ITBA
Instituto Tecnológicc
de Buenos Aires

Predicción de Aprobación de Préstamos

FINAL Análisis Predictivo

Nicolás Peric

Caso de Negocio - Bancos

Objetivo: Identificar quienes o no puedan acceder a un préstamo conlleva muchos recursos, un modelo predictivo con el fin de predecir si un cliente puede o no acceder puede reducir costos asociados al riesgo de otorgar préstamos. Se busca **reducir** riesgos, manteniendo **altas** las tasas de aprobación.

DATASET

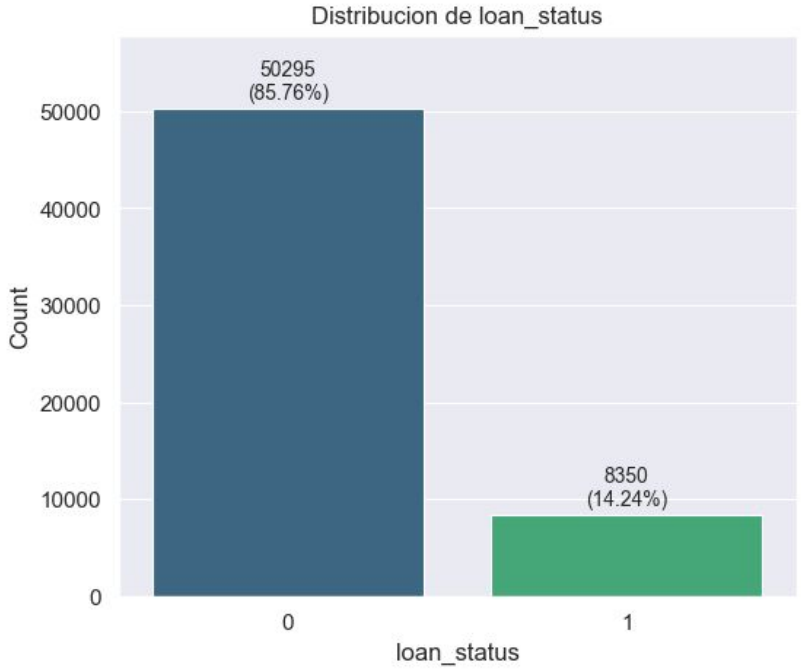
https://www.kaggle.com/competitions/playground-series-s4e10/overview

Train: 58645 → 80%

20%

Test: 39098 → Validación

- id
- person_age
- person_income
- person_home_ownership
- person_emp_length
- loan_intent
- loan_grade
- loan_amount
- loan_int_rate
- loan_percent_income
- cb_person_default_on_file
- cb_person_cred_hist_length
- **loan_status → Target**



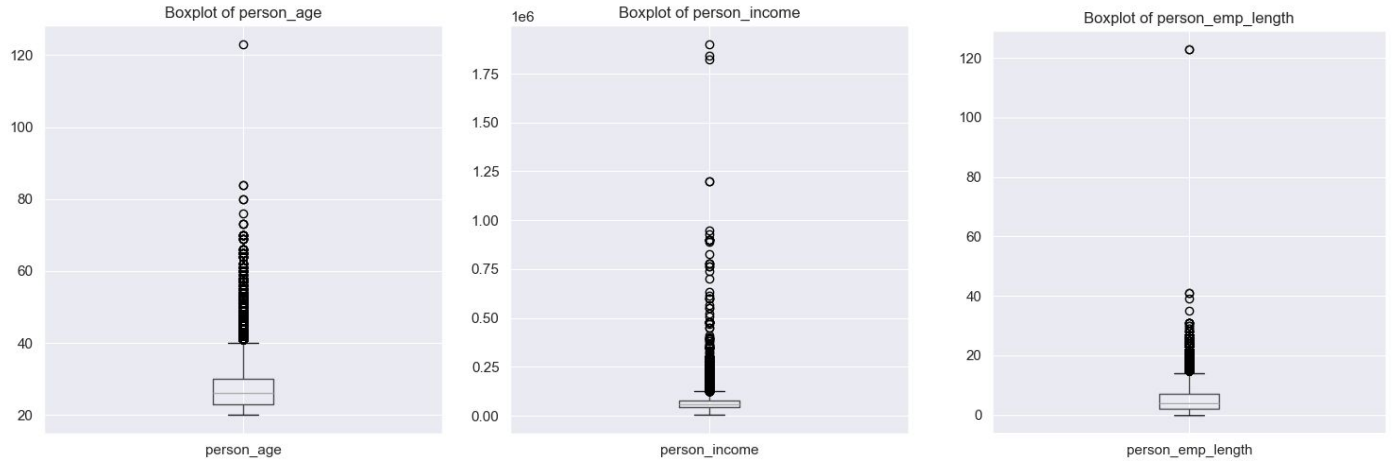
Métrica de Evaluación:

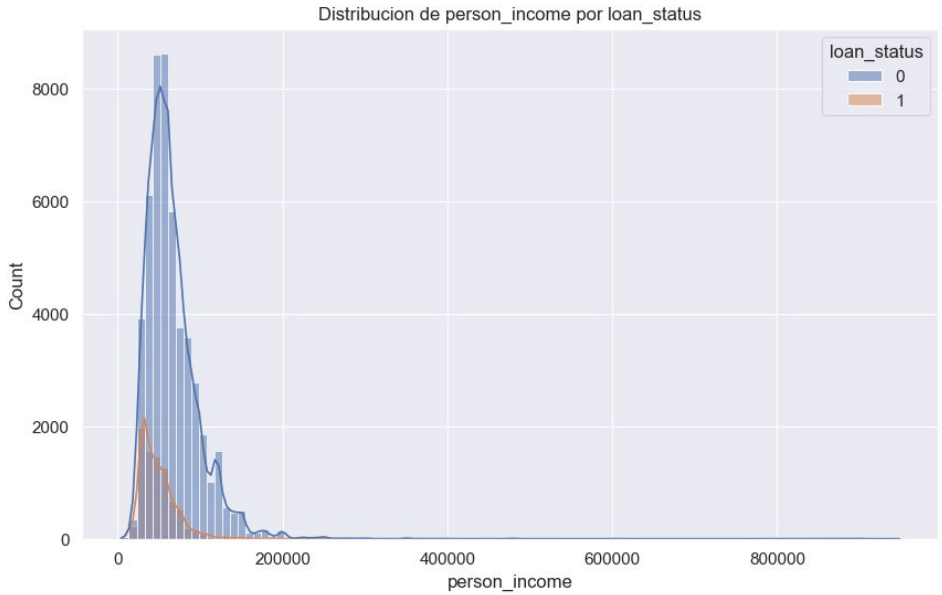
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

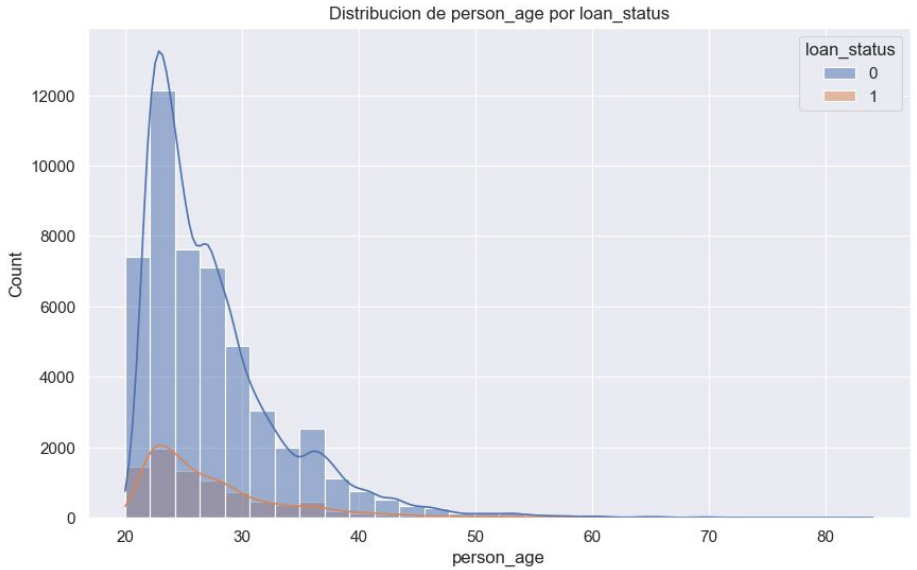
$$\text{Weighted F1 Score} = \sum_{i=1}^{N} w_i \times \text{F1 Score}_i$$

Validación del rendimiento del modelo con Kaggle (AUC).

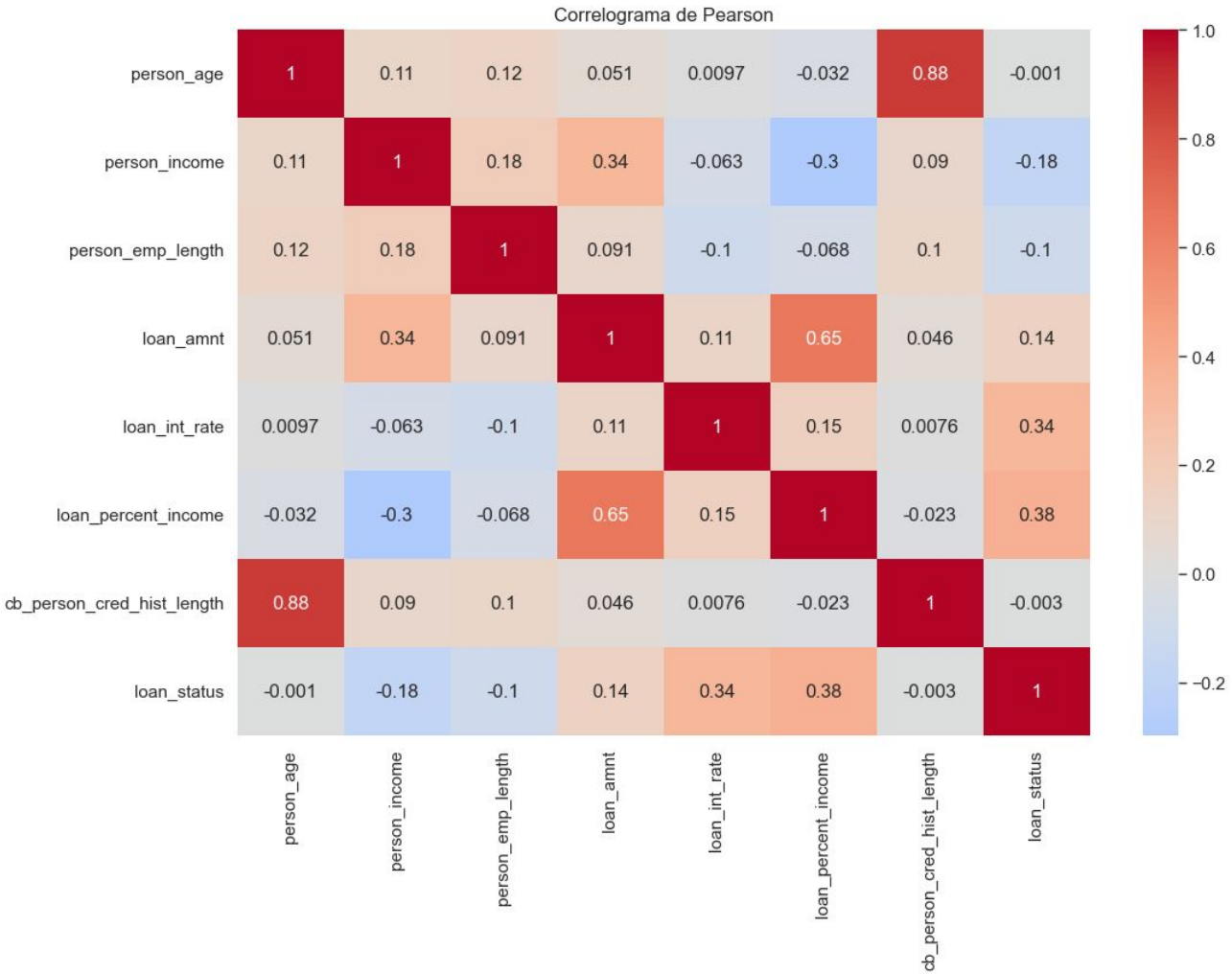
EDA

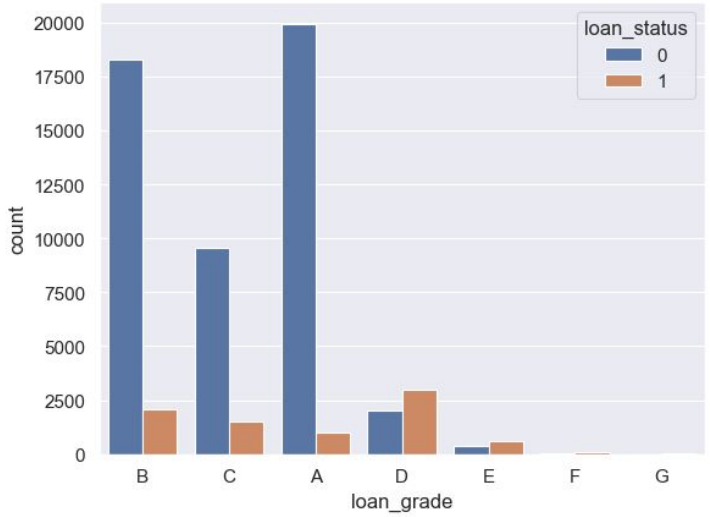

Boxplot of person_age
person_age
Boxplot of person_income
1e6
person_income
Boxplot of person_emp_length
person_emp_length

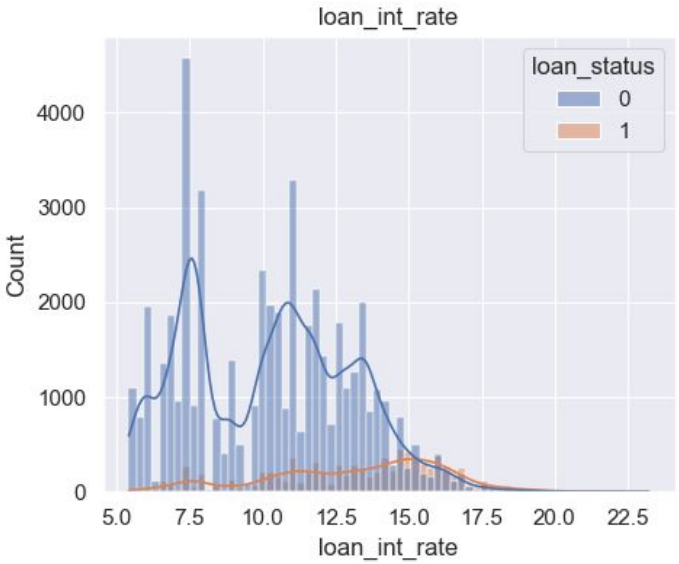

Distribucion de person_income por loan_status
loan_status
0
1
Count
person_income


Distribucion de person_age por loan_status
loan_status
0
1
Count
person_age

EDA


Correlograma de Pearson
person_age 1 0.11 0.12 0.051 0.0097 -0.032 0.88 -0.001
person_income 0.11 1 0.18 0.34 -0.063 -0.3 0.09 -0.18
person_emp_length 0.12 0.18 1 0.091 -0.1 -0.068 0.1 -0.1
loan_amnt 0.051 0.34 0.091 1 0.11 0.65 0.046 0.14
loan_int_rate 0.0097 -0.063 -0.1 0.11 1 0.15 0.0076 0.34
loan_percent_income -0.032 -0.3 -0.068 0.65 0.15 1 -0.023 0.38
cb_person_cred_hist_length 0.88 0.09 0.1 0.046 0.0076 -0.023 1 -0.003
loan_status -0.001 -0.18 -0.1 0.14 0.34 0.38 -0.003 1
person_age
person_income
person_emp_length
loan_amnt
loan_int_rate
loan_percent_income
cb_person_cred_hist_length
loan_status
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


loan_status
0
1
count
20000
17500
15000
12500
10000
7500
5000
2500
0
B
C
A
D
E
F
G
loan_grade


loan_int_rate
loan_status
0
1
Count
4000
3000
2000
1000
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
loan_int_rate

Features

Se crearon nuevas columnas con el fin de mejorar el rendimiento de los modelos:

income_loan_ratio = person_income/loan_amnt

age_category = categorizar las edades en grupos: young adult, adult, senior. (19-30, 30-55, 55+)

emp_length_group = categorizar en grupos el employment length.

age_income_ratio = person_age/person_income

emp_age_ratio = person_emp_length/person_age

age_loan_ratio = person_age/loan_amnt

Se utilizaron pipelines y ColumnTransformer para transformar las variables numéricas y categóricas utilizando StandardScaler() y OneHotEncoder().

Modelos

Modelo	Accuracy	Precision	Recall	F1-Score	AUC	K. Private	K. Public
Logistic Regression Base	0.91269	0.90601	0.9127	0.9059	0.9019	0.89888	0.89943
Logistic Regression SMOTE	0.8442	0.8968	0.8442	0.8603	0.9048	0.90073	0.90133
Random Forest	0.9500	0.9494	0.9500	0.9469	0.9359	0.93124	0.93821
LightGBM	**0.9532**	**0.9519**	**0.9532**	**0.9511**	**0.9588**	**0.95757**	**0.95757**
KNN	0.9301	0.9265	0.9301	0.9256	0.8902	0.88237	0.88486
XGBoost	0.9527	0.9515	0.9527	0.9506	0.9557	0.95394	0.95603

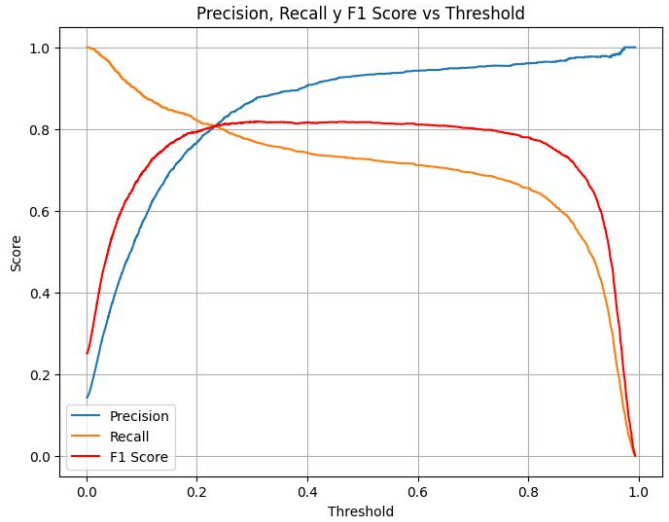
*Las métricas se midieron con weighted average.

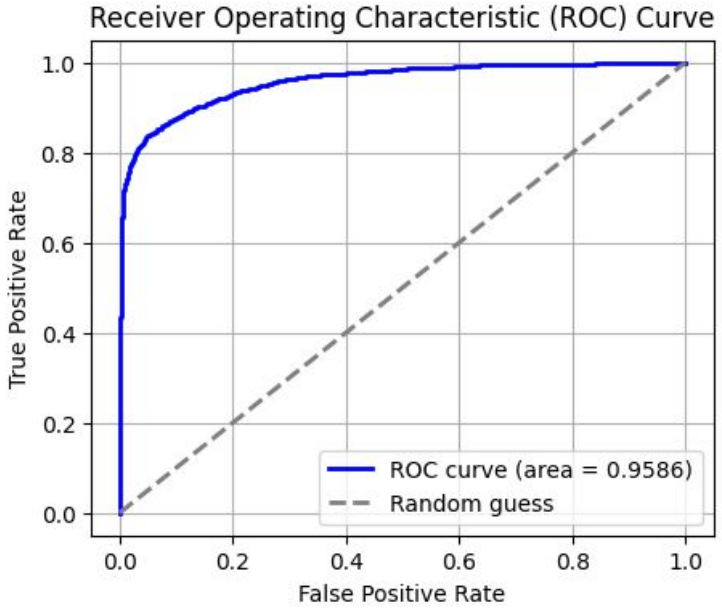
En todos los modelos, se usa validación cruzada estratificada (5 folds), se definieron una distribución de hiperparámetros y RandomizedSearchCV para encontrar la mejor combinación usando el AUC como métrica. Se entrena el modelo final con los mejores hiperparámetros.

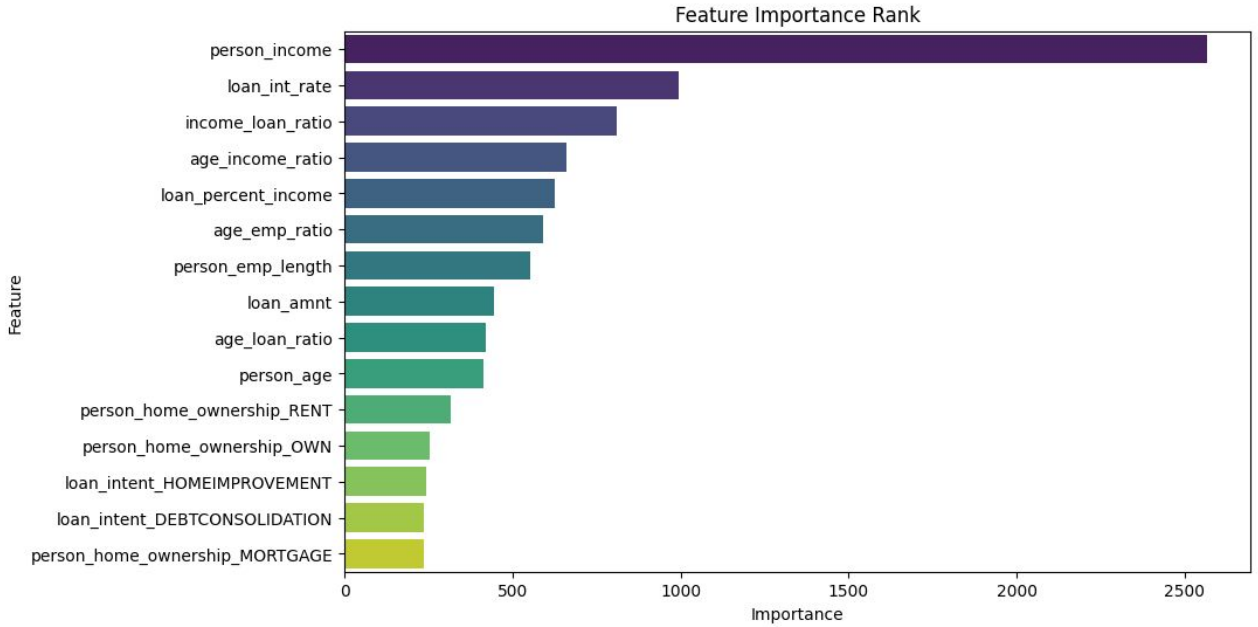
Modelo Final

LightGBM con hiperparámetros:

- Learning Rate: 0.023
- max_depth: 10
- n_estimators: 365







Conclusión

- LightGBM y XGBoost fueron los mejores modelos, siendo el primero el mejor de los dos.
- Además, LightGBM entrena a una velocidad mayor, ayudando al caso de negocio.
- **Recomendación:** Medir con AUC la capacidad del modelo de separar las clases y F1 Score el rendimiento del modelo ya que queremos asegurar un equilibrio entre riesgo y oportunidades.

Posibles Mejoras

- Enriquecimiento de la base, recopilar más variables.
- Probar KNN con Smote.
- Probar múltiples modelos a la vez con CV en lugar de probar uno por uno.
- Probar otras métricas de interpretabilidad como valores SHAP.

Muchas gracias!

Link al repositorio:
https://github.com/Pericsen/loan-approval-prediction