Loan default detection

Katherine Oghalis Bravo Fernandez
Data Analyst

Table of Contents

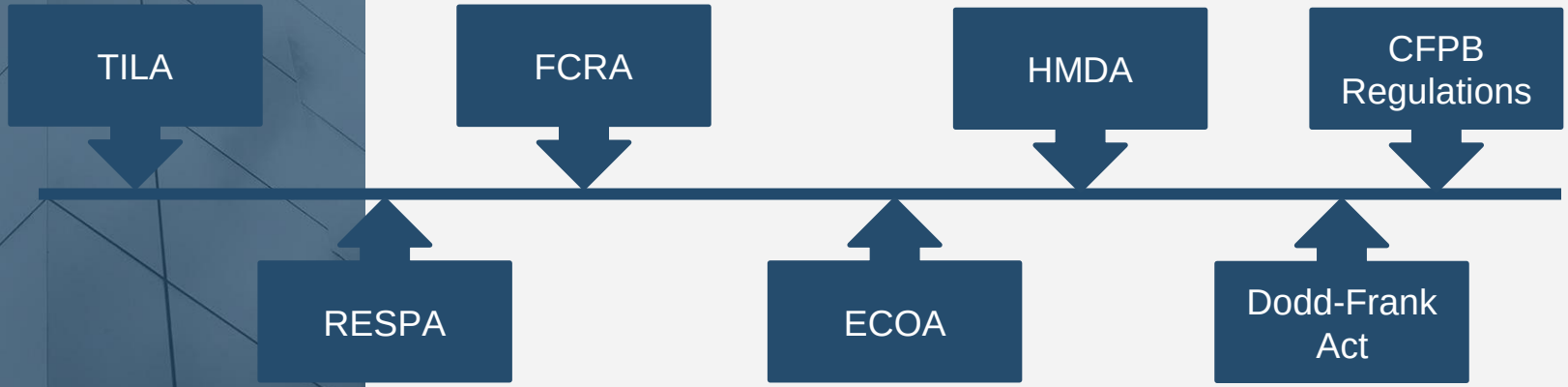
01 Introduction

The term "loan default" refers to when a borrower fails to meet the conditions of a loan, such as interest or principal payments, due to financial difficulties, job loss, or other economic problems. Banks, which generate significant revenue through mortgage loans, face the risk of default by borrowers. Traditional risk assessment methods are not always accurate, which is why this project focuses on solving this issue by using advanced Machine Learning techniques to provide a more accurate prediction and help banks mitigate the risk of mortgage loan defaults.

Problem & Context

Bibliography

To address the issue of "loan default" in the United States, it is important to become familiar with several key regulations and guidelines that impact the mortgage lending process and risk management. Below are some of the most relevant regulations:



EDA

02

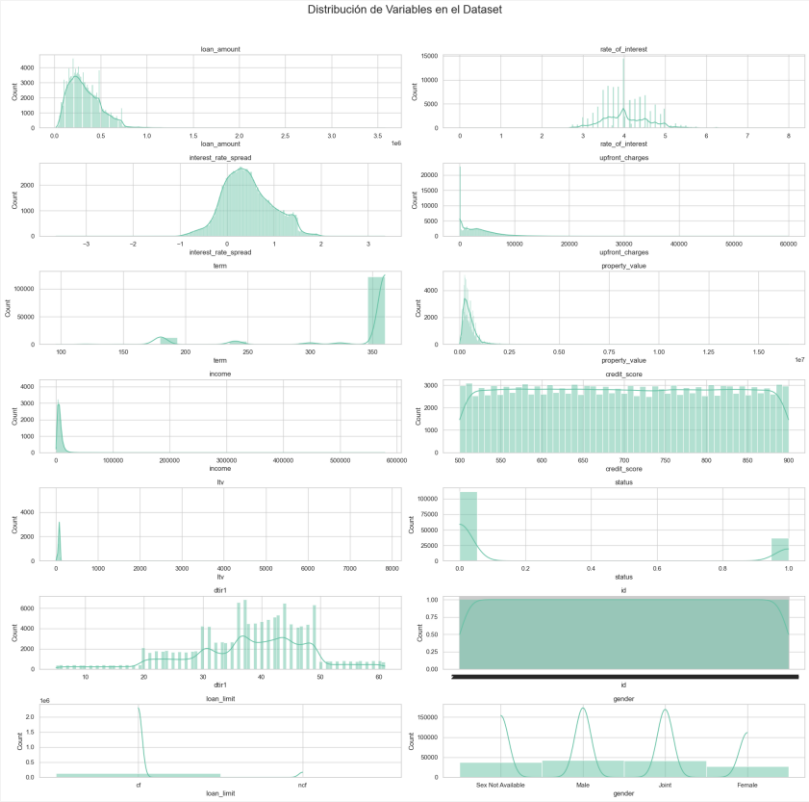
Dataset

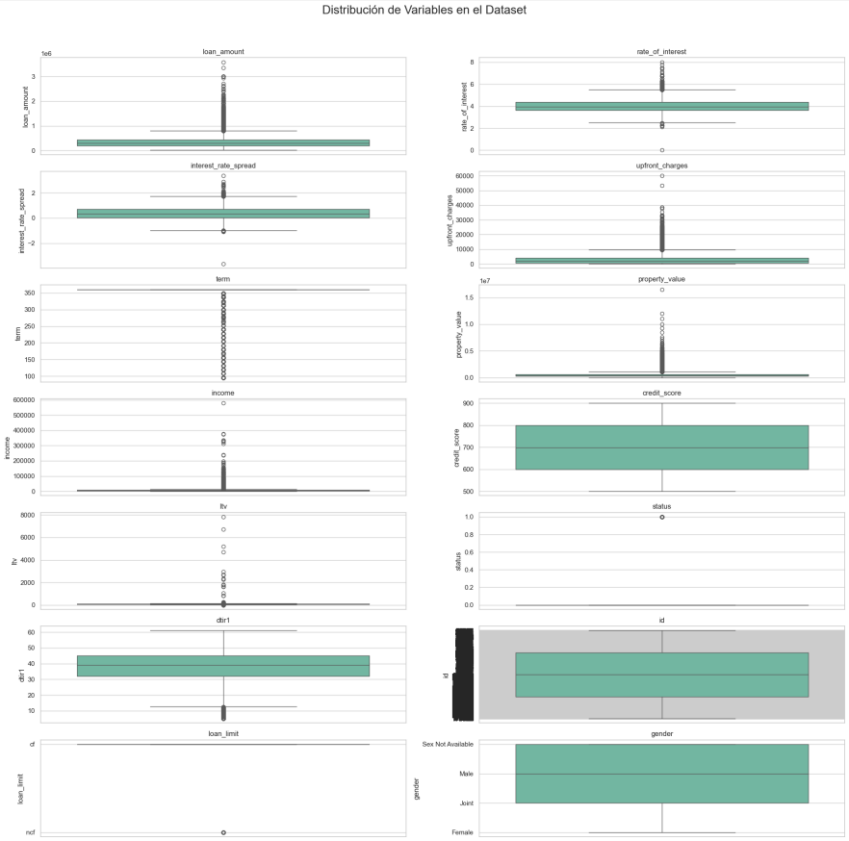
```
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   ID                         148670 non-null  int64
 1   year                       148670 non-null  int64
 2   loan_limit                 145326 non-null  object
 3   Gender                     148670 non-null  object
 4   approv_in_adv              147762 non-null  object
 5   loan_type                  148670 non-null  object
 6   loan_purpose               148536 non-null  object
 7   Credit_Worthiness          148670 non-null  object
 8   open_credit                148670 non-null  object
 9   business_or_commercial     148670 non-null  object
 10  loan_amount                148670 non-null  int64
 11  rate_of_interest           112231 non-null  float64
 12  Interest_rate_spread       112031 non-null  float64
 13  Upfront_charges            109028 non-null  float64
 14  term                       148629 non-null  float64
 15  Neg_ammortization          148549 non-null  object
 16  interest_only              148670 non-null  object
 17  lump_sum_payment           148670 non-null  object
 18  property_value             133572 non-null  float64
 19  construction_type          148670 non-null  object
 20  occupancy_type             148670 non-null  object
 21  Secured_by                 148670 non-null  object
 22  total_units                148670 non-null  object
 23  income                     139520 non-null  float64
 24  credit_type                148670 non-null  object
 25  Credit_Score               148670 non-null  int64
 26  co-applicant_credit_type   148670 non-null  object
 27  age                        148470 non-null  object
 28  submission_of_application  148470 non-null  object
 29  LTV                        133572 non-null  float64
 30  Region                     148670 non-null  object
 31  Security_Type              148670 non-null  object
 32  Status                     148670 non-null  int64
 33  dtir1                      124549 non-null  float64
```

Column	Type	Description
loan_amount	int64	Total loan amount granted
rate_of_interest	float64	Interest rate applied to the loan (%)
Credit_Score	int64	Credit score of the borrower
income	float64	Income of the borrower
LTV	float64	Loan-to-value ratio
term	float64	Loan term in months
property_value	float64	Property value
dti	float64	Debt-to-income ratio

These are the top 8 feature that can influence the prediction of "loan default".

The whole dataset has 34 columns and 148,669 records.


Distribución de Variables en el Dataset
loan_amount
rate_of_interest
interest_rate_spread
upfront_charges
term
property_value
income
credit_score
ltv
status
dtir1
id
loan_limit
gender
Count
cf
ncf
Sex Not Available
Male
Joint
Female


Distribución de Variables en el Dataset
loan_amount
rate_of_interest
interest_rate_spread
upfront_charges
term
property_value
income
credit_score
ltv
status
dtir1
id
loan_limit
gender
cf
ncf
Sex Not Available
Male
Joint
Female

03 Key fidings & insights

Model 04

Deep Learning

```
model_reg = tf.keras.models.Sequential([
    Dense(512, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(256, activation='relu', kernel_regularizer=regularizers.l2(0.01)),
    BatchNormalization(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

Random Forest

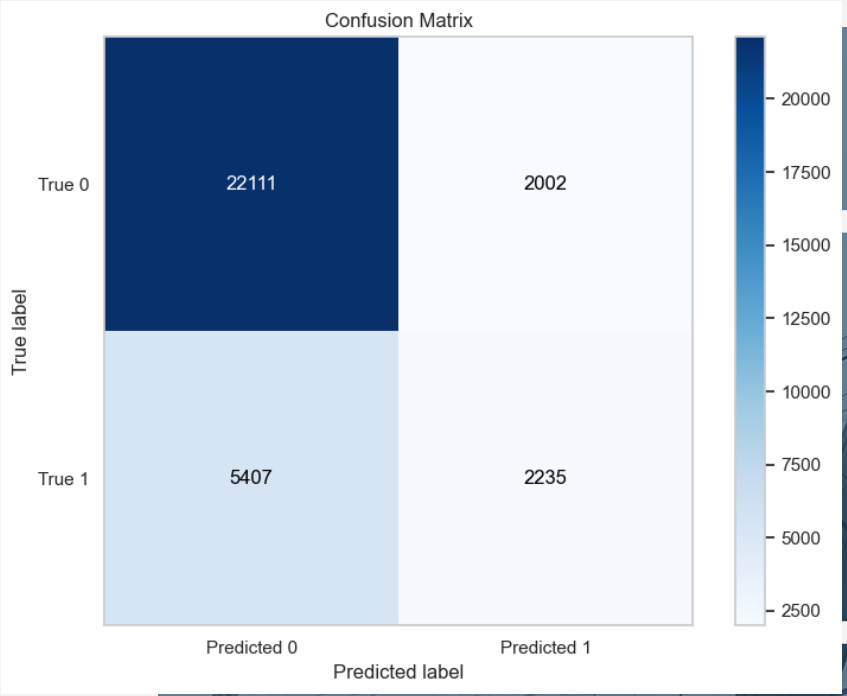
```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_features': [2, 4, 6],
    'bootstrap': [True, False],
}

forest_class = RandomForestClassifier()

grid_search = GridSearchCV(forest_class, param_grid, cv=5,
                           scoring = 'accuracy', verbose = 2, n_jobs = -1)

grid_search.fit(X_train_pca, y_train)
```

Model description

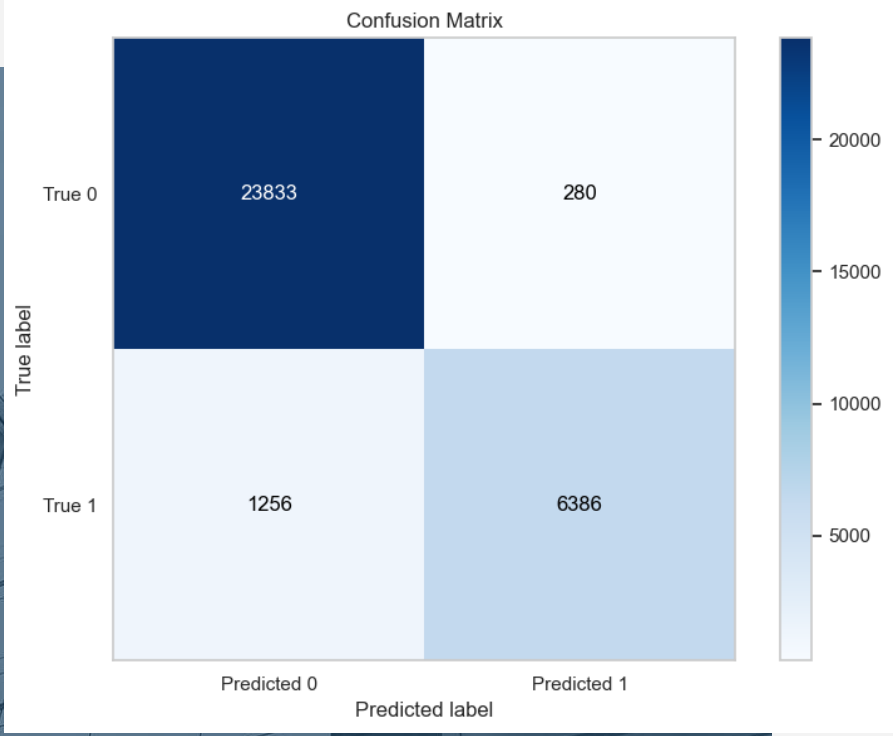
The image shows a grid search setup for a Random Forest classifier to find the best hyperparameters for a classification task. The parameter grid (param_grid) includes options for the number of trees (n_estimators: 50, 100, 200), the maximum number of features considered for splitting a node (max_features: 2, 4, 6), and whether bootstrap samples are used (bootstrap: True, False). The RandomForestClassifier is defined and a GridSearchCV object is created to perform 5-fold cross-validation (cv=5). The grid search aims to optimize the model based on accuracy (scoring='accuracy'). The fitting process is carried out on the training data (X_train_pca, y_train) to find the best combination of parameters.

The param_grid dictionary specifies the hyperparameters to be tested: n_estimators (number of trees in the forest) with values 50, 100, and 200; max_features (maximum number of features considered for splitting a node) with values 2, 4, and 6; and bootstrap (whether bootstrap samples are used) with True and False options. A RandomForestClassifier instance is created and then passed to GridSearchCV along with the param_grid to perform a 5-fold cross-validation (cv=5). The grid search is configured to optimize for accuracy (scoring='accuracy'). The grid_search object is set to verbose mode (level 2) for detailed output and to use all available processors (n_jobs=-1). The fit method is called on the training data (X_train_pca, y_train) to find the best combination of hyperparameters.

Deep Learning



Random Forest



05 Conclusions

Dataset

- En base a las observaciones, se recomienda que los bancos consideren ajustar las tasas de interés en función del riesgo percibido en diferentes zonas geográficas. Dado que los intereses son significativamente más altos en la zona norte en comparación con la zona centro, los bancos podrían evaluar los factores específicos que contribuyen a este mayor riesgo percibido. Al hacer esto, podrían implementar medidas de mitigación específicas para reducir el riesgo en áreas de alto interés, como ofrecer programas de educación financiera o productos de seguro adicionales para los prestatarios.

- Otra recomendación es que los bancos presten atención a la demografía de los prestatarios, especialmente en relación con el género y la edad. La observación de que los hombres en la zona norte son más jóvenes que las mujeres en la misma región puede indicar diferentes necesidades y comportamientos financieros entre estos grupos. Los bancos podrían diseñar productos financieros personalizados que se adapten mejor a las características demográficas y necesidades de cada grupo, mejorando así la satisfacción del cliente y reduciendo el riesgo de incumplimiento.

- Finalmente, se sugiere que los bancos continúen utilizando técnicas avanzadas de Machine Learning para evaluar el riesgo de incumplimiento. La construcción de modelos de Machine Learning robustos que utilicen datos históricos y características determinísticas ayudará a los bancos a identificar patrones y predictores de incumplimiento con mayor precisión. Además, compartir los resultados y el código en plataformas públicas como GitHub fomentará la colaboración y la mejora continua de estos modelos.

Thanks

You can find the spanish version here!

https://github.com/Oghalis/proyecto

Contact info:

https://www.linkedin.com/in/katherineoghalisbravo/