Data Science

Proyecto Final

Evaluación Crediticia

Comisión 41875

Profesor: Ignacio Ruso

Tutor a cargo: Juan Cruz Alric

Alumno: Juan Ignacio Rainoldi

Contenido

Objetivos

En este caso, se deberá entregar el Proyecto Final. Para lo cual se entrenarán y optimizarán diversos modelos de machine learning para resolver una problemática específica, la cual ha sido detectada en la instancia de entrega anterior. Los objetivos de este trabajo son:

1. Abstracto con motivación y audiencia: Descripción de alto nivel de lo que motiva a analizar los datos elegidos y audiencia que se podría beneficiar de este análisis.
2. Preguntas/Problema que buscamos resolver: Si bien puede haber más de una problemática a resolver, la problemática principal debe encuadrarse como un problema de clasificación o regresión.
3. Breve Análisis Exploratorio de Datos (EDA): Análisis descriptivo de los datos mediante visualizaciones y herramientas estadísticas, análisis de valores faltantes.
4. Ingeniería de atributos: Creación de nuevas variables, transformación de variables existentes (i.e normalización de variables, encoding, etc.).
5. Entrenamiento y Testeo: Entrenamiento y testeo de al menos 2 modelos distintos de Machine Learning utilizando algún método de validación cruzada.
6. Optimización: Utilizar alguna técnica de optimización de hiperparámetros (e.g gridsearch, randomizedsearch, etc.).
7. Selección de modelos: utilizar las métricas apropiadas para la selección del mejor modelo (e.g AUC, MSE, etc.).

Introducción

Contexto Empresarial: En el contexto de una institución financiera, como un banco o una compañía de préstamos, se desea automatizar y mejorar el proceso de evaluación crediticia de los solicitantes. Para lograrlo, se decide utilizar técnicas de aprendizaje automático y construir un modelo capaz de clasificar automáticamente a una persona que solicite un crédito.

Problema empresarial: Su tareas es procesar la base de datos brindada con la cual se realizará el modelo. Se deberán realizar visualizaciones y resúmenes numéricos para responder a la pregunta planteada en contexto empresarial.

Contexto analítico: Se proporciona un archivo CSV ("test.csv"), el cual contiene información detallada sobre el historial bancario de diversos clientes como ingreso mensual, número de cuentas bancarias, monto adeudado, etc. Se empleará esta base de datos para poder realizar estimaciones acerca de la idoneidad de un cliente para recibir un crédito, basándose en su historial bancario. En este dataset, los datos están etiquetados; es decir, hay una variable que dice calidad del cliente a la hora de afrontar una deuda. Por lo que, se deberá utilizar modelos de clasificación para abordar este problema de aprendizaje supervisado.

Descripción de datos

El dataset es un archivo público el cual se obtuvo de la página Kaggle (https://www.kaggle.com/datasets/parisrohan/credit-score-classification). Este dataset cuenta con 27 campos y 50000 registros.

A continuación hay una breve descripción de las variables:

- ID: Representa el número de identificación de una entrada
- Customer_ID: Representa el código único de identificación de cada cliente
- Month: Represente el mes del año
- Name: Representa el nombre del cliente
- Age: Representa la edad del cliente
- SSN: Representa el número de seguridad social de cada cliente
- Occupation: Representa la profesión del cliente
- Annual_Income: Representa el ingreso anual del cliente
- Monthly_Inhand_Salary: Representa el ingreso mensual del cliente
- Num_Bank_Accounts: Representa el número de cuentas bancarias que tiene el cliente
- Num_Credit_Card: Representa el número de tarjetas de créditos que tiene el cliente
- Interest_Rate: Representa la tasa de interés de la tarjeta de crédito
- Num_of_Loan: Representa el número de préstamos tomados del banco
- Type_of_Loan: Representa el tipo de préstamo tomado por cliente
- Delay_from_due_date: Representa el número promedio de días de retraso desde la fecha de pago
- Num_of_Delayed_Payment: Representa el número promedio de pagos demorado por cliente
- Changed_Credit_Limit: Representa el cambio porcentual en el límite de la tarjeta de crédito
- Num_Credit_Inquiries: Representa el número de consultas de tarjetas de créditos
- Credit_Mix: Representa la clasificación del cliente frente a la combinación de créditos.
- Outstanding_Debt: Representa la deuda restante por pagar (en USD)
- Credit_Utilization_Ratio: Representa el índice de la utilización de la tarjeta de crédito.
- Credit_History_Age: Representa la historia de créditos en años de la persona
- Payment_of_Min_Amount: Representa si la persona realizó el pago del monto mínimo. Donde Yes significa que ha realizado el pago mínimo, No significa que no ha realizado el pago del monto mínimo y NM significa que no hay un monto mínimo para ese periodo.
- Total_EMI_per_month: Representa los pagos mensuales del EMI (en USD)
- Amount_invested_monthly: Representa el monto mensual invertido del cliente (en USD)

- Payment_Behaviour: Representa el comportamiento de pago de los clientes
- Monthly_Balance: Representa el saldo mensual del cliente (en USD)

En principio se realizó un análisis de todas las variables. No se detectaron elementos duplicados. Se observó que muchos campos poseían valores nulos y, a su vez, muchos de estos debían poseer datos tipo numérico y eran del tipo objeto, lo que indicaba que las mismas poseían datos erróneos (Tabla 1).

Tabla 1. Resumen de variables del data frame.

```
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 27 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   ID                        50000 non-null  object
 1   Customer_ID               50000 non-null  object
 2   Month                     50000 non-null  object
 3   Name                      44985 non-null  object
 4   Age                       50000 non-null  object
 5   SSN                       50000 non-null  object
 6   Occupation                50000 non-null  object
 7   Annual_Income             50000 non-null  object
 8   Monthly_Inhand_Salary     42502 non-null  float64
 9   Num_Bank_Accounts         50000 non-null  int64
 10  Num_Credit_Card           50000 non-null  int64
 11  Interest_Rate             50000 non-null  int64
 12  Num_of_Loan               50000 non-null  object
 13  Type_of_Loan              44296 non-null  object
 14  Delay_from_due_date       50000 non-null  int64
 15  Num_of_Delayed_Payment    46502 non-null  object
 16  Changed_Credit_Limit      50000 non-null  object
 17  Num_Credit_Inquiries      48965 non-null  float64
 18  Credit_Mix                50000 non-null  object
 19  Outstanding_Debt          50000 non-null  object
 20  Credit_Utilization_Ratio  50000 non-null  float64
 21  Credit_History_Age        45530 non-null  object
 22  Payment_of_Min_Amount     50000 non-null  object
 23  Total_EMI_per_month       50000 non-null  float64
 24  Amount_invested_monthly   47729 non-null  object
 25  Payment_Behaviour         50000 non-null  object
 26  Monthly_Balance           49438 non-null  object
dtypes: float64(4), int64(4), object(19)
```

Data Wrangling

En el primer relevamiento de las variables, nos encontramos con algunos campos que no aportan información adicional por lo que se los eliminara en una primera instancia. Las variables excluidas son:

- ID: Es la identificación del registro.
- Month: Es el mes del registro.
- Name: Es el nombre del cliente.
- SSN: Es el número de seguridad del cliente.

Del total de los 27 campos, se eliminaron los cuatro campos mencionados, pasando a tener 23 campos. De los 23 campos se poseen se poseen 15 del tipo categórico y 8 del tipo cuantitativo.

Se identifican que campos deben ser categóricos y cuales deben de ser numéricos, encontrando que hay varios campos numéricos que están en formato texto.

Se procede a buscar todos los caracteres erróneos en los campos y se los elimina, luego se convierten el tipo de dato de cada columna al que corresponde.

Luego se procede a buscar los out layers, una vez identificados, se elimina dichos valores y se procede a completar dichos valores.

- En el caso de 'Credit_Mix' al ser el dato etiqueta se procede a eliminar la fila completa.
- En el caso de 'Monthly_Inhand_Salary','Total_EMI_per_month', 'Historial_crediticio_meses','Num_of_Delayed_Payment','Interest_Rate',' Num_of_Loan' y 'Num_Credit_Card' se procede a rellenarlo haciendo una interpolación lineal.
- Para el caso de los datos restantes se completaran con el valor más cercano, incluyendo a todos los campos que se les aplico interpolación lineal, esto se hace para que complete los valores que habían quedado sin completar debido a que la interpolación lineal no completa todos los valores.

Se procede a crear un nuevo campo por cada tipo de préstamo que se encuentran en el campo 'Type_of_Loan', obteniéndose un total de 10 campos nuevos, estos campos son de datos tipo booleano, luego se crea el campo 'Diferent_Loans', este campo cuantifica la cantidad de los diferentes tipos de préstamos que toma cada cliente. También se crea el campo 'Historial_crediticio_meses' el cual la normalización del campo 'Credit_History_Age'. Se eliminan los campos 'Type_of_Loan' y 'Credit_History_Age'.

Se aplica Hot Encoder para los campos 'Payment_Behaviour' y 'Payment_of_Min_Amount', a los campos 'Occupation' y 'Credit_Mix' se les aplica Label Encoder. De esta forma se obtiene un total de 11 campos nuevos.

Al aparecer cada cliente 4 veces en el data frame se procede a eliminar los registros dejando solo uno por cada cliente, conservando el último registro de cada cliente.

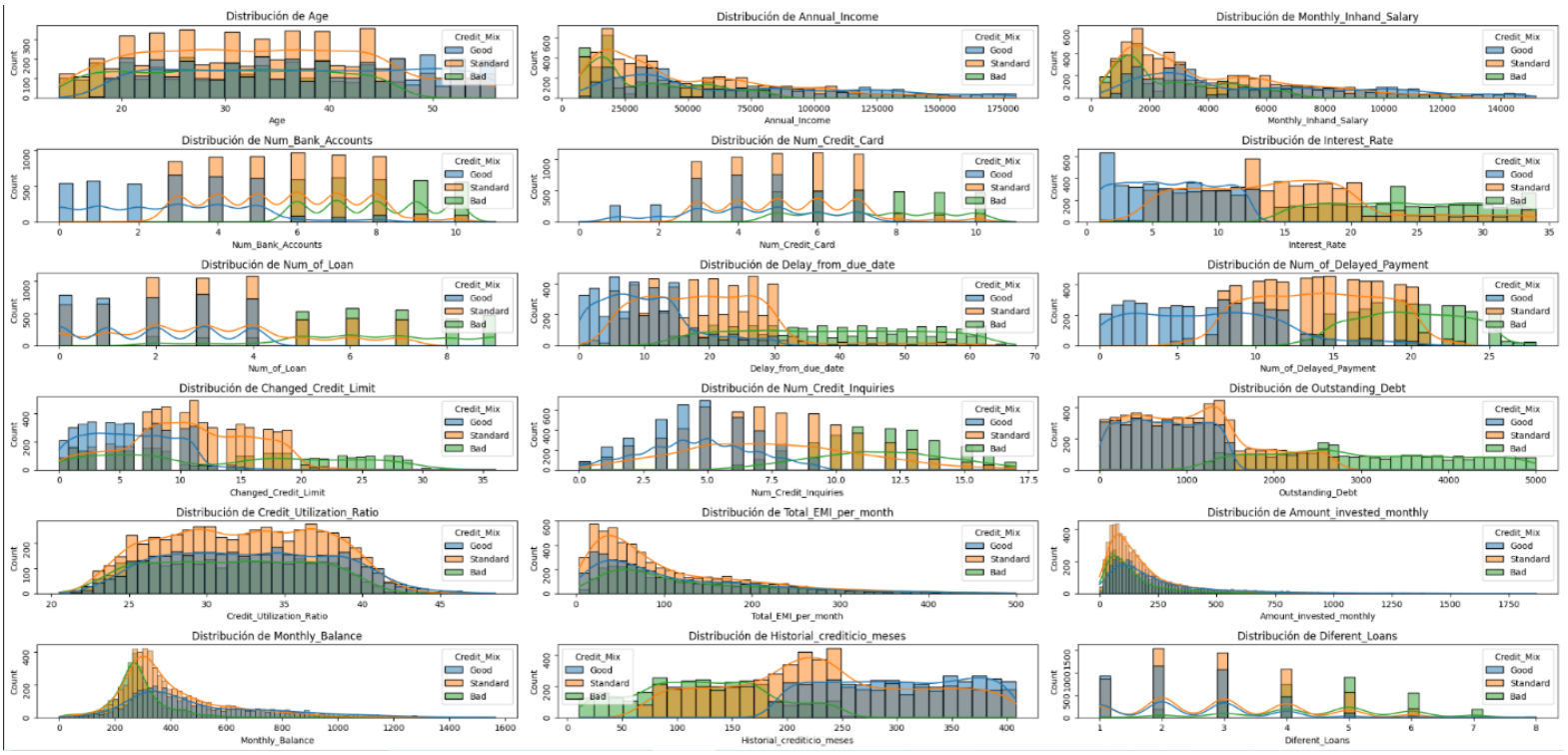
Una vez terminado el Data Wrangling el dataframe queda compuesto por 43 campos y 12479 registros.

EDA (Exploratory Data Analysis)

En un primer momento se debió enfocar en realizar una limpieza y corrección de los datos, ya que las variables numéricas se encontraban notablemente afectadas por información incorrecta. Una vez completada esta etapa de data wrangling, se procedió a examinar las distribuciones de las variables categóricas y numéricas, lo que permitió una exploración más precisa y significativa de los datos.

Se realizó un histograma de las variables numéricas donde se puede observar las distribuciones de las variables en los distintos campos en función de las etiquetas que poseen los clientes. Se destaca que en la distribución de los etiquetados como malos clientes en los campos de ´Num_of_Credit_Inquiries´, ´Interest_Rate´ y ´Num_of_Delay_Payment´ poseen una distribución asimétrica negativa, mientras que los que están etiquetados como buenos clientes poseen una distribución asimétrica positiva. En el caso de ´Historial_Crediticio_mese´ los clientes que están etiquetados como malos clientes poseen una distribución asimétrica positiva, mientras los que están etiquetados como buenos clientes poseen una distribución asimétrica negativa.

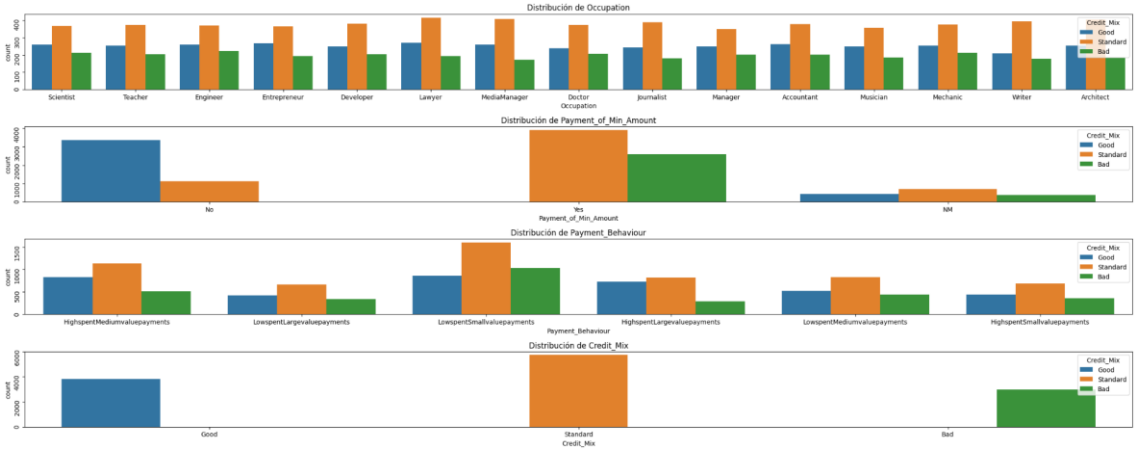
Grafico 1: Histograma de las variables más relevantes.



Se realizó un gráfico de barras para ver la distribución de las variables categóricas más relevante, en estos gráficos, se puede observar que la distribución de datos etiqueta ('Credit_Mix') no poseen una distribución desbalanceada de modo tal que vaya a afectar el entrenamiento del modelo. También se puede observar que los que se consideran buenos clientes no realizaron pagos de las cantidades mínimas, mientras que los clasificados como malos clientes tienden solo a realizar pagos de las cantidades mínimas. Se

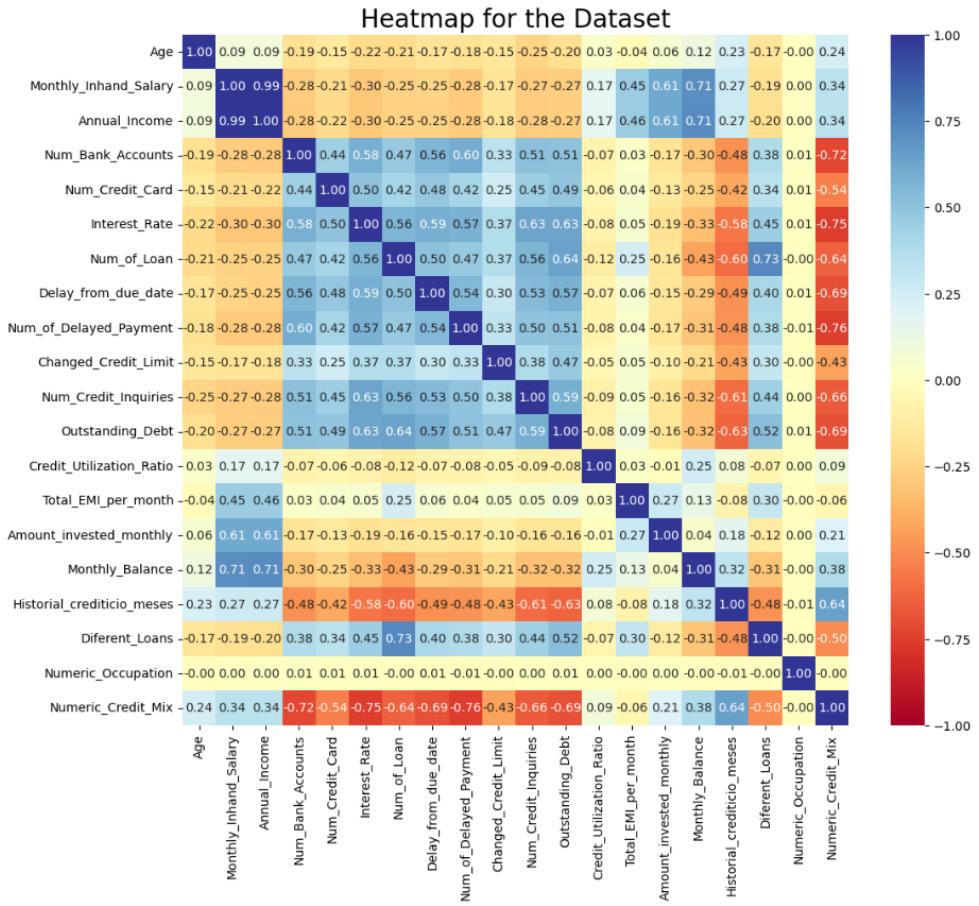
observa que los clientes que no poseen un monto mínimo establecido de pago son un grupo significativamente menor con respecto a aquellos que si lo poseen.

Gráfico 2: gráfico de barras de los campos categóricos.



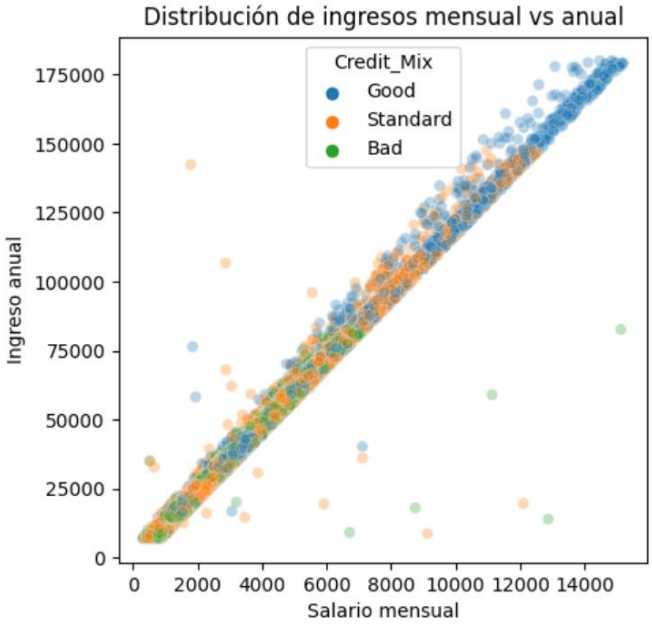
Se procedió a realizar un análisis bivariado para intentar determinar si hay alguna correlación del tipo lineal entre alguna de las variables numéricas. Se observa que los únicos campos con una correlación alta son el ingreso mensual y el ingreso anual. Se observa que la variable 'Numeric_Occupation' posee una correlación igual a 0 con todas las demás variables.

Gráfico 3: Gráfico de correlación entre las variables cuantitativas.



Se ha generado un gráfico de dispersión entre el ingreso mensual y el ingreso anual, y se ha observado una correlación lineal prácticamente perfecta entre ambas variables. Esta fuerte correlación indica que trabajar con ambas variables es redundante. Por lo tanto, se concluye que la contribución del campo 'Annual_Income' es insignificante y, en consecuencia, prescindible.

Gráfico 4: Gráfico de dispersión entre el ingreso anual y el ingreso mensual.



Luego del análisis univariado y bivariado se eliminaran los campos que se consideran que no aportan información relevante. Los cuales son:

- 'Customer_ID', 'Annual_Income' y 'Occupation' ya que no han demostrado proporcionar información significativa, como se evidenció en el análisis bivariado y univariado.
- En el caso de 'Numeric_Occupation' y 'Numeric_Credit_Mix' solo se empleó para realizar el análisis bivariado , por lo que no son necesarios en el modelo.
- 'Payment_Behaviour' y 'Payment_of_Min_Amount' a estas se les aplico one hot encoder o level encoder por lo que no son necesarias.

Una vez terminado el EDA el dataframe queda compuesto por 36 campos y 12479 registros.

Entrenamiento de Machine Learning

Se realizaran modelos de Machine Learning para darle solución al problema planteado, el cual consiste en clasificar a los clientes a la hora de pedir un préstamo. Al ser un problema de clasificación se realizarán modelos supervisados, los modelos que se entrenaran son:

- Random Fores (max_depth=5)
- Tree Descision (max_depth=5)
- KKN (n_neighbors=5)
- SVC (C=0.1, kernel=´Sigmoid´)
- xgboost (objective='multi:softmax', n_estimators=10, learning_rate=0.01, seed=42, max_depth=6).

A cada modelo se le aplicará la validación cruzada K-fold en 6 iteraciones. A la hora de evaluar el modelo se contemplaran en la métrica Recall y en segundo lugar la métrica Accuracy y el desvío Standard de las mismas. Se obtiene que el modelo que mejor responde al problema es el Random Forest con un Recall del 92,84% (0,59%) y un Accuracy de 92,07% (0,68%).

Tabla 2: Métricas de los modelos entrenados con el total de los datos.

Modelos:	Recall:	STD_Recall:	Accuracy:	STD_Accuracy:	Campos:
RF_full	92.84	0.59	92.07	0.68	35.0
xgb_full	91.8	0.49	91.59	0.71	35.0
TD_full	89.75	0.51	89.59	0.53	35.0
KNN_full	65.16	0.69	63.96	0.7	35.0
SVM_full	44.54	1.18	46.54	1.34	35.0

Se aplica importan feature en los modelos Random Forest, Tree Decision y eXtreme Gradient Boostingpara y se observa que los campos que no aportan información muy relevante para el modelo son:

'Payment_Behaviour_LowspentLargevaluepayments',
'Payment_Behaviour_HighspentMediumvaluepayments',
'Payment_Behaviour_HighspentSmallvaluepayments',
'Payment_Behaviour_LowspentMediumvaluepayments',
'Payment_Behaviour_LowspentSmallvaluepayments',
'Payment_Behaviour_HighspentLargevaluepayments',
'Debt_Consolidation_Loan', 'Student_Loan', 'Not_Specified', 'Credit-Builder_Loan', 'Mortgage_Loan', 'Auto_Loan', 'Personal_Loan', 'Payday_Loan', 'Total_EMI_per_month', 'Home_Equity_Loan', 'Amount_invested_monthly', 'Age', 'Credit_Utilization_Ratio', 'Monthly_Balance', 'Monthly_Inhand_Salary', 'Payment_of_Min_Amount_NM'

Se procederá a entrenar nuevamente los mismos modelos bajo condiciones idénticas, con la particularidad de que en esta ocasión se reducirá el Data Frame eliminando aquellos campos que, de acuerdo con el "Feature Importance," no aportan información relevante. Se trabajara con un data frame de entrenamiento y testeo reducido.

Se observa que una vez más, el modelo Random Forest demuestra ser el más eficaz para resolver el problema, con un Recall del 93.43% (0.48%) y un Accuracy del 92.7% (0.55%).

Se observa que los valores en las métricas son muy similares que cuando se trabajó con el data frame completo. Esto indica que los campos eliminados no aportaban información relevante. Se procede a eliminar de forma definitiva estos campos en el Data Frame, que pasa de tener 36 campos y 12,479 registros a contar con 12 campos y 12,479 registros.

Tabla 3: Métricas de los modelos entrenados con data frame reducido.

Modelos:	Recall:	STD_Recall:	Accuracy:	STD_Accuracy:	Campos:
RF_red	93.5	0.45	93.14	0.44	11.0
xgb_red	91.92	0.5	91.71	0.79	11.0
TD_red	89.83	0.46	89.66	0.59	11.0
KNN_red	82.2	0.56	81.3	0.68	11.0
SVM_red	24.53	0.2	23.39	0.43	11.0

Se procederá a entrenar nuevamente los mismos modelos bajo las mismas condiciones, pero en esta ocasión se aplicara PCA al data freme empleado recientemente. Se observa que el KKN (K Nearest Neighbors) es el modelo que mejor responde al problema planteado con un Recall de 91,71% (0,44%) y un Accuracy de 91,31% (0,49).

Tabla 4: Métricas de los modelos entrenados con PCA.

Modelos:	Recall:	STD_Recall:	Accuracy:	STD_Accuracy:	Campos:
KNN_PCA	92.14	0.38	91.79	0.39	8.0
RF_PCA	91.21	0.63	90.74	0.76	8.0
xgb_PCA	90.85	0.86	90.66	0.78	8.0
TD_PCA	89.63	1.18	89.66	0.95	8.0
SVM_PCA	82.39	0.84	81.99	0.91	8.0

El modelo que se considera óptimo según los resultados de las métricas obtenidas, es el Random Forest con el Data Frame reducido.

Optimización de hiperparámetros

Se llevará a cabo una optimización de hiperparámetros utilizando el método Grid Search, con un énfasis en el Recall como métrica principal. Este proceso se aplicará al modelo seleccionado, el cual posee las siguientes características de hiperparámetros:

- 'n_estimators': [50, 100, 150],
- 'criterion': ['gini', 'entropy', 'log_loss'],
- 'max_features': ['sqrt', 'log2', None],
- 'max_depth': [10, 5, 3],
- 'random_state': [42]

Luego de aplicar Greed Search se obtiene que el modelo con la métrica más alta en la que se centré el modelo es un Recall de 94,47%, los hiperparámetros obtenidos del proceso fueron los siguientes:

- 'criterion': 'entropy',
- 'max_depth': 10,
- 'max_features': 'sqrt',
- 'n_estimators': 100,
- 'random_state': 42

Métricas finales del modelo optimizado

El modelo considerado óptimo según los resultados de las métricas obtenidas, fue la Random Forest, para la cual se obtuvo.

Tabla 5: Métricas del modelo elegido.

	precision	recall	f1-score	support
Bad	0.93	0.97	0.95	886
Good	0.93	0.93	0.93	1111
Standard	0.94	0.92	0.93	1747
accuracy			0.93	3744
macro avg	0.93	0.94	0.94	3744
weighted avg	0.93	0.93	0.93	3744

Se puede observar que todas las métricas muestran valores por encima del 90%, lo que sugiere que el modelo Random Forest (con los hiperparámetros especificados: criterion: 'entropy', max_depth: 10, max_features: 'sqrt', n_estimators: 150, random_state: 42) es una elección altamente efectiva para abordar el problema en cuestión.

Conclusión

Se ha logrado desarrollar un modelo sólido y efectivo, el cual es capaz de predecir con alta precisión si un cliente será catalogado como "bueno," "estándar" o "malo" al solicitar un préstamo.

Los resultados obtenidos respaldan la elección de este modelo, ya que ha demostrado métricas consistentemente superiores al 90%. Esta consistencia confirma su aptitud para tomar decisiones precisas en el contexto de la concesión de préstamos a clientes.

Por lo tanto, se ha logrado obtener un modelo confiable y efectivo que contribuirá positivamente a la toma de decisiones en el ámbito de la clasificación de clientes en categorías de préstamos.