Predicción de crédito bancario

Bootcamp de Ciencia de Datos

Enero 2024

Equipo 28

Diana Mory

Data Scientist Jr.

Ing. Informático, apasionada por la información inmersa en los datos

dianamorypaz

Fabián Trejo

Data Scientist Jr.

Matemático convencido que los datos nos ayudan a describir y entender el mundo

fabiantrejomath

Alejandra Cruz

Data Scientist Jr.

Ingeniera de Sistemas Incursionando en el mundo de los datos.

alejandramcruzr

Luis Silvera

Data Scientist Jr.

Economista apasionado por los datos y la programación

ale-uy

Contenido

01

Introducción

Se plantea la construcción e implementación de un modelo de Machine Learning, capaz de determinar la probabilidad de omisión del record crediticio para los clientes de una importante institución financiera alemana. Buscando así reducir el riesgo de incumplimiento en el pago de sus clientes, aplicando soluciones tecnológicas de vanguardia en el análisis de sus datos.

02

Preprocesamiento de datos

- Limpieza de datos
- Manejo de datos faltantes
- Codificación de variables categóricas
- Normalización/escalado de datos

Evaluación de la data original

```
1 df_banco = pd.read_csv("german_credit.csv")
2 df_banco.head()
```

	default	account_check_status	duration_in_month	credit_history	purpose	credit_amount	savings	present_emp_since	installment_as_income_perc
0	0	< 0 DM	6	critical account/ other credits existing (not ...	domestic appliances	1169	unknown/ no savings account	.. >= 7 years	4
1	1	0 <= ... < 200 DM	48	existing credits paid back duly till now	domestic appliances	5951	... < 100 DM	1 <= ... < 4 years	2
2	0	no checking account	12	critical account/ other credits existing (not ...	(vacation - does not exist?)	2096	... < 100 DM	4 <= ... < 7 years	2
3	0	< 0 DM	42	existing credits paid back duly till now	radio/television	7882	... < 100 DM	4 <= ... < 7 years	2
4	1	< 0 DM	24	delay in paying off in the past	car (new)	4870	... < 100 DM	1 <= ... < 4 years	3

Procesamos los datos mediante mediante un map para convertir los valores de texto a numéricos

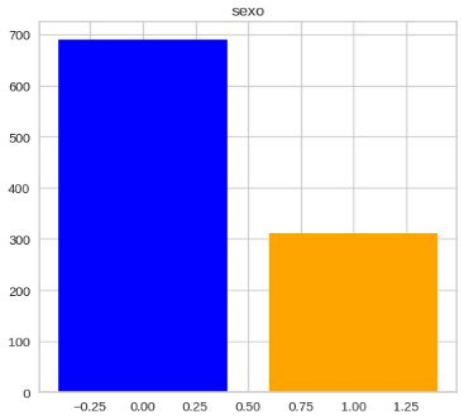
```
1 procesar_datos()
2 df_banco.head()
```

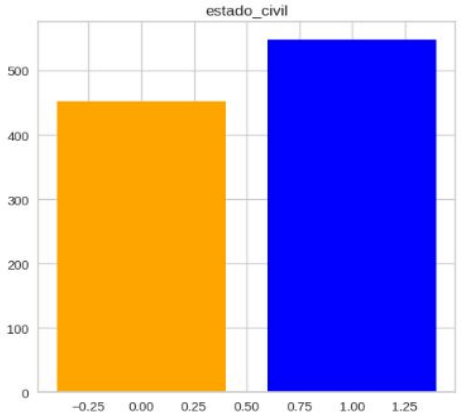
	default	account_check_status	duration_in_month	credit_history	purpose	credit_amount	savings	present_emp_since	installment_as_income_perc	personal_status_sex
0	0	1	6	5	5	1169	1	1	4	3
1	1	2	48	3	5	5951	5	3	2	2
2	0	4	12	5	8	2096	5	2	2	3
3	0	1	42	3	4	7882	5	2	2	3
4	1	1	24	4	1	4870	5	3	3	3

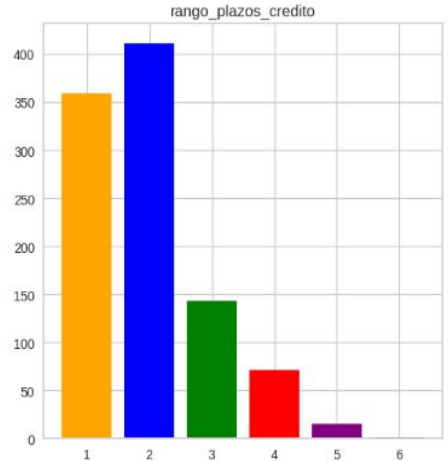
03

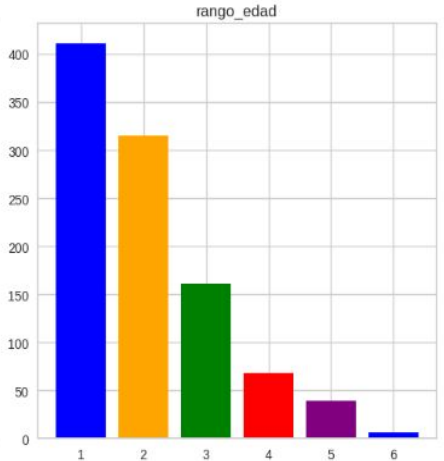
Exploración de Datos

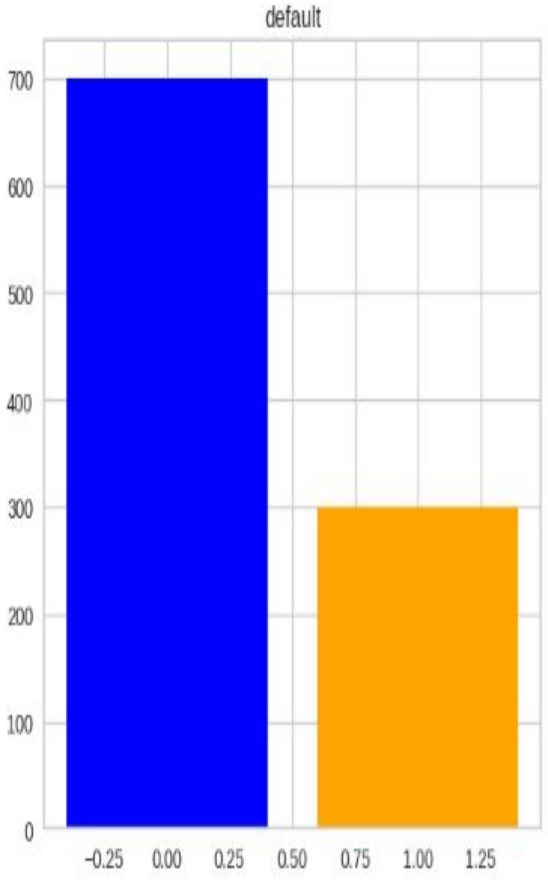
- Separamos las columnas discretas para su posterior transformación
- Realizamos la transformación de estas columnas mediante el método cut
- Graficamos las columnas categóricas de nuestro set de datos



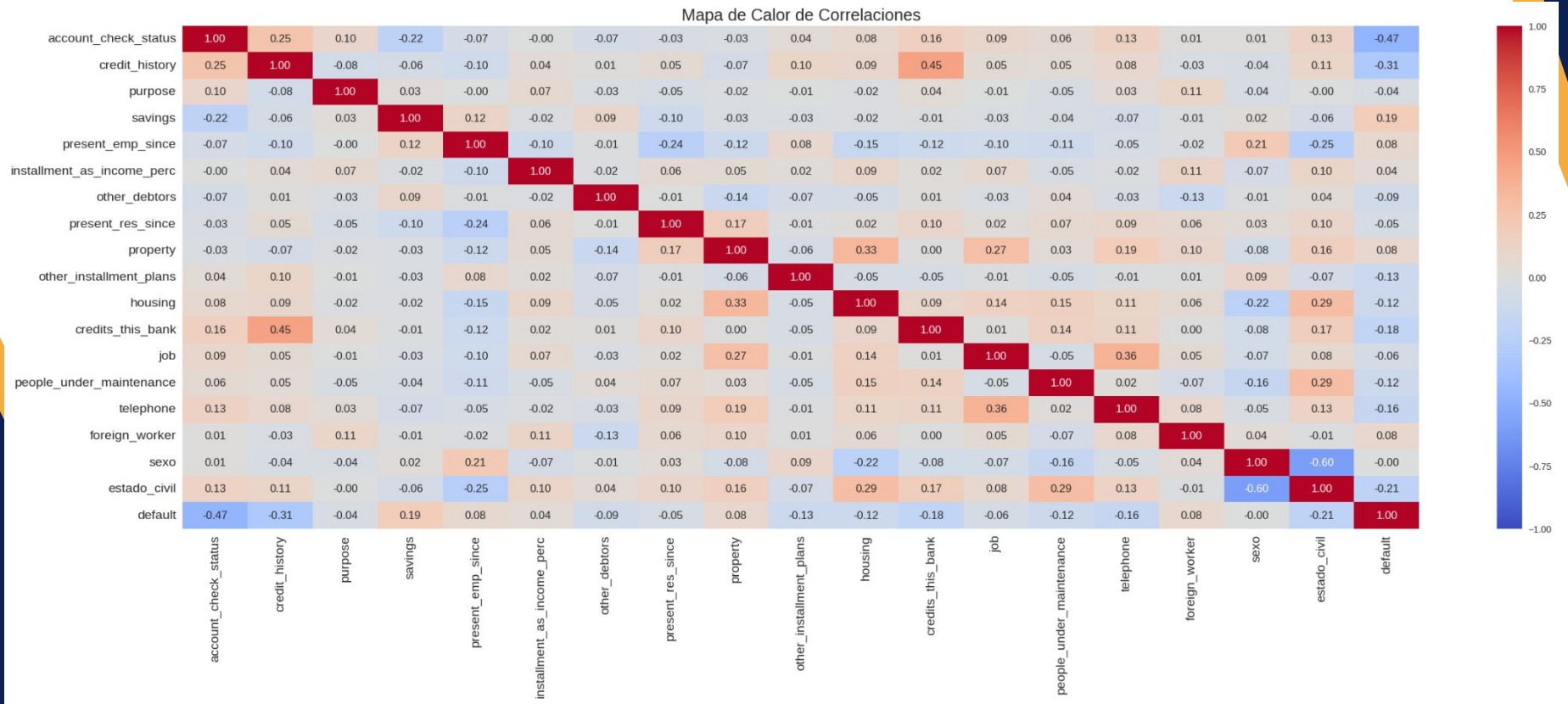








- Aplicamos la técnica de sobremuestreo en nuestros datos para balancear la columna default
- Visualizamos el mapa de calor de los datos



04

Construcción de modelos

Pycaret Es una biblioteca de Python, con PyCaret, puedes:

- Aplicar imputación de valores perdidos, escalado, ingeniería de características o selección de características de una forma muy sencilla.
- Entrenar más de 100 modelos de machine learning, de todo tipo (clasificación, regresión, pronóstico) con una sola línea de código.
- Registrar los modelos entrenados en MLFlow de una forma muy sencilla.
- Crear una API o un Docker para poner el modelo en producción.
- Subir tu modelo a la nube para poder agilizar el despliegue en producción.

"Es compatible con cualquier tipo de notebook de Python y permite realizar comparaciones de varios modelos automáticamente".

```
# Comparar todos los modelos que incluye pycaret ordenado por su valor 'AUC'
best = compare_models(sort='AUC')
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.8049	0.8904	0.8282	0.7906	0.8080	0.6099	0.6123	0.5500
et	Extra Trees Classifier	0.7869	0.8873	0.8000	0.7799	0.7880	0.5739	0.5772	0.4320
catboost	CatBoost Classifier	0.8060	0.8816	0.8282	0.7919	0.8085	0.6122	0.6151	1.5920
xgboost	Extreme Gradient Boosting	0.7946	0.8719	0.8128	0.7847	0.7978	0.5893	0.5909	0.2110
lightgbm	Light Gradient Boosting Machine	0.7933	0.8706	0.8205	0.7765	0.7974	0.5867	0.5885	0.2100
gbc	Gradient Boosting Classifier	0.7907	0.8667	0.8359	0.7660	0.7980	0.5817	0.5868	0.2430
ada	Ada Boost Classifier	0.7705	0.8381	0.7744	0.7680	0.7696	0.5411	0.5433	0.3110
knn	K Neighbors Classifier	0.7615	0.8347	0.8282	0.7315	0.7755	0.5234	0.5303	0.2420
lr	Logistic Regression	0.7654	0.8319	0.7769	0.7590	0.7664	0.5310	0.5332	1.4900
lda	Linear Discriminant Analysis	0.7667	0.8298	0.7846	0.7573	0.7695	0.5336	0.5358	0.1110
nb	Naive Bayes	0.5305	0.7392	0.1000	0.7033	0.1733	0.0571	0.1105	0.1900
dt	Decision Tree Classifier	0.7283	0.7287	0.7718	0.7111	0.7383	0.4570	0.4613	0.4160
qda	Quadratic Discriminant Analysis	0.5141	0.7138	0.1128	0.6535	0.1602	0.0244	0.0652	0.3740
dummy	Dummy Classifier	0.5025	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1030
svm	SVM - Linear Kernel	0.6964	0.0000	0.7872	0.7074	0.7153	0.3940	0.4377	0.2300
ridge	Ridge Classifier	0.7718	0.0000	0.7923	0.7613	0.7752	0.5438	0.5462	0.1740

AYUDA SOBRE MÉTRICAS EN EL CRÉDITO BANCARIO

1. Área bajo la curva ROC (AUC): Esta métrica es crucial ya que mide la capacidad del modelo para distinguir entre los clientes que cumplirán con sus obligaciones de crédito y los que no. Un AUC-ROC más alto indica un mejor rendimiento del modelo.

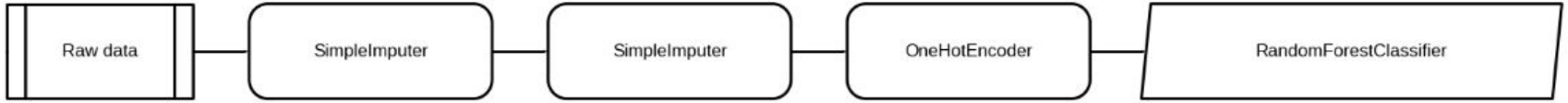
2. Exactitud (Accuracy): Esta métrica mide la proporción de predicciones correctas hechas por el modelo. En el contexto del scoring bancario, esto podría ser la proporción de clientes que el modelo predijo correctamente que pagarían o incumplirían sus obligaciones de crédito.

3. Sensibilidad (Recall): Esta métrica es importante en el scoring bancario porque mide la proporción de incumplimientos reales que el modelo es capaz de capturar.

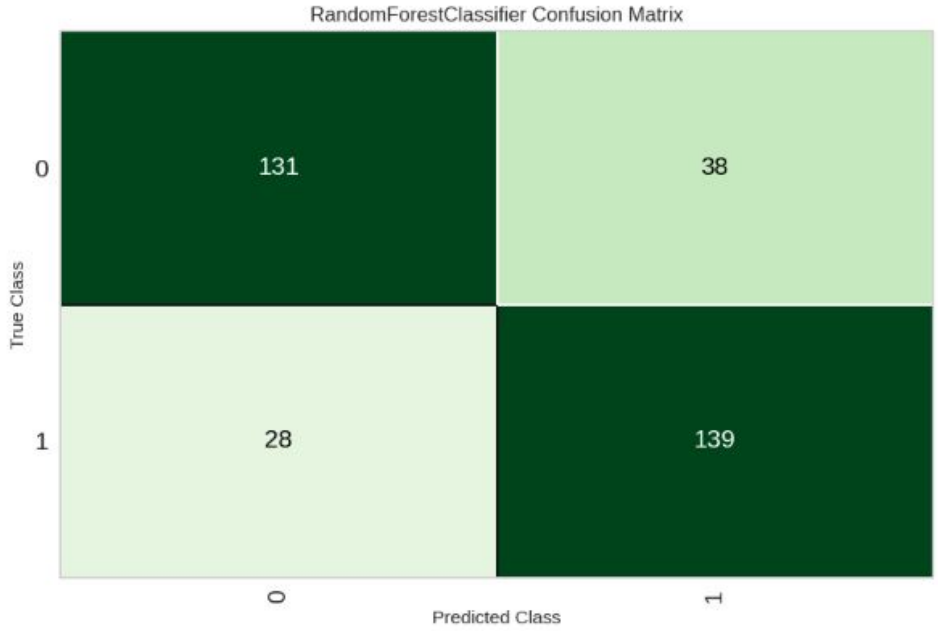
4. Valor predictivo positivo (Precision): Esta métrica mide la proporción de incumplimientos predichos que son realmente incumplimientos.

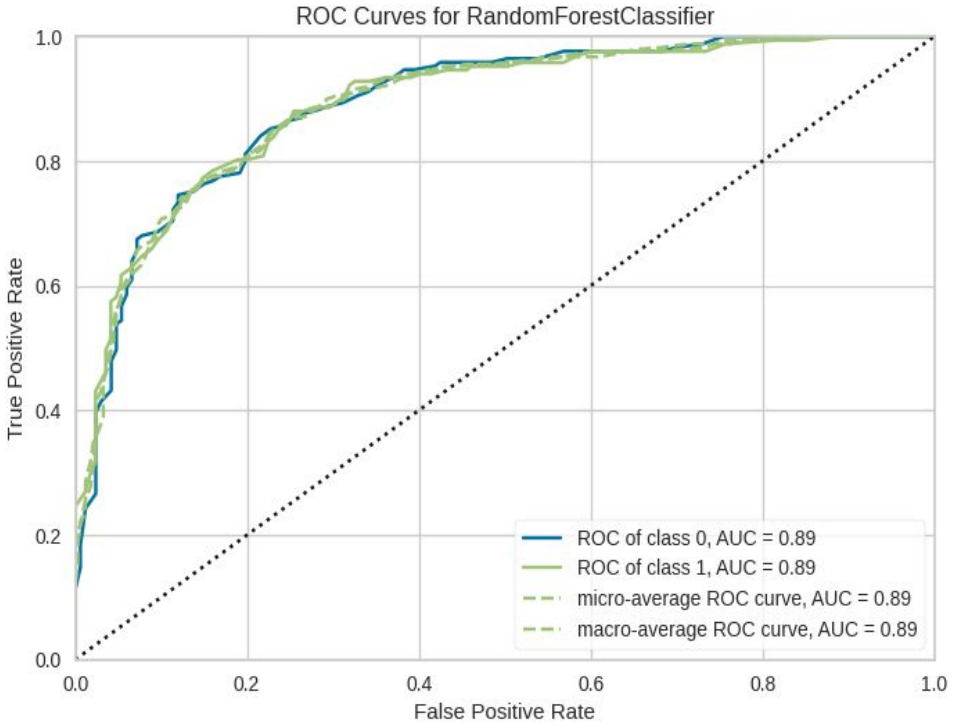
Visualizamos los resultados del entrenamiento con el mejor de los modelos entrenados

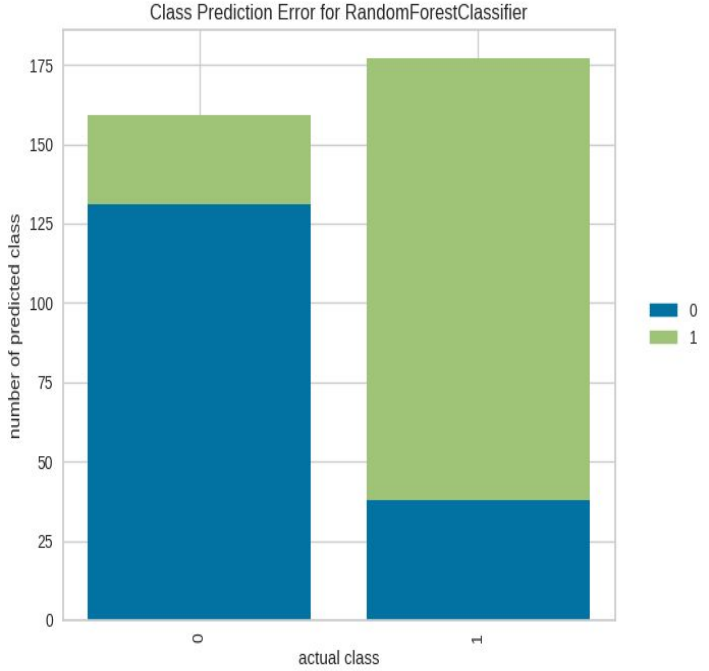
```
[ ]  1 # Visualiza las transformaciones que aplicó pycaret de forma automática
     2 plot_model(best, plot = 'pipeline')
```

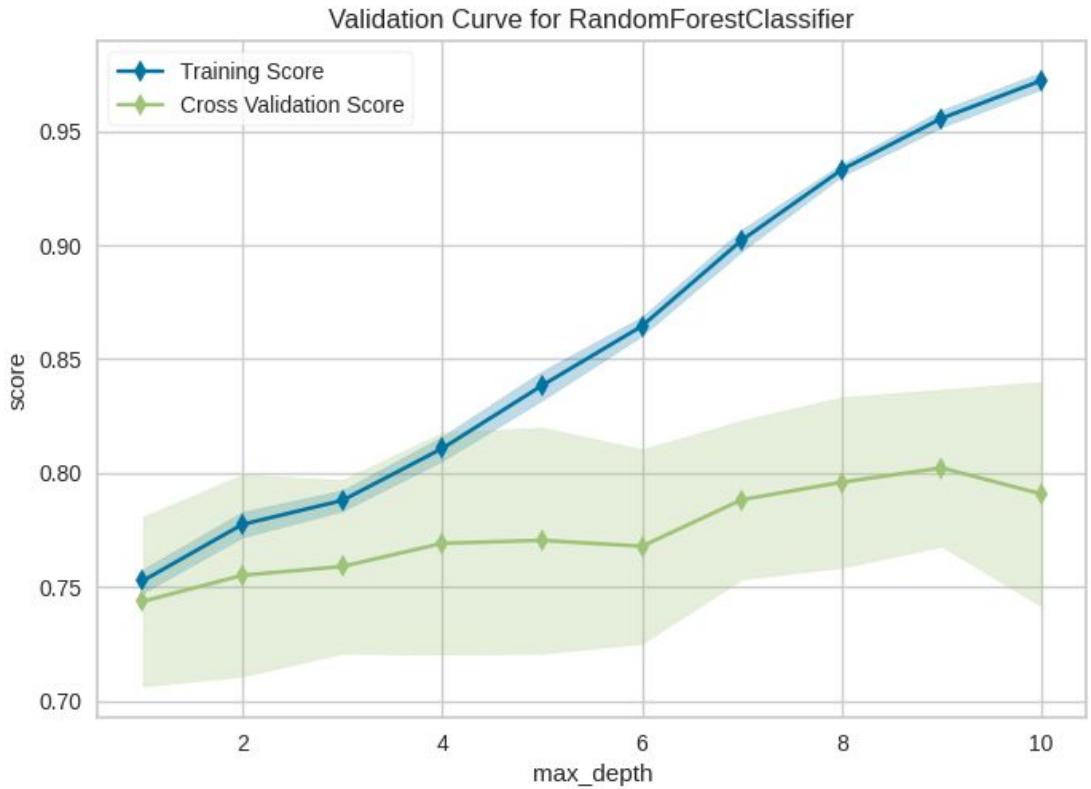



```
1 # Generamos una matriz de confusión
2 plot_model(best, plot = 'confusion_matrix')
```




ROC Curves for RandomForestClassifier
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC of class 0, AUC = 0.89
ROC of class 1, AUC = 0.89
micro-average ROC curve, AUC = 0.89
macro-average ROC curve, AUC = 0.89


Class Prediction Error for RandomForestClassifier
number of predicted class
actual class
0
25
50
75
100
125
150
175
0
1


Validation Curve for RandomForestClassifier
Training Score
Cross Validation Score
score
0.95
0.90
0.85
0.80
0.75
0.70
2
4
6
8
10
max_depth

```
# finalizar el modelo
final_best = finalize_model(best)

# guarda el modelo como archivo pickle
save_model(final_best, 'credit_score')

# Generamos las predicciones con los datos de test
predictions = predict_model(final_best, data=X_test)
predictions.head()
```

```
Transformation Pipeline and Model Successfully Saved
```

ıone	foreign_worker	sexo	estado_civil	rango_edad	rango_plazos_credito	rango_valor_credito	prediction_label	prediction_score
0	1	1	0	1	2	5	1	0.77
0	1	1	0	3	2	4	0	0.75
1	1	1	0	1	1	2	1	0.61
1	1	0	1	2	4	6	0	0.66
0	1	0	0	1	2	4	1	0.91

05

Evaluación y selección de modelos

El mejor modelo que seleccionó Pycaret es:
Clasificador Random Forest

AUC score	Accuracy score	Recall score	Precision score	F1 score
0.835	0.836	0.853	0.830	0.841

06

Conclusiones

1. Mejora de la Eficiencia Operativa: La implementación de nuestros modelos ha llevado a una mejora significativa en la eficiencia operativa al acelerar el desarrollo de scoring bancario. Esto se traduce en una reducción de los tiempos de implementación y una mayor agilidad en la toma de decisiones.

2. Adaptabilidad a Cambios en el Entorno Financiero: Tenemos una herramienta adaptable, capaz de ajustarse a cambios en el entorno financiero y en las preferencias del mercado. Esta flexibilidad es crucial para garantizar que nuestros modelos de scoring sigan siendo relevantes y efectivos a lo largo del tiempo.

3. Aumento de la Precisión en las Decisiones Crediticias: La capacidad del modelo para evaluar rápidamente el riesgo crediticio, ha contribuido a una gestión más efectiva de las carteras y una reducción de pérdidas.

4. Reducción de Errores y Pérdidas: La implementación, llevará a una reducción significativa de errores en la evaluación del riesgo crediticio, lo que se traduce directamente en una disminución de pérdidas para la institución financiera.