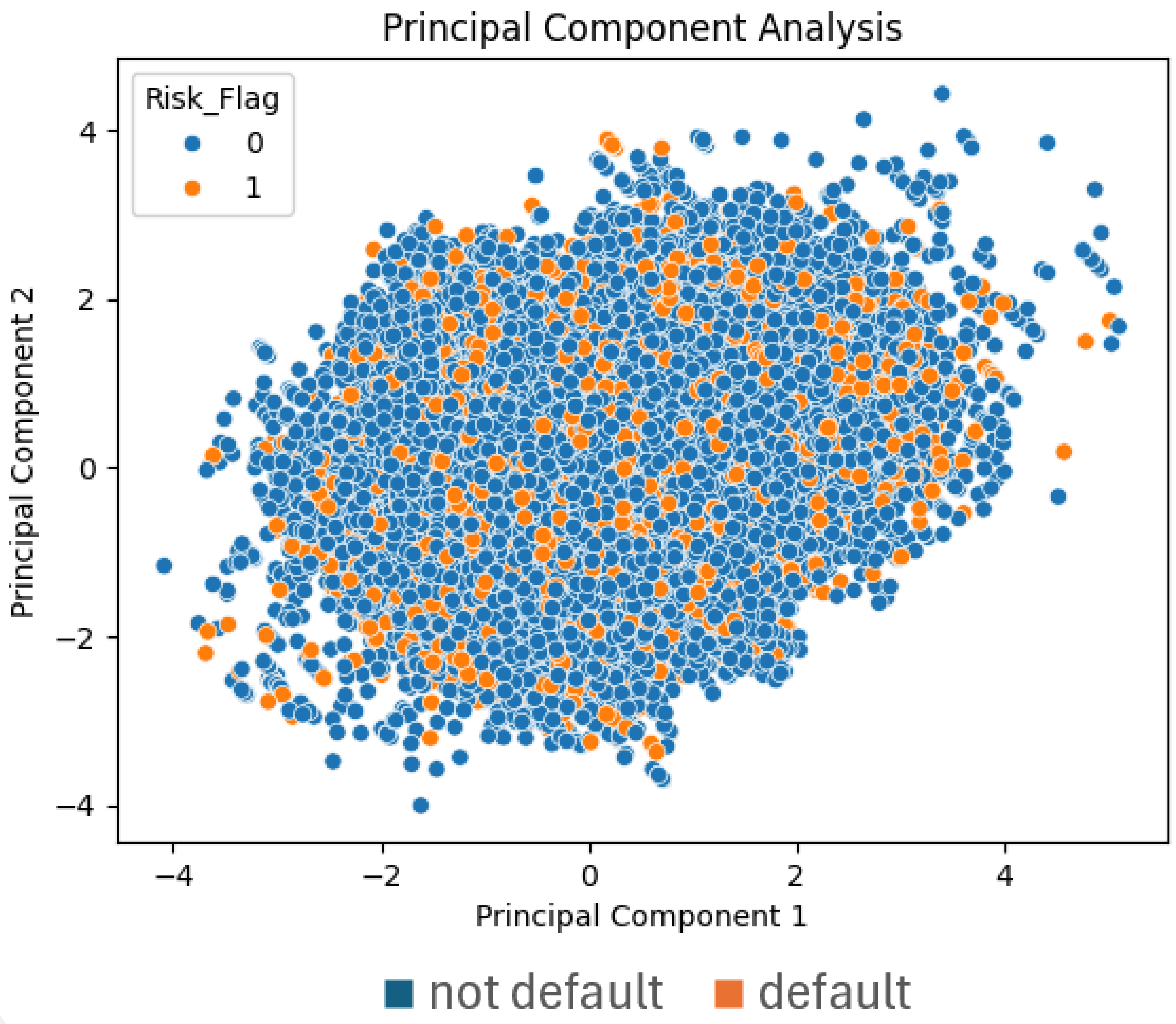
source: https://www.moodys.com/web/en/us/about/insights/data-stories/us-corporate-default-risk-in-2025.html



Stacked Bar Chart: Age Group by Risk_Flag
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
20-30
30-40
40-50
50-60
60-70
70-80
100% Stacked Bar Chart: Age Group by Risk_Flag
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
20-30
30-40
40-50
50-60
60-70
70-80
Stacked Bar Chart: Income Group by Risk_Flag
35000
30000
25000
20000
15000
10000
5000
0
0M-1.25M
1.25M-2.50M
2.5M-3.75M
3.750M-5.00M
5.00M-6.25M
6.25M-7.50M
7.50M-8.75M
8.75M-10.00M
100% Stacked Bar Chart: Income Group by Risk_Flag
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0M-1.25M
1.25M-2.50M
2.5M-3.75M
3.750M-5.00M
5.00M-6.25M
6.25M-7.50M
7.50M-8.75M
8.75M-10.00M
not default
default

How does ‘Marital Status’ affect default rate?





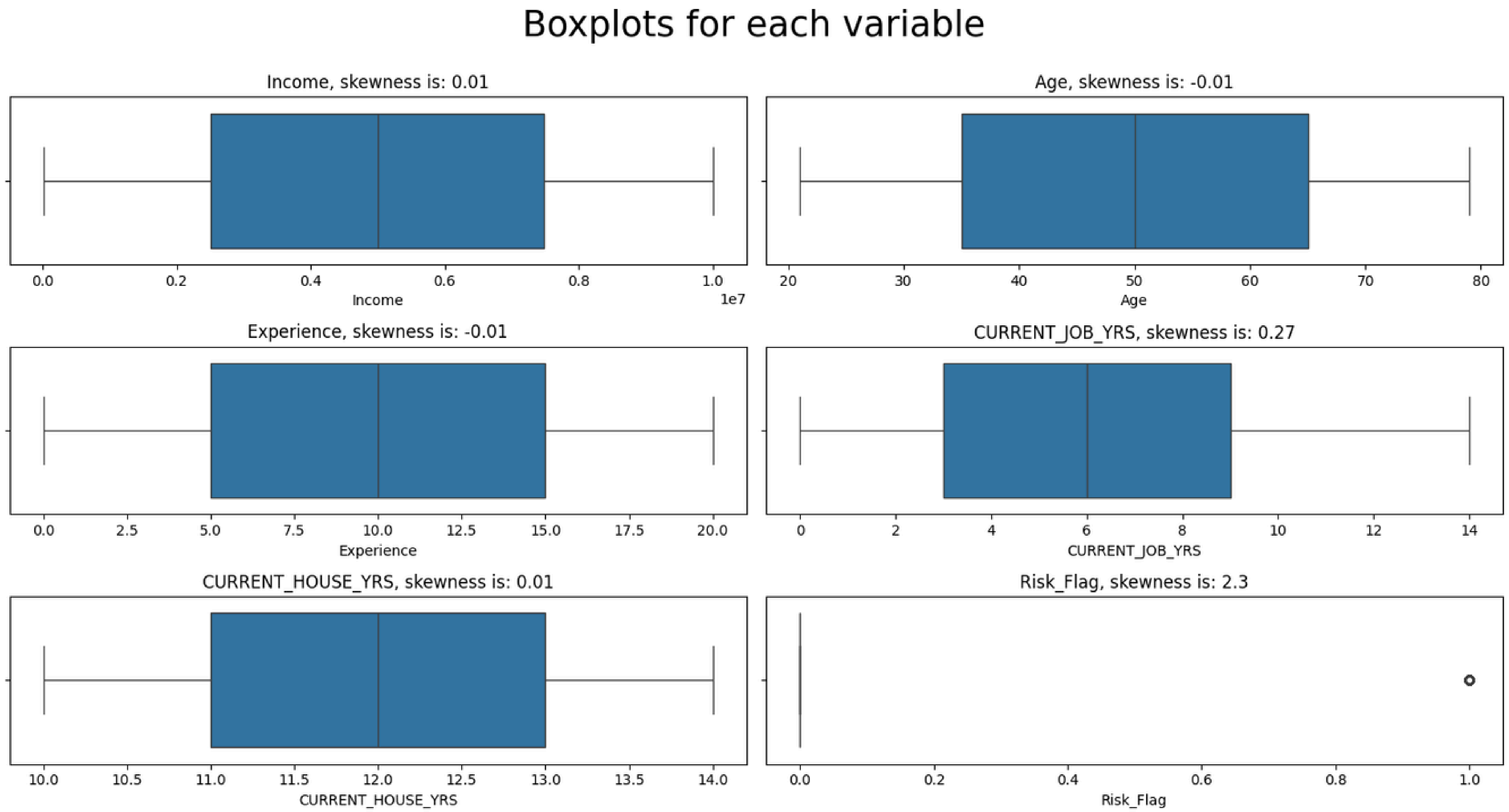
Insights from EDA:

- It is difficult to distinguish high-risk & low-risk borrowers with these two main factors.
- Risk evaluation cannot be based on simple factors alone.
- AI & machine learning approach is needed for better accuracy.

Data Preprocessing & Feature Engineering

Data Preprocessing lays the foundation for transforming raw data into meaningful features that drive accurate and insightful analysis.

Data Cleaning




```
[5] # mengecek data duplikat
    df.duplicated().sum()

0
```

The “Training Data.csv” data is already quite clean from duplicate data and outliers.

No duplicate data, outliers, or missing values were found in the numeric data.

Data Cleaning

```
'Warangal[11][12]' 'Jhansi' 'Bulandshahr' 'Narasaraopet' 'Chinsurah'
'Jehanabad[38]' 'Dhanbad' 'Gudivada' 'Gandhidham' 'Raiganj'
'Kishanganj[35]' 'Varanasi' 'Belgaum' 'Tirupati[21][22]' 'Tumkur'
'Coimbatore' 'Kurnool[18]' 'Gurgaon' 'Muzaffarnagar' 'Aurangabad'
'Bhavnagar' 'Arrah' 'Munger' 'Tirunelveli' 'Mumbai' 'Mango' 'Nashik'
'Kadapa[23]' 'Amritsar' 'Khora,_Ghaziabad' 'Ambala' 'Agra' 'Ratlam'
'Surendranagar_Dudhrej' 'Delhi_city' 'Bhopal' 'Hapur' 'Rohtak' 'Durg'
'Korba' 'Bangalore' 'Shivpuri' 'Thrissur' 'Vijayanagaram' 'Farrukhabad'
'Nangloi_Jat' 'Madanapalle' 'Thoothukudi' 'Nagercoil' 'Gaya'
'Chandigarh_city' 'Jammu[16]' 'Kakinada' 'Dewas' 'Bhalswa_Jahangir_Pur'
'Baranagar' 'Firozabad' 'Phusro' 'Allahabad' 'Guna' 'Thane' 'Etawah'
```

Ada beberapa inconsistency di data lokasi("STATE" dan "CITY") di mana di belakan kode lokasi ada kode angka dengan format "...[99]".

Ini menjadi permasalahan karena ada beberapa nama lokasi yang berulang akibat adanya kode angka tersebut.

Contoh: 'Jammu' dan 'Jammu[16]' merupakan kota yang sama namun karena diformat berbeda, dikategorikan menjadi 2 lokasi yang berbeda.

Therefore, this inconsistency needs to be handled.

```python
# Function to remove trailing numbers in square brackets
def clean_city_name(city):
    return re.sub(r'\[\d+\]', '', city)

# Apply cleaning to all cities
df['CITY'] = np.array([clean_city_name(city) for city in df['CITY']])

# memunculkan kolom CITY
a = df['CITY'].unique()

print(a)
```

This handling is also done for the 'STATE' feature.

Feature Engineering

1. Career Maturity Index

This feature was created to overcome the problem of multicollinearity between the ‘Experience’ and ‘CURRENT_JOB_YRS’ features.

CMI = 0.8 x MinMaxScaler**([’Experience’]) + 0.2 x** MinMaxScaler**([’CURRENT_JOB_YRS’])**

IThis index is created by considering real-world job stability.

Source [1] emphasizes that starting with a stable job significantly increases long-term job stability, income growth, and career development opportunities, while starting with an unstable job leads to steadily declining employment rates and income.

source: *https://docs.iza.org/dp13887.pdf* [1]

Feature Engineering

2. job_groups

There are many unique category values in the ‘Profession’ column and it would be ineffective to use label encoding one by one. To facilitate encoding, ‘Profession’ will be categorized according to the industry of each job in the ‘job_groups’ feature.

```python
# Feature job_encoded (dari 'Profession')
job_groups = {
    'Engineering': ['Mechanical_engineer', 'Civil_engineer', 'Chemical_engineer', 'Design_Engineer',
                    'Computer_hardware_engineer', 'Petroleum_Engineer', 'Industrial_Engineer', 'Engineer'],
    'IT/Software': ['Software_Developer', 'Web_designer', 'Computer_operator', 'Technology_specialist'],
    'Creative': ['Graphic_Designer', 'Technical_writer', 'Fashion_Designer', 'Artist', 'Designer'],
    'Healthcare': ['Physician', 'Dentist', 'Surgeon', 'Psychologist', 'Biomedical_Engineer'],
    'Management': ['Hotel_Manager', 'Consultant', 'Architect', 'Official', 'Chef', 'Analyst'],
    'Legal/Government': ['Politician', 'Magistrate', 'Lawyer', 'Civil_servant', 'Police_officer', 'Firefighter', 'Army_officer'],
    'Financial': ['Financial_Analyst', 'Chartered_Accountant', 'Economist'],
    'Science/Research': ['Scientist', 'Geologist', 'Microbiologist', 'Statistician', 'Technician'],
    'Aviation': ['Flight_attendant', 'Air_traffic_controller', 'Aviator'],
    'Miscellaneous': ['Librarian', 'Secretary', 'Drafter', 'Comedian', 'Surveyor']
}

# Map job positions to groups
df['job_groups'] = df['Profession'].map({job: group for group, jobs in job_groups.items() for job in jobs})
```

Feature Engineering

3. community_type

Similar to the ‘Profession’ feature, the location features (‘CITY’ and ‘STATE’) also have too many unique values, making it ineffective to use label encoding one by one. Therefore, the location features are grouped according to the classification of Indian cities based on the following sources:

Classification of Indian cities

7 languages

Article Talk Tools

From Wikipedia, the free encyclopedia

The *classification of Indian cities* is a ranking system used by the Government of India to allocate House Rent Allowance (HRA) to public servants employed in cities in India. HRA is also used by the Indian Revenue Service (IRS) to provide income tax exemptions. Cities are classified on the basis of

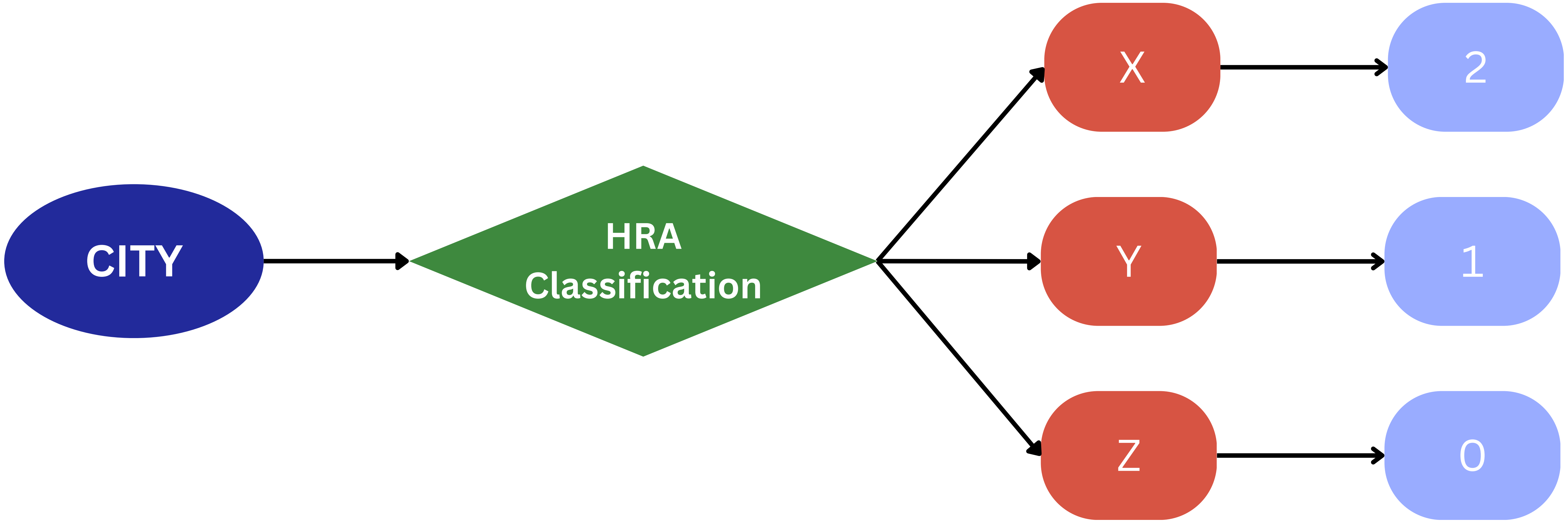
HRA classification	City
X	Ahmedabad, Bengaluru, Chennai, Delhi, Hyderabad, Kolkata, Mumbai, and Pune
Y	Agra, Ajmer, Aligarh, Amravati, Amritsar, Anand, Asansol, Aurangabad, Bareilly, Belagavi, Brahmapur, Bhavnagar, Bhiwandi, Bhopal, Bhubaneswar, Bikaner, Bilaspur, Bokaro Steel City, Burdwan, Bellary, Chandigarh, Coimbatore, Cuttack, Dahod, Dehradun, Dombivli, Dhanbad, Bhilai, Durgapur, Erode, Faridabad, Ghaziabad, Gorakhpur, Guntur, Gurgaon, Guwahati, Gwalior, Hamirpur, Hubballi–Dharwad, Indore, Jabalpur, Jaipur, Jalandhar, Jalgaon, Jammu, Jamshedpur, Jamnagar, Jhansi, Jodhpur, Kalaburagi, Kakinada, Kannur, Kanpur, Karnal, Kochi, Kolhapur, Kollam, Kota, Kozhikode, Kumbakonam, Kurnool, Ludhiana, Lucknow, Madurai, Malappuram, Mathura, Mangaluru, Meerut, Mohali, Moradabad, Mysuru, Nagpur, Nanded, Nadiad, Nashik, Nellore, Noida, Patna, Pimpri-Chinchwad, Puducherry, Purulia, Prayagraj, Raipur, Rajkot, Ranchi, Rourkela, Ratlam,Raichur,Saharanpur, Salem, Sangli, Shimla, Siliguri, Solapur, Srinagar, Surat, Thanjavur, Thiruvananthapuram, Thrissur, Tiruchirappalli, Tirunelveli, Tiruvannamalai, Ujjain, Vijayapura, Vadodara, Varanasi, Vasai-Virar, Vijayawada, Visakhapatnam, Vellore, karimnagar and Warangal.
Z	All other cities and Towns

source: https://en.wikipedia.org/wiki/Classification_of_Indian_cities

Feature Engineering

3. community_type

Encoding with ordinal encoding technique.



source: https://en.wikipedia.org/wiki/Classification_of_Indian_cities

Feature Engineering

4. Encoding Categorical data

For the other categorical data, we do a simple label encoding:

```
# Categorical data yang menggunakan Label Encoding
df['Married/Single_encode'] = df['Married/Single'].map({'married': 1, 'single': 0})
df['House_Ownership_encode'] = df['House_Ownership'].map({'norent_noown':0,'rented': 1, 'owned': 2})
df['Car_Ownership_encode'] = df['Car_Ownership'].map({'yes': 1, 'no': 0})
```

After all the encoding is done, all the data is scaled using MinMaxScaler().

Model Selection & Training

Choosing the right model is critical for achieving accurate predictions, as it determines how well the data's patterns are captured and leveraged for insights.

Model Selection

Logistic Regression

Decision Tree

Random Forest

Extra Tree

XGBoost

LightGBM

CatBoost

X_train and y_train

Models	X_train											y_train
Logistic Regression	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	CMI		community_type		Risk_flag
Decision Tree	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	community_type		Risk_flag
Random Forest	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	community_type		Risk_flag
Extra Tree	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	community_type		Risk_flag
XGBoost	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	community_type		Risk_flag
LightGBM	Income	Age	Married/Single	House_Ownership	Car_Ownership	job_group	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	community_type		Risk_flag
CatBoost	Income	Age	Married/Single	House_Ownership	Car_Ownership	Profession	CURRENT_HOUSE_YRS	Experience	CURRENT_JOB_YRS	CITY	STATE	Risk_flag

encoded =

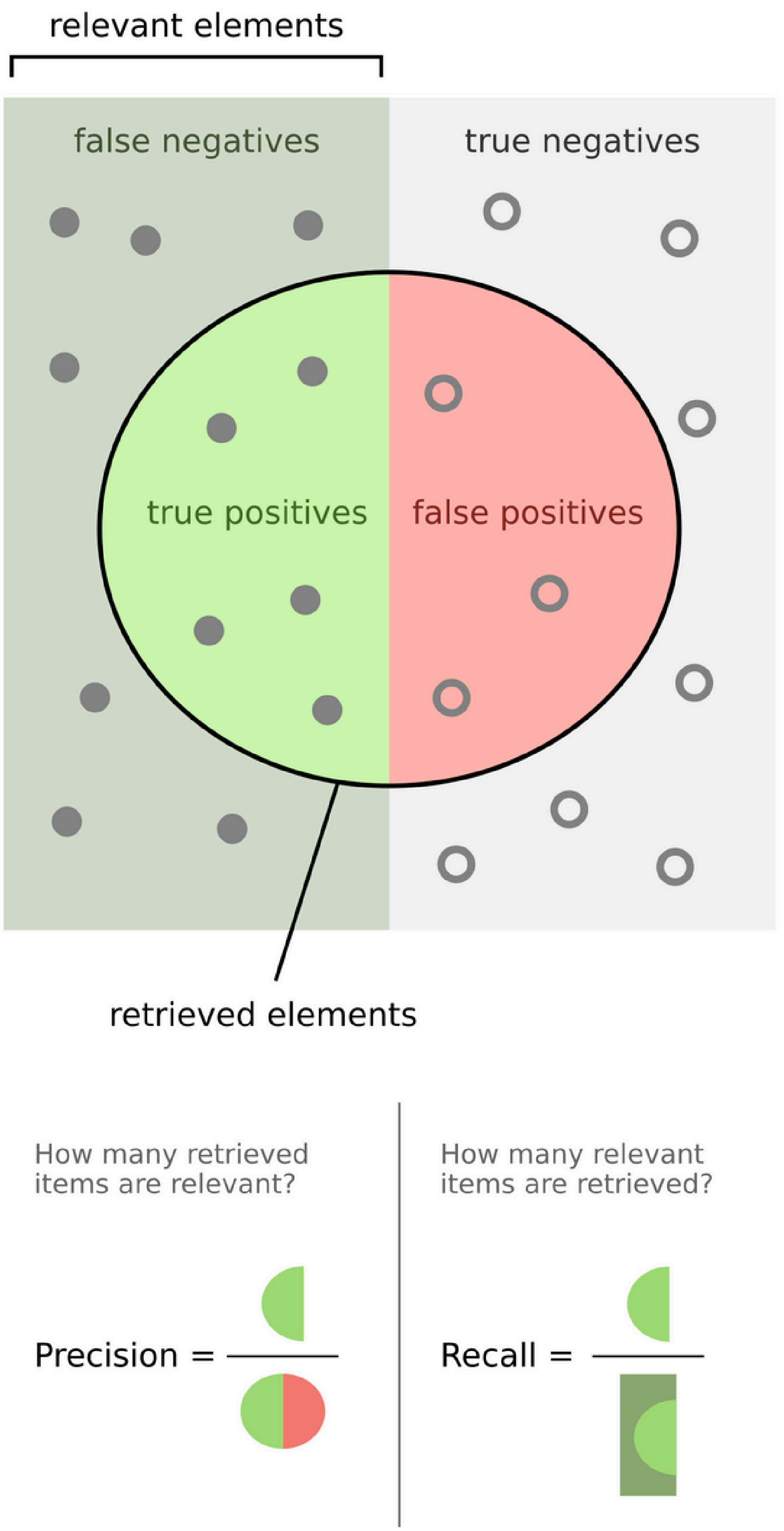
Not encoded =

scaling = [feature]

No scaling = **[feature]**

Why Recall?

In a machine learning project aimed at detecting high-risk loan applicants, recall is a crucial evaluation metric because it measures the model's ability to correctly identify actual high-risk individuals. Missing a high-risk loaner (a false negative) can lead to significant financial losses for the bank if a loan is granted to someone likely to default. By prioritizing recall, the model ensures that most high-risk applicants are flagged, even if that means occasionally misclassifying some low-risk individuals. This trade-off is acceptable in high-stakes scenarios where the cost of overlooking a risky applicant outweighs the cost of mistakenly rejecting a safe one.



Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.52	0.14	0.55	0.22
Decision Tree	0.87	0.48	0.75	0.59
Random Forest	0.90	0.56	0.68	0.61
Extra Tree Classifier	0.89	0.55	0.72	0.62
XGBoost	0.83	0.40	0.81	0.54
LightGBM	0.76	0.30	0.74	0.43
CatBoost	0.83	0.41	0.95	0.57

CatBoost

Since we are looking for a model with the highest Recall metric, the model we chose is CatBoost. Here are the complete results of the evaluation metrics from Catboost:

```
[[53692 12637]
 [  484  8787]]
Accuracy (Test Set): 0.83
Accuracy (Train Set): 0.86
Precision (Test Set): 0.41
Recall (Test Set): 0.95
F1-Score (Test Set): 0.57
roc_auc (test-proba): 0.94
roc_auc (train-proba): 0.95
recall (crossval test): 0.9762242245107171
recall (crossval train): 0.9765536013699183
```

Model Tuning and Optimization

Hyperparameter tuning on CatBoost is done using GridSearch as follows:

```
param_grid = {
    "iterations": [1000, 1500],
    "learning_rate": [0.02, 0.05, 0.1],
    "depth": [4, 6, 8],
    "l2_leaf_reg": [3, 5, 10],
    "bagging_temperature": [0.2, 0.5, 0.8],
    "border_count": [32, 64],
}
```

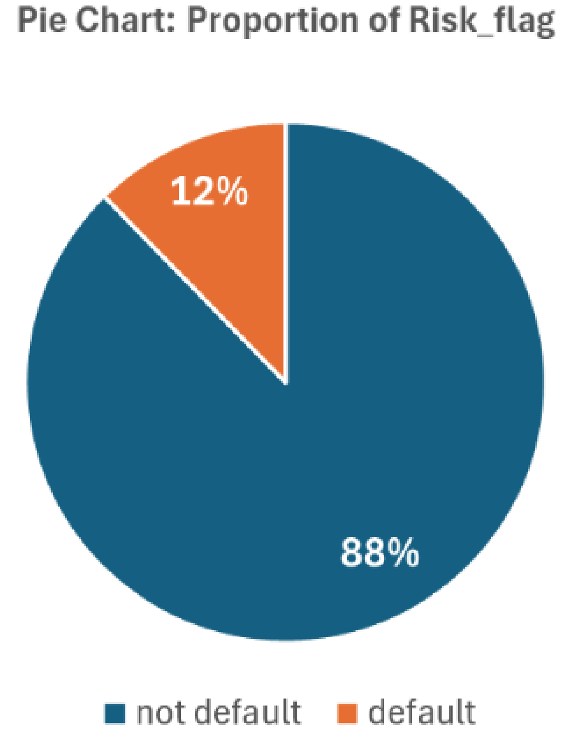
Best Parameters:

- 'bagging_temperature': 0.2,
- 'border_count': 32,
- 'depth': 6,
- 'iterations': 1500,
- 'l2_leaf_reg': 10,
- 'learning_rate': 0.02

Hasil Evaluasi Model akhir:

```
[[54574 11755]
 [  281  8990]]
Accuracy (Test Set): 0.84
Accuracy (Train Set): 0.84
Precision (Test Set): 0.43
Recall (Test Set): 0.97
F1-Score (Test Set): 0.60
roc_auc (test-proba): 0.95
roc_auc (train-proba): 0.95
recall (crossval test): 0.9697084597411653
recall (crossval train): 0.9687703796706983
```

Insights



Current Default Rate

12%

This is very alarming considering 'normal' default rate is below 2%.

12% x (100%-97%) =

0.36%

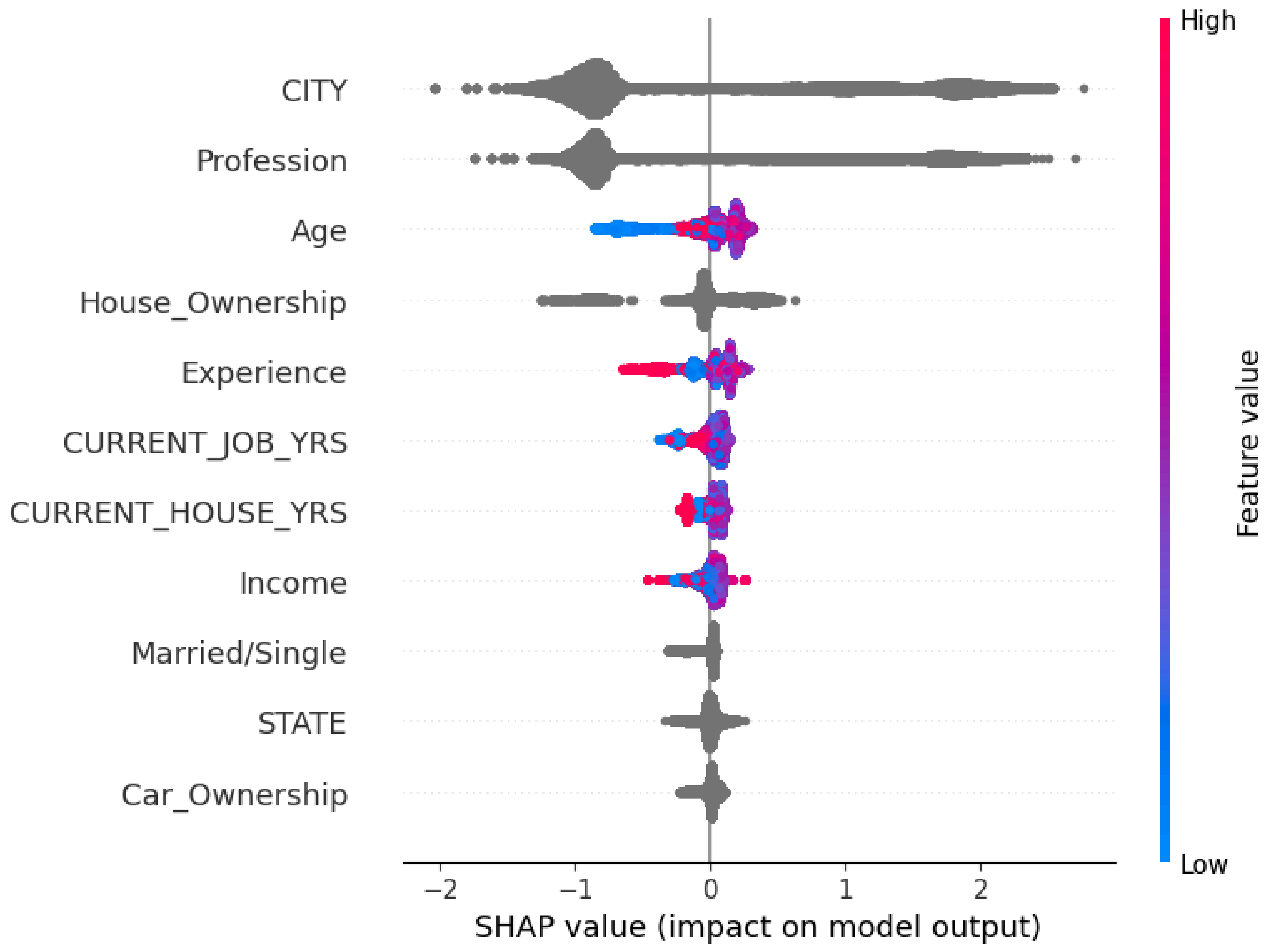
Default rate after ML optimization

a 11.64% drop in default rate

Model Evaluation

SHAP Analysis

- CITY and Profession have the most impact, with wide distributions of SHAP values.
- Age, Experience, and Income also influence predictions but to a lesser extent.
- Features like Car Ownership and Married/Single status have little impact.

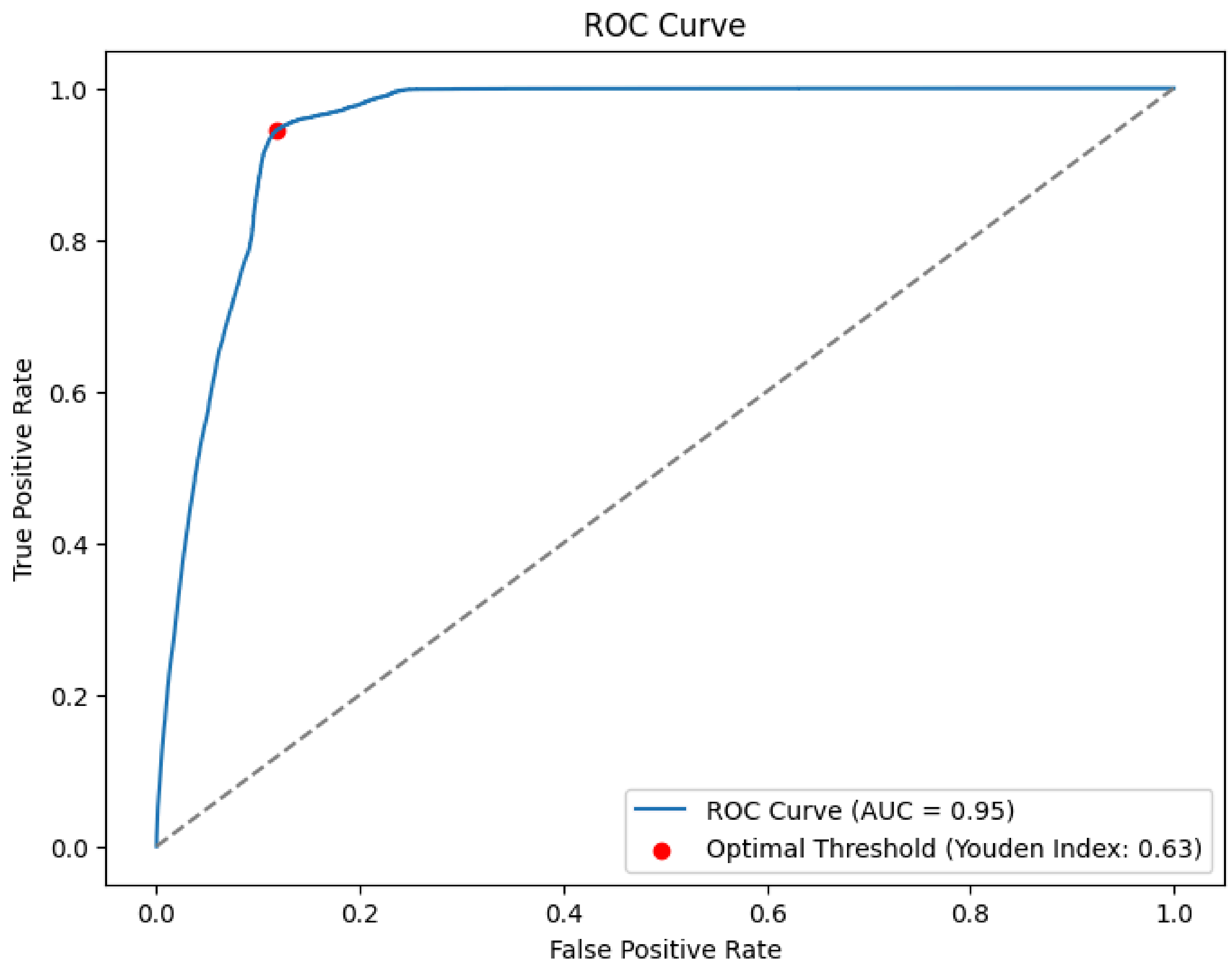


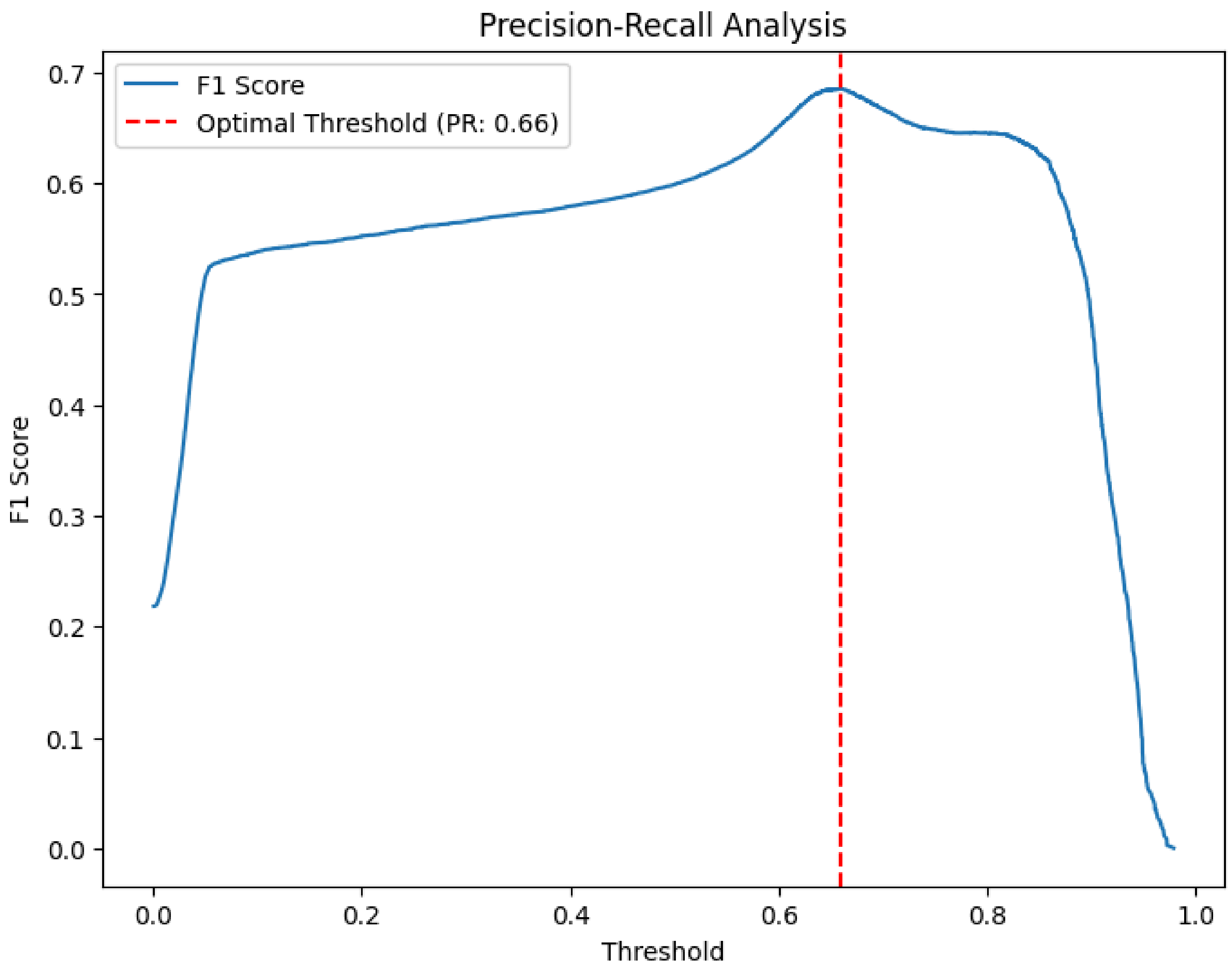
Confusion Matrix

	Predicted Not Default	Predicted Default
Actually Not Default	54574	11755
Actually Default	281	8990

Evaluation

Risk Threshold





Final Model Deployment & Use Case

Model Deployment

Backend: FastAPI

Frontend: Streamlit

app

Proyek analisis data ini bertujuan untuk membangun model machine learning yang dapat memprediksi ...

Streamlit

DataMinds
Analytics

Do you have any questions?

Send it to us! We hope you learned something new.