

---

# Classifying Loan Default

---

- This project involves predicting loan defaults, where misclassifications carry significant financial implications. A false positive (predicting default when none exists) costs a lot due to lost revenue, and a false negative (failing to predict an actual default) costs even more from unrecovered loan principal and recovery expenses. With false positives and negatives costing over millions annually, optimizing this model is critical to minimize financial losses.

```
In [1]: ► import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, f1_score, make_scorer, PrecisionRecallDisplay
from sklearn.model_selection import train_test_split, GridSearchCV, StratifiedKFold, cross_val_score

from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE

import optuna
import logging
optuna.logging.set_verbosity(optuna.logging.WARNING)
```

## Data

```
In [2]: ► df = pd.read_csv('Loan_default.csv')
```

In [3]: `df.head()`

Out[3]:

	LoanID	Age	Income	LoanAmount	CreditScore	MonthsEmployed	NumCreditLines	InterestRate	LoanTerm	DTIRatio	Education	En
0	I38PQUQS96	56	85994	50587	520	80	4	15.23	36	0.44	Bachelor's	
1	HPSK72WA7R	69	50432	124440	458	15	1	4.81	60	0.68	Master's	
2	C1OZ6DPJ8Y	46	84208	129188	451	26	3	21.17	24	0.31	Master's	
3	V2KKSFM3UN	32	31713	44799	743	0	3	7.07	24	0.23	High School	
4	EY08JDHTZP	60	20437	9139	633	8	4	6.51	48	0.73	Bachelor's	

## Clean

In [4]: `print(f'shape:{df.shape}')`  
`print(f'missing:{df.isna().sum().sum()}')`  
`print(f'duplicates:{df.duplicated().sum()}')`

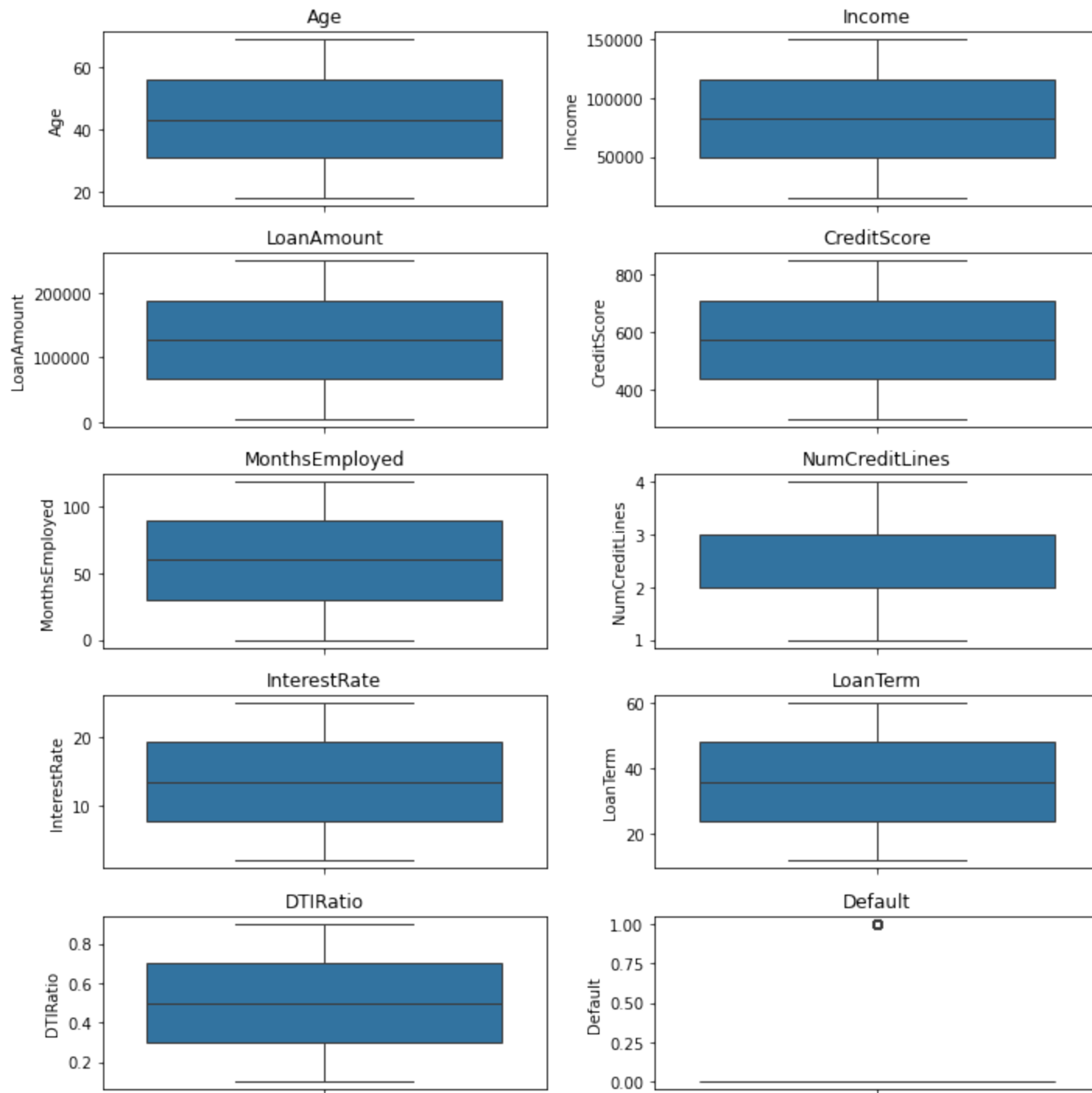
shape:(255347, 18)  
missing:0  
duplicates:0

In [5]: `df.drop(columns=['LoanID'], inplace=True)`

## Outliers

In [6]: `numerical = df.select_dtypes(include=[np.number])`

```
In [7]: ▶ fig, ax = plt.subplots(5,2, figsize=(10,10))
        ax = ax.flatten()
        ▼ for i,col in enumerate(numerical.columns):
            sns.boxplot(df[col], ax=ax[i])
            ax[i].set_title(f'{col}', fontsize=12)
        plt.tight_layout()
```



- There doesn't appear to be any outliers.

## Categorical Variables

```
In [8]: education_mappings = {"High School":1, "Bachelor's":2, "Master's":3, "PhD":4}
df['Education'] = df['Education'].map(education_mappings)
```

```
In [9]: dummies = df.select_dtypes(include=['object']).columns
df = pd.get_dummies(df, columns=dummies, drop_first=False)
```

```
In [10]: df.iloc[:,16:]
```

Out[10]:

	MaritalStatus_Married	MaritalStatus_Single	HasMortgage_No	HasMortgage_Yes	HasDependents_No	HasDependents_Yes	LoanPurpose_...
0	False	False	False	True	False	True	
1	True	False	True	False	True	False	
2	False	False	False	True	False	True	
3	True	False	True	False	True	False	
4	False	False	True	False	False	True	
...	...	...	...	...	...	...	
255342	True	False	True	False	True	False	
255343	False	False	True	False	True	False	
255344	True	False	False	True	False	True	
255345	False	True	False	True	False	True	
255346	False	False	False	True	True	False	

255347 rows × 13 columns

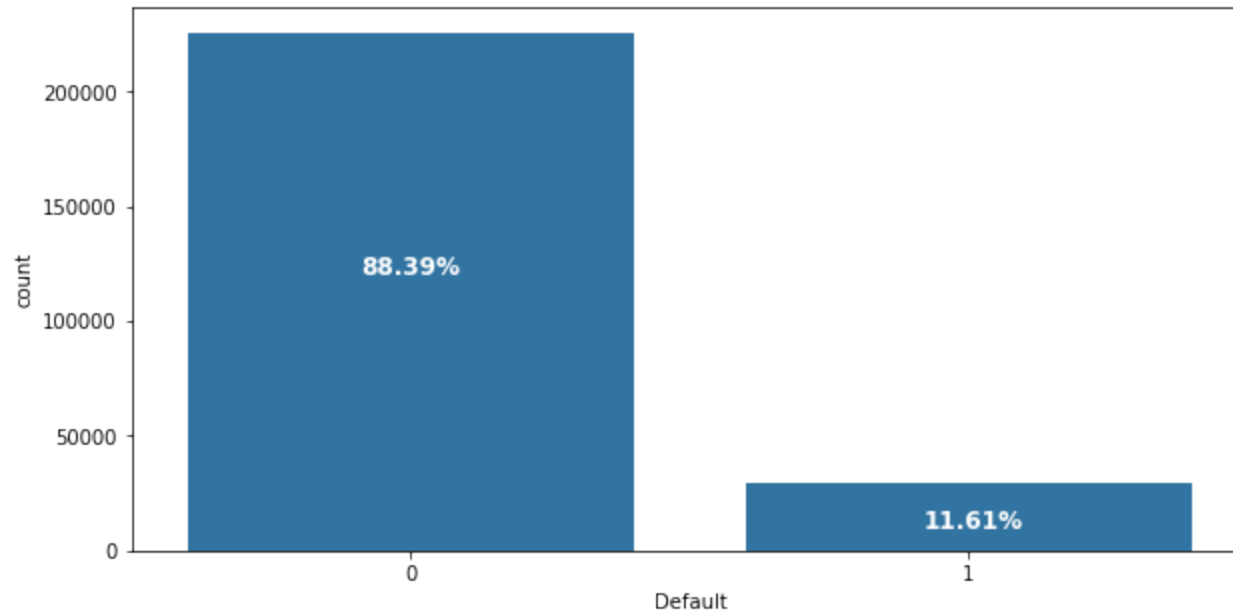
## Class Balance

```
In [11]: ▶ plt.figure(figsize=(10,5))

sns.barplot(df.Default.value_counts())

negative = len(df[df.Default==0])/len(df)*100
positive = len(df[df.Default==1])/len(df)*100

plt.text(-0.09, 120000, f'{round(negative,2)}%', fontsize = 12, color='white', weight='bold')
plt.text(0.92, 10000, f'{round(positive,2)}%', fontsize = 12, color='white', weight='bold');
```



- If someone guessed no every time they would be right about 88% of the time.

## Train/Validation/Test Split

```
In [31]: X = df.drop(columns='Default')
y = df.Default

# Train (60%), Temp (40%)
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=1, stratify=y)

# Validation (20%) and Test (20%) from Temp
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=1, stratify=y_temp)
```

- The target variable is stratified to ensure it is equally represented in the splits.

## Baseline Model

```
In [32]: model = RandomForestClassifier(random_state=1, n_jobs=-1)
model.fit(X_train,y_train)
y_pred = model.predict(X_val)
print(classification_report(y_val,y_pred))
```

	precision	recall	f1-score	support
0	0.89	1.00	0.94	45139
1	0.63	0.03	0.06	5930
accuracy			0.89	51069
macro avg	0.76	0.51	0.50	51069
weighted avg	0.86	0.89	0.84	51069

- This first model is getting 3% of the people who defaulted. The f1 score for the positive class is 0.06.

## Feature Engineering

```
In [33]: ▶ def credit_cats(score):
    ▼     if score >= 750:
    ▼         return 3
    ▼     elif score >= 650:
    ▼         return 2
    ▼     elif score >= 550:
    ▼         return 1
    ▼     else:
    ▼         return 0
```

```
In [34]: ▶ # features for training data
X_train['LoanToIncomeRatio'] = X_train['LoanAmount'] / X_train['Income']
X_train['InterestToIncomeRatio'] = X_train['InterestRate'] / X_train['Income']
X_train['MonthlyDebtToIncomeRatio'] = (X_train['LoanAmount'] / X_train['LoanTerm']) / (X_train['Income'] / 12)
X_train['CreditUtilization'] = X_train['LoanAmount'] / X_train['NumCreditLines']
X_train['LoanBurdenMonths'] = X_train['LoanAmount'] / (X_train['Income'] / 12)
X_train['CreditCategory'] = X_train['CreditScore'].apply(credit_cats)
X_train['LoanTermCategory'] = X_train['LoanTerm'].apply(lambda x: 0 if x < 36 else 1 if x < 60 else 2)

▼ X_train['LoanSizeCategory'] = pd.cut(X_train['LoanAmount'], bins=[0, 5000, 20000, 50000, 100000, np.inf],
    labels=[0, 1, 2, 3, 4])

# applying same features to validation and test set
▼ for df_subset in [X_val, X_test]:
    df_subset['LoanToIncomeRatio'] = df_subset['LoanAmount'] / df_subset['Income']
    df_subset['InterestToIncomeRatio'] = df_subset['InterestRate'] / df_subset['Income']
    df_subset['MonthlyDebtToIncomeRatio'] = (df_subset['LoanAmount'] / df_subset['LoanTerm']) / (df_subset['Income'] / 12)
    df_subset['CreditUtilization'] = df_subset['LoanAmount'] / df_subset['NumCreditLines']
    df_subset['LoanBurdenMonths'] = df_subset['LoanAmount'] / (df_subset['Income'] / 12)
    df_subset['CreditCategory'] = df_subset['CreditScore'].apply(credit_cats)
    df_subset['LoanTermCategory'] = df_subset['LoanTerm'].apply(lambda x: 0 if x < 36 else 1 if x < 60 else 2)

▼ df_subset['LoanSizeCategory'] = pd.cut(df_subset['LoanAmount'], bins=[0, 5000, 20000, 50000, 100000, np.inf],
    labels=[0, 1, 2, 3, 4])
```

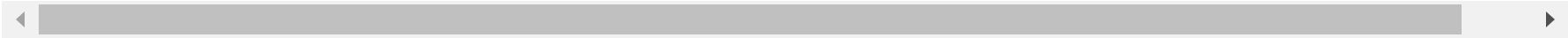


```
In [35]: X_train.iloc[:, 29:]
```

Out[35]:

	InterestToIncomeRatio	MonthlyDebtToIncomeRatio	CreditUtilization	LoanBurdenMonths	CreditCategory	LoanTermCategory	LoanSizeCa
81133	0.000052	0.585708	76387.500000	14.057001	0	0	
165234	0.000142	0.493794	149104.000000	23.702102	2	1	
242324	0.000115	0.187743	53238.000000	9.011680	2	1	
224777	0.000280	0.560673	96087.000000	26.912299	3	1	
115292	0.000023	0.164290	21442.500000	3.942965	0	0	
...	...	...	...	...	...	...	...
52242	0.000051	0.345334	25627.666667	8.288006	2	0	
96610	0.000116	0.166025	85485.000000	7.969205	0	1	
41562	0.000079	0.559495	59509.250000	33.569679	0	2	
235280	0.000141	1.713994	55318.000000	20.567923	1	0	
61614	0.000138	1.785579	62034.000000	21.426951	0	0	

153208 rows × 7 columns



## Model 2

```
In [36]: model2 = RandomForestClassifier(random_state=1, n_jobs=-1)
model2.fit(X_train,y_train)
y_pred = model2.predict(X_val)
print(classification_report(y_val,y_pred))
```

	precision	recall	f1-score	support
0	0.89	1.00	0.94	45139
1	0.60	0.04	0.08	5930
accuracy			0.89	51069
macro avg	0.75	0.52	0.51	51069
weighted avg	0.86	0.89	0.84	51069

- The recall increased to 4% and the f1 score increased to 0.08.
- 



## **Feature Elimination (maximize f1 for positive class)**

```

In [37]: ▶ remaining_features = list(X_train.columns[:])
best_f1_macro = 0
best_features = None

original_X_train, original_X_val = X_train.copy(), X_val.copy()

▼ while len(remaining_features) > 1:
    model.fit(X_train[remaining_features], y_train)

    # predict on validation set and compute f1
    y_pred = model.predict(X_val[remaining_features])
    f1_macro = f1_score(y_val, y_pred, pos_label=1)
    print(f"Features: {remaining_features},\nValidation F1 Score: {f1_macro:.4f}")

    # track best feature set
    ▼ if f1_macro > best_f1_macro:
        best_f1_macro = f1_macro
        best_features = remaining_features[:]

    # find least important feature
    feature_importances = model.feature_importances_
    least_important_idx = np.argmin(feature_importances)
    least_important_feature = remaining_features[least_important_idx]

    # remove least important feature
    print(f"Removing least important feature: {least_important_feature}\n")
    remaining_features.pop(least_important_idx)

print('-' * 100)
print("\nBest Features Selected:", best_features)
print(f"Best Validation F1 Score: {best_f1_macro:.4f}")

```

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Full-time', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'HasDependents\_Yes', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Home', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'HasCoSigner\_Yes', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory', 'LoanSizeCategory'],

Validation F1 Score: 0.0818

Removing least important feature: HasCoSigner\_Yes

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Full-time', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'HasDependents\_Yes', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Home', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory', 'LoanSizeCategory'],

Validation F1 Score: 0.0823

Removing least important feature: LoanSizeCategory

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Full-time', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'HasDependents\_Yes', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Home', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0830

Removing least important feature: EmploymentType\_Full-time

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'HasDependents\_Yes', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Home', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0839

Removing least important feature: HasDependents\_Yes

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Home', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0818

Removing least important feature: LoanPurpose\_Home

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Married', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0868  
Removing least important feature: MaritalStatus\_Married

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Education', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0853  
Removing least important feature: LoanPurpose\_Education

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_No', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0827  
Removing least important feature: HasMortgage\_No

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Auto', 'LoanPurpose\_Business', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0873  
Removing least important feature: LoanPurpose\_Auto

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Self-employed', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Business', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0846  
Removing least important feature: EmploymentType\_Self-employed

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Business', 'LoanPurpose\_Other', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0846  
Removing least important feature: LoanTermCategory

rdenMonths', 'CreditCategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0879  
Removing least important feature: LoanPurpose\_Other

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'Loan  
Term', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced',  
'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanPurpose\_Business', 'HasCoSigner\_No', 'LoanToIn  
comeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditC  
ategory', 'LoanTermCategory'],  
Validation F1 Score: 0.0912  
Removing least important feature: LoanPurpose\_Business

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'HasCoSigner\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0880

Removing least important feature: HasCoSigner\_No

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Part-time', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0890

Removing least important feature: EmploymentType\_Part-time

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'HasDependents\_No', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0815

Removing least important feature: HasDependents\_No

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'MaritalStatus\_Single', 'HasMortgage\_Yes', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0885

Removing least important feature: MaritalStatus\_Single

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'EmploymentType\_Unemployed', 'MaritalStatus\_Divorced', 'HasMortgage\_Yes', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0890

Removing least important feature: EmploymentType\_Unemployed

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'MaritalStatus\_Divorced', 'HasMortgage\_Yes', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0929

Removing least important feature: MaritalStatus\_Divorced

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'HasMortgage\_Yes', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0907

Removing least important feature: HasMortgage\_Yes

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory', 'LoanTermCategory'],

Validation F1 Score: 0.0948

Removing least important feature: LoanTermCategory

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths', 'CreditCategory'],

Validation F1 Score: 0.0948

Removing least important feature: CreditCategory

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0898

Removing least important feature: NumCreditLines

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'InterestRate', 'LoanTerm', 'DTIRatio', 'Education', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0910

Removing least important feature: LoanTerm

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'InterestRate', 'DTIRatio', 'Education', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0940

Removing least important feature: Education

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'InterestRate', 'DTIRatio', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0995

Removing least important feature: DTIRatio

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0987

Removing least important feature: MonthsEmployed

Features: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],

Validation F1 Score: 0.0850

Removing least important feature: Age

Features: ['Income', 'LoanAmount', 'CreditScore', 'InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'M



onthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0548  
Removing least important feature: LoanAmount

Features: ['Income', 'CreditScore', 'InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0502  
Removing least important feature: CreditScore

Features: ['Income', 'InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0535  
Removing least important feature: Income

Features: ['InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0529  
Removing least important feature: MonthlyDebtToIncomeRatio

Features: ['InterestRate', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0648  
Removing least important feature: InterestRate

Features: ['LoanToIncomeRatio', 'InterestToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths'],  
Validation F1 Score: 0.0732  
Removing least important feature: LoanBurdenMonths

Features: ['LoanToIncomeRatio', 'InterestToIncomeRatio', 'CreditUtilization'],  
Validation F1 Score: 0.0626  
Removing least important feature: CreditUtilization

Features: ['LoanToIncomeRatio', 'InterestToIncomeRatio'],  
Validation F1 Score: 0.0779  
Removing least important feature: LoanToIncomeRatio

-----  
Best Features Selected: ['Age', 'Income', 'LoanAmount', 'CreditScore', 'MonthsEmployed', 'InterestRate', 'DTIRatio', 'LoanToIncomeRatio', 'InterestToIncomeRatio', 'MonthlyDebtToIncomeRatio', 'CreditUtilization', 'LoanBurdenMonths']  
Best Validation F1 Score: 0.0995

- 
- The f1 score for the positive class increased from 0.0818 to 0.0995. It appears the created features were useful as 5 out of 8 of them made it through this selection process.

## Model 3

```
In [19]: X_train = X_train[best_features]
X_val = X_val[best_features]
X_test = X_test[best_features]
```

```
In [20]: model3 = RandomForestClassifier(random_state=1, n_jobs=-1)
model3.fit(X_train,y_train)
y_pred = model3.predict(X_val)
print(classification_report(y_val,y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.99	0.94	45139
1	0.54	0.05	0.10	5930
accuracy			0.88	51069
macro avg	0.71	0.52	0.52	51069
weighted avg	0.85	0.88	0.84	51069

## Hyperparameter Tuning

```

In [21]: ▶ def objective(trial):
    # hyperparameters
    n_estimators = trial.suggest_int('n_estimators', 100, 300, step=50)
    max_depth = trial.suggest_categorical('max_depth', [10, 20, 30, None])
    min_samples_split = trial.suggest_int('min_samples_split', 2, 10, step=2)
    min_samples_leaf = trial.suggest_int('min_samples_leaf', 1, 11, step=2)
    class_weight = trial.suggest_categorical('class_weight', [None, 'balanced', 'balanced_subsample'])

    #classifier
    rf = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        min_samples_split=min_samples_split,
        min_samples_leaf=min_samples_leaf,
        class_weight=class_weight,
        random_state=1,
        n_jobs=-1)

    # cross val
    skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
    scorer = make_scorer(f1_score, average='binary', pos_label=1)
    scores = cross_val_score(rf, X_train, y_train, cv=skf, scoring=scorer, n_jobs=-1)
    return scores.mean()

    # run optimization
    study = optuna.create_study(direction='maximize')
    study.optimize(objective, n_trials=50, n_jobs=-1)

    print("Best Parameters:", study.best_params)
    print("Best F1 Score for Class 1:", study.best_value)

```

Best Parameters: {'n\_estimators': 300, 'max\_depth': 10, 'min\_samples\_split': 6, 'min\_samples\_leaf': 7, 'class\_weight': 'balanced\_subsample'}

Best F1 Score for Class 1: 0.3429166252101683

## Model 4

In [27]: `best_params = study.best_params`

```
best_rf = RandomForestClassifier(  
    **best_params,  
    random_state=1,  
    n_jobs=-1)
```

```
best_rf.fit(X_train, y_train)  
y_pred = best_rf.predict(X_val)
```

In [28]: `print(classification_report(y_val,y_pred))`

	precision	recall	f1-score	support
0	0.93	0.75	0.83	45139
1	0.24	0.60	0.34	5930
accuracy			0.73	51069
macro avg	0.59	0.67	0.59	51069
weighted avg	0.85	0.73	0.78	51069

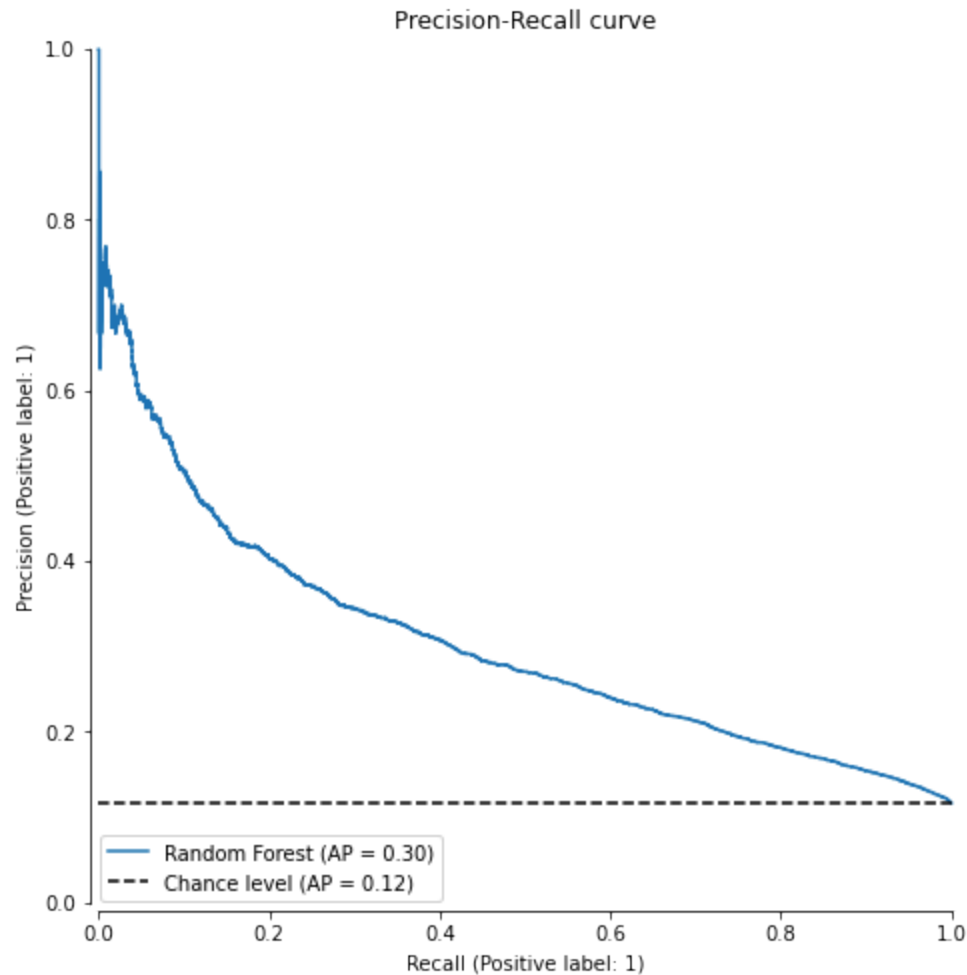
- The recall is up to 60% after tuning, but the precision went down to 24%. This model might offer a better tradeoff at a different threshold.

## Precision-Recall Curve

```
In [30]: fig, ax = plt.subplots(figsize=(14, 8))

display = PrecisionRecallDisplay.from_estimator(
    best_rf, X_val, y_val, name="Random Forest", plot_chance_level=True, despine=True, ax=ax
)

_ = display.ax_.set_title("Precision-Recall curve")
```



- The average precision (AP) for the Random Forest model is 0.30 which indicates the model has some predictive power but it is still low.
- The chance level (baseline) AP is 0.12 so the model is performing better than random guessing but there is room for significant improvement.
- To show a significant improvement over a naive approach a recall  $\geq 60\text{--}70\%$  and precision  $\geq 40\%$  is the goal.

- This is a work in progress. My next approach will be to use an ensemble of models (random forest + lightgbm + catboost) to take a majority vote. These models will be tuned on the training and validation set, and if they can achieve those metrics, the last step will be verifying performance with the test set.
-