FACULTAD DE INGENIERÍA

INGENIERÍA CIVIL INFORMÁTICA

Ciencia de Datos

Análisis y Preprocesamiento

https://github.com/JavierTallarin/proyecto-CreditoBancario-Ciencia-De-Datos

Jorge Ahumada Margarit

Javier Bravo Orellana

Cristóbal Olave Herrera

Luis Rodriguez Zamora

Índice

Introducción

Este conjunto de datos de UCI contiene cantidad crediticia, datos demográficos, historial de pagos y extractos de facturas de clientes de tarjetas de crédito en Taiwán desde abril de 2005 hasta septiembre de 2005. Finalmente esta investigación tiene como objetivo el caso de los pagos por incumplimiento de los clientes en Taiwán.

Objetivos

- Realizar estadística descriptiva básica para conocer los datos que se encuentran en el dataset y cuáles son sus características.
- Reconocer patrones presentes en el dataset que nos servirán para clasificar posteriormente con la ayuda de modelos de predicción y clasificadores probabilísticos.
- Finalmente la finalidad de este trabajo investigativo es estimar la probabilidad de incumplimiento de los clientes de Taiwán, de esta manera se podrá decidir si se le asigna el crédito a algún cliente del banco.

Descripción de los datos

El dataset default of credit card clients.csv contiene 24 variables, entre ellas datos demográficos e historial crediticio.
Las variables incluidas en el dataset son:

- ID: ID de cada cliente, variable numérica
- LIMIT_BAL: cantidad de crédito otorgado en dólares de taiwán (variable númerica)
- SEX: Sexo, variable categórica (1 masculino, 2 femenino)
- EDUCATION: Nivel máximo educacional, variable categórica (1 postgrado, 2 universidad, 3 bachillerato, 4 otros)

- MARRIAGE: Estado civil, variable categórica (1 casado, 2 soltero, 3 otros)
- AGE: edad, variable numérica
- PAY_X: Historial de pagos pasados, variable numérica
- BILL_AMTX: Monto del estado de la cuenta en dólares de taiwán, variable numérica
- PAY_AMTX: Monto del pago anterior en dólares de taiwán, variable numérica

Imagen 1.
En esta imagen se muestran los 10 primeros datos del dataset.
En este caso utilizamos la matriz traspuesta para visualizar
de mejor forma los datos.

	0	1	2	3	4	5	6	7	8	9
ID	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
LIMIT_BAL	20000.0	120000.0	90000.0	50000.0	50000.0	50000.0	500000.0	100000.0	140000.0	20000.0
SEX	2.0	2.0	2.0	2.0	1.0	1.0	1.0	2.0	2.0	1.0
EDUCATION	2.0	2.0	2.0	2.0	2.0	1.0	1.0	2.0	3.0	3.0
MARRIAGE	1.0	2.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0	2.0
AGE	24.0	26.0	34.0	37.0	57.0	37.0	29.0	23.0	28.0	35.0
PAY_1	2.0	-1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	-2.0
PAY_2	2.0	2.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	-2.0
PAY_3	-1.0	0.0	0.0	0.0	-1.0	0.0	0.0	-1.0	2.0	-2.0
PAY_4	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.0
PAY_5	-2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
PAY_6	-2.0	2.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0
BILL_AMT1	3913.0	2682.0	29239.0	46990.0	8617.0	64400.0	367965.0	11876.0	11285.0	0.0
BILL_AMT2	3102.0	1725.0	14027.0	48233.0	5670.0	57069.0	412023.0	380.0	14096.0	0.0
BILL_AMT3	689.0	2682.0	13559.0	49291.0	35835.0	57608.0	445007.0	601.0	12108.0	0.0
BILL_AMT4	0.0	3272.0	14331.0	28314.0	20940.0	19394.0	542653.0	221.0	12211.0	0.0
BILL_AMT5	0.0	3455.0	14948.0	28959.0	19146.0	19619.0	483003.0	-159.0	11793.0	13007.0
BILL_AMT6	0.0	3261.0	15549.0	29547.0	19131.0	20024.0	473944.0	567.0	3719.0	13912.0
PAY_AMT1	0.0	0.0	1518.0	2000.0	2000.0	2500.0	55000.0	380.0	3329.0	0.0
PAY_AMT2	689.0	1000.0	1500.0	2019.0	36681.0	1815.0	40000.0	601.0	0.0	0.0
PAY_AMT3	0.0	1000.0	1000.0	1200.0	10000.0	657.0	38000.0	0.0	432.0	0.0
PAY_AMT4	0.0	1000.0	1000.0	1100.0	9000.0	1000.0	20239.0	581.0	1000.0	13007.0
PAY_AMT5	0.0	0.0	1000.0	1069.0	689.0	1000.0	13750.0	1687.0	1000.0	1122.0
PAY_AMT6	0.0	2000.0	5000.0	1000.0	679.0	800.0	13770.0	1542.0	1000.0	0.0
default	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Análisis 1D y 2D de datos

1D

Imagen 2.

En esta segunda imagen generamos una tabla con las estadísticas unidimensionales, entre las cuales se encuentra:

- La media
- La desviación estándar
- El valor mínimo de cada columna
- El cuartil q1(25%), q2(50%), q3(75%)
- El valor máximo de cada columna

Con estos datos se puede armar una idea de qué tan dispersos están los datos.

	count	mean	std	min	25%	50%	75%	max
ID	30000.0	15000.500000	8660.398374	1.0	7500.75	15000.5	22500.25	30000.0
LIMIT_BAL	30000.0	167484.322667	129747.661567	10000.0	50000.00	140000.0	240000.00	1000000.0
SEX	30000.0	1.603733	0.489129	1.0	1.00	2.0	2.00	2.0
EDUCATION	30000.0	1.853133	0.790349	0.0	1.00	2.0	2.00	6.0
MARRIAGE	30000.0	1.551867	0.521970	0.0	1.00	2.0	2.00	3.0
AGE	30000.0	35.485500	9.217904	21.0	28.00	34.0	41.00	79.0
PAY_1	30000.0	-0.016700	1.123802	-2.0	-1.00	0.0	0.00	8.0
PAY_2	30000.0	-0.133767	1.197186	-2.0	-1.00	0.0	0.00	8.0
PAY_3	30000.0	-0.166200	1.196868	-2.0	-1.00	0.0	0.00	8.0
PAY_4	30000.0	-0.220667	1.169139	-2.0	-1.00	0.0	0.00	8.0
PAY_5	30000.0	-0.266200	1.133187	-2.0	-1.00	0.0	0.00	8.0
PAY_6	30000.0	-0.291100	1.149988	-2.0	-1.00	0.0	0.00	8.0
BILL_AMT1	30000.0	51223.330900	73635.860576	-165580.0	3558.75	22381.5	67091.00	964511.0
BILL_AMT2	30000.0	49179.075167	71173.768783	-69777.0	2984.75	21200.0	64006.25	983931.0
BILL_AMT3	30000.0	47013.154800	69349.387427	-157264.0	2666.25	20088.5	60164.75	1664089.0
BILL_AMT4	30000.0	43262.948967	64332.856134	-170000.0	2326.75	19052.0	54506.00	891586.0
BILL_AMT5	30000.0	40311.400967	60797.155770	-81334.0	1763.00	18104.5	50190.50	927171.0
BILL_AMT6	30000.0	38871.760400	59554.107537	-339603.0	1256.00	17071.0	49198.25	961664.0
PAY_AMT1	30000.0	5663.580500	16563.280354	0.0	1000.00	2100.0	5006.00	873552.0
PAY_AMT2	30000.0	5921.163500	23040.870402	0.0	833.00	2009.0	5000.00	1684259.0
PAY_AMT3	30000.0	5225.681500	17606.961470	0.0	390.00	1800.0	4505.00	896040.0
PAY_AMT4	30000.0	4826.076867	15666.159744	0.0	296.00	1500.0	4013.25	621000.0
PAY_AMT5	30000.0	4799.387633	15278.305679	0.0	252.50	1500.0	4031.50	426529.0
PAY_AMT6	30000.0	5215.502567	17777.465775	0.0	117.75	1500.0	4000.00	528666.0
default	30000.0	0.221200	0.415062	0.0	0.00	0.0	0.00	1.0

Imagen 3.

En esta imagen se hizo un gráfico de barras para visualizar el nivel de educación máximo que alcanzaron las personas del banco en Taiwán. De esta manera podremos identificar el nivel de educación que más predomina en el dataset.

Para este caso aproximadamente 14.000 personas tienen como nivel educacional máximo la universidad, por lo que la mayoría de las personas solamente hizo el pregrado.

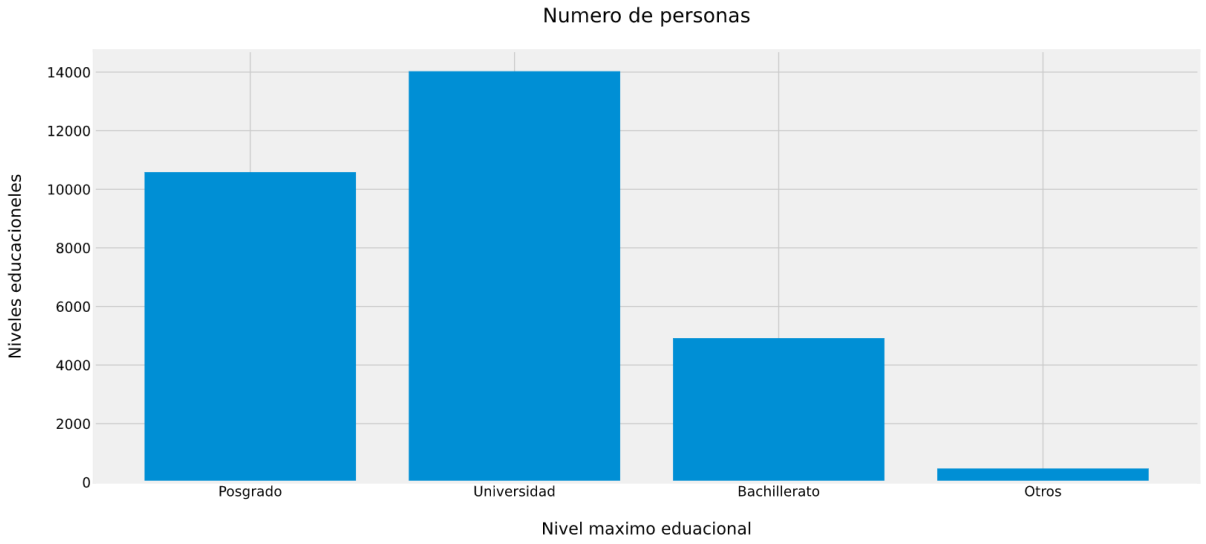


Imagen 4.

En esta imagen hicimos un gráfico de barras para visualizar el estado civil de las personas del banco en Taiwán.

Como resultado se obtuvo aproximadamente 16.000 personas solteras, por lo tanto podemos concluir que hay una cantidad considerable de personas solteras dentro del dataset.

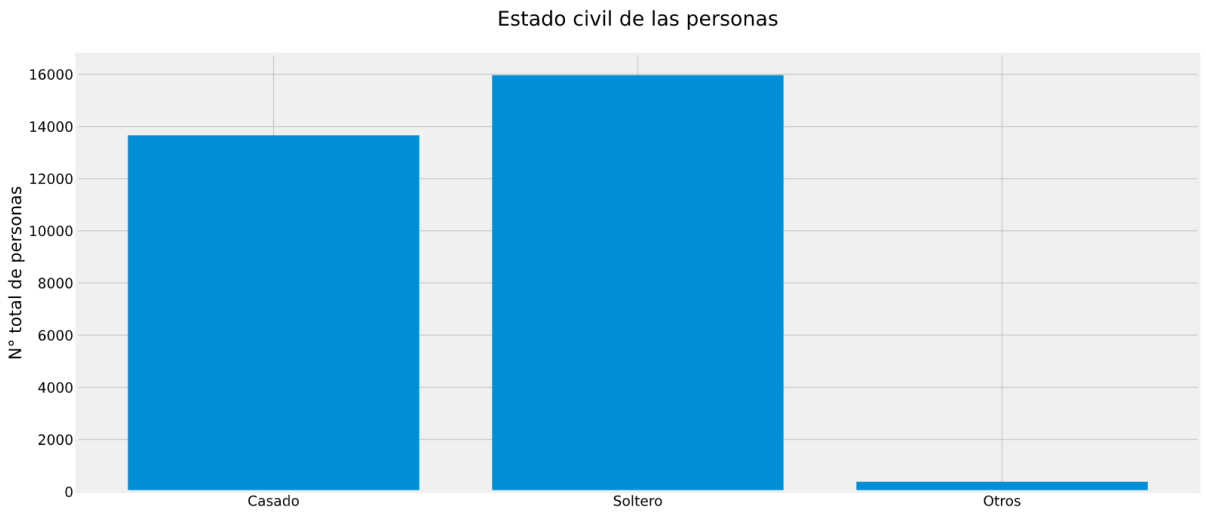


Imagen 5.

En esta imagen hicimos un gráfico de barras para visualizar cuál es el sexo que mas predomina el el dataset.

Claramente se visualiza que la mayoría de las personas en el dataset pertenecen al sexo femenino con mas de 17.500 personas.

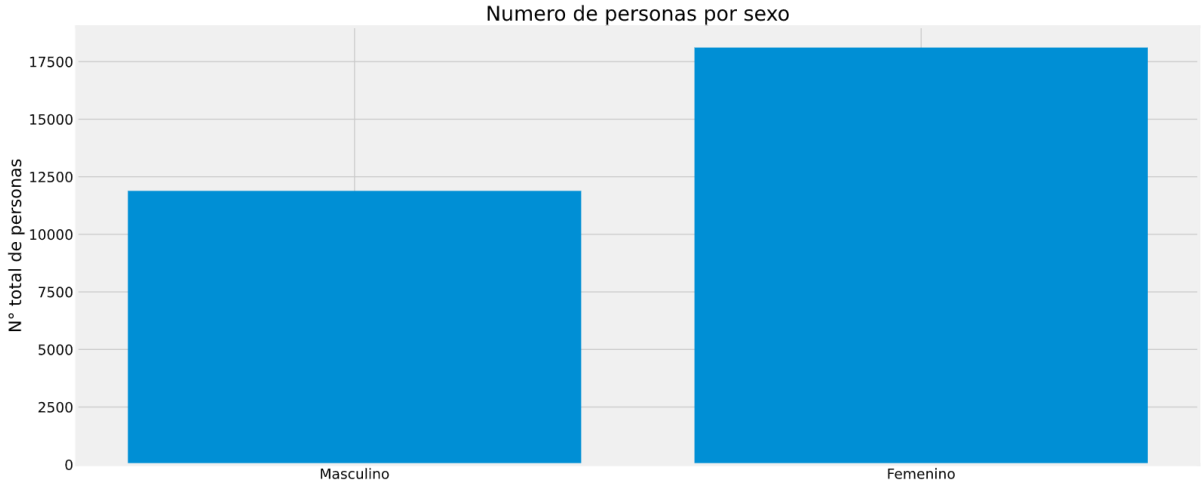
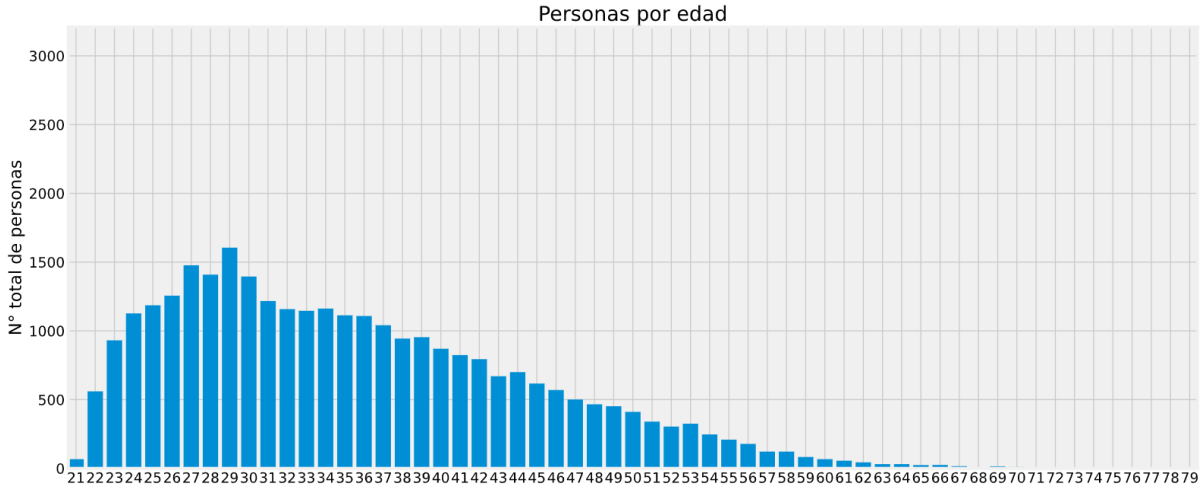


Imagen 6.

En esta imagen hicimos un gráfico para visualizar la distribución de las edades en el dataset.

Si hiciéramos una agrupación de edades en un rango de 5 años. tendríamos que la mayoría de los clientes del banco de Taiwán están entre los 25 y 30 años de edad, incluso la mediana es de 34 años.



2D

Imagen 7.

En esta imagen hicimos un gráfico Heatmap para visualizar la correlación entre las variables del dataset.

El objetivo es encontrar las variables que tengan relación lineal entre ellas o aquellas que tengan una relación lineal inversa.

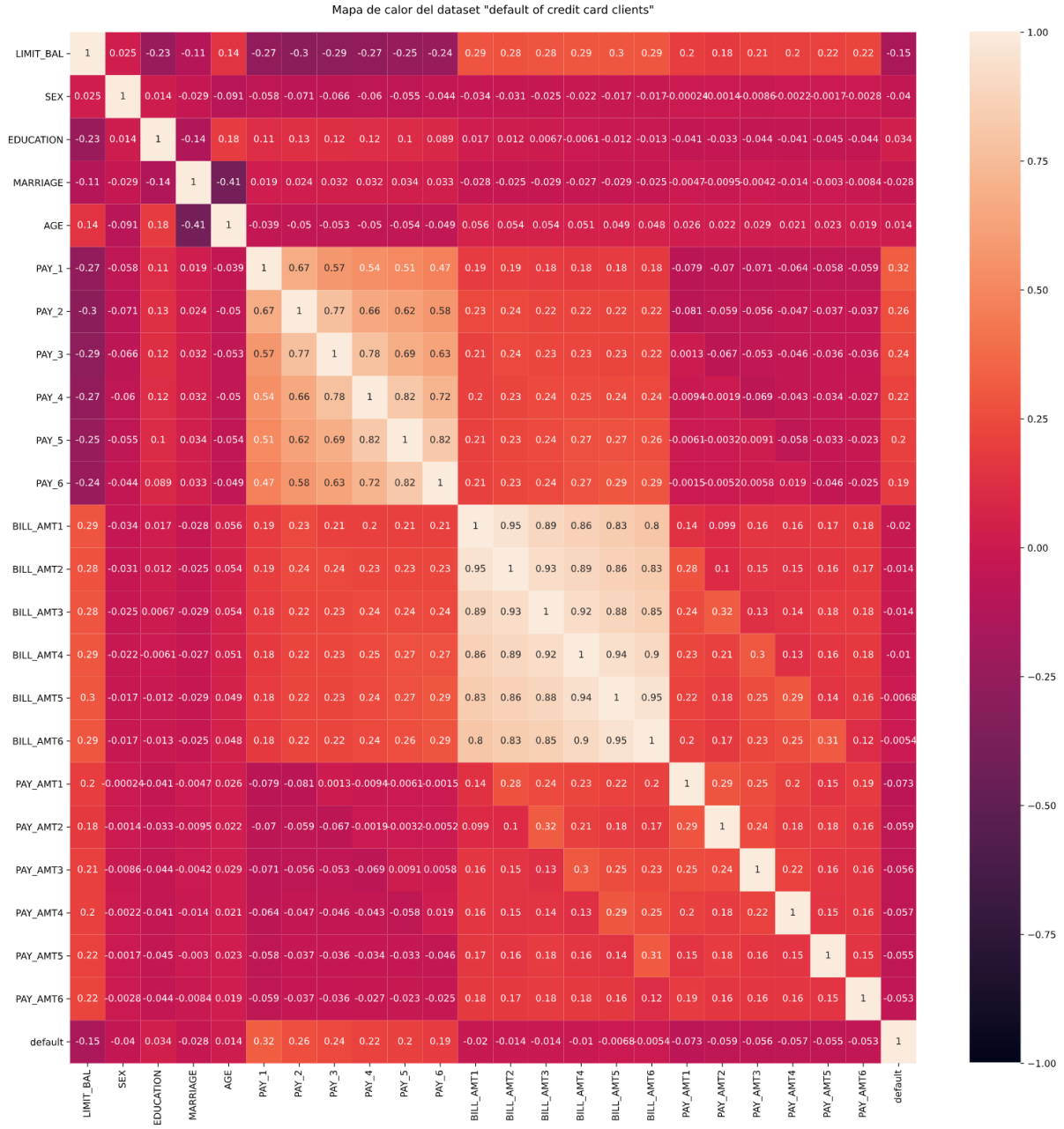
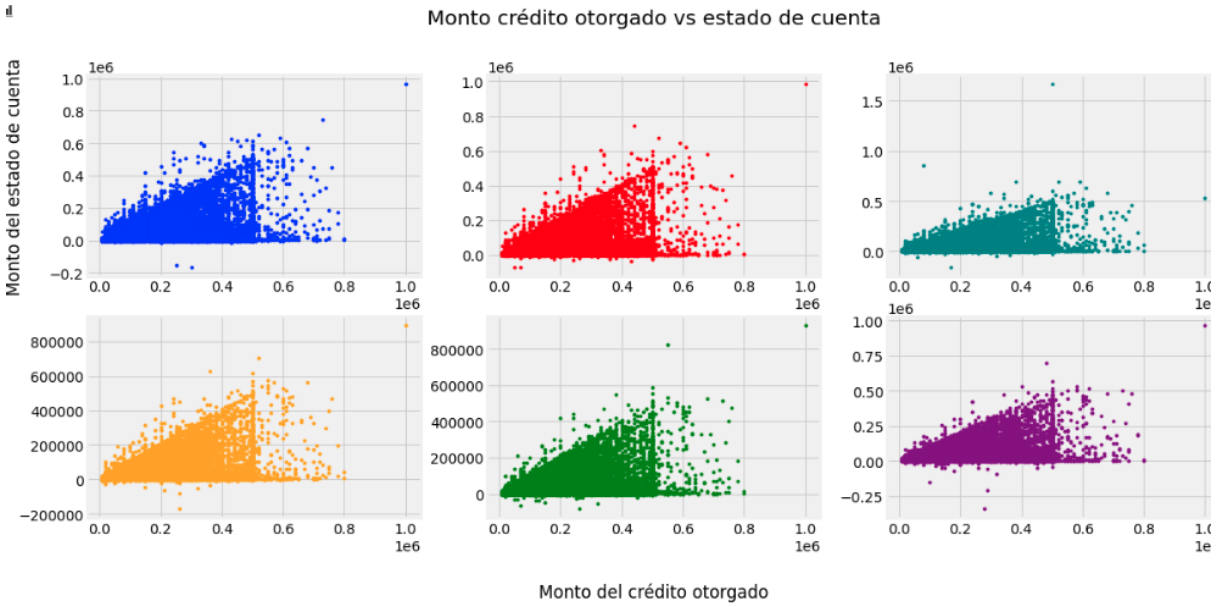


Imagen 8.

En esta imagen hicimos un diagrama de dispersión entre el monto del crédito otorgado y el estado de la cuenta.

En el Heatmap vimos que estas variables tenían un coeficiente de correlación de 0.3 aproximadamente, por lo que se puede decir que existe una relación leve entre estas.



Preprocesamiento de datos

Se parte bajo la condición que no existen valores nulos dentro del data set.
Tampoco hay valores faltantes, cada columna tiene sus respectivos 30000 datos.

#	Column	Non-Null Count	Dtype
0	ID	30000 non-null	int64
1	LIMIT_BAL	30000 non-null	int64
2	SEX	30000 non-null	int64
3	EDUCATION	30000 non-null	int64
4	MARRIAGE	30000 non-null	int64
5	AGE	30000 non-null	int64
6	PAY_1	30000 non-null	int64
7	PAY_2	30000 non-null	int64
8	PAY_3	30000 non-null	int64
9	PAY_4	30000 non-null	int64
10	PAY_5	30000 non-null	int64
11	PAY_6	30000 non-null	int64
12	BILL_AMT1	30000 non-null	int64
13	BILL_AMT2	30000 non-null	int64
14	BILL_AMT3	30000 non-null	int64
15	BILL_AMT4	30000 non-null	int64
16	BILL_AMT5	30000 non-null	int64
17	BILL_AMT6	30000 non-null	int64
18	PAY_AMT1	30000 non-null	int64
19	PAY_AMT2	30000 non-null	int64
20	PAY_AMT3	30000 non-null	int64
21	PAY_AMT4	30000 non-null	int64
22	PAY_AMT5	30000 non-null	int64
23	PAY_AMT6	30000 non-null	int64
24	default	30000 non-null	int64

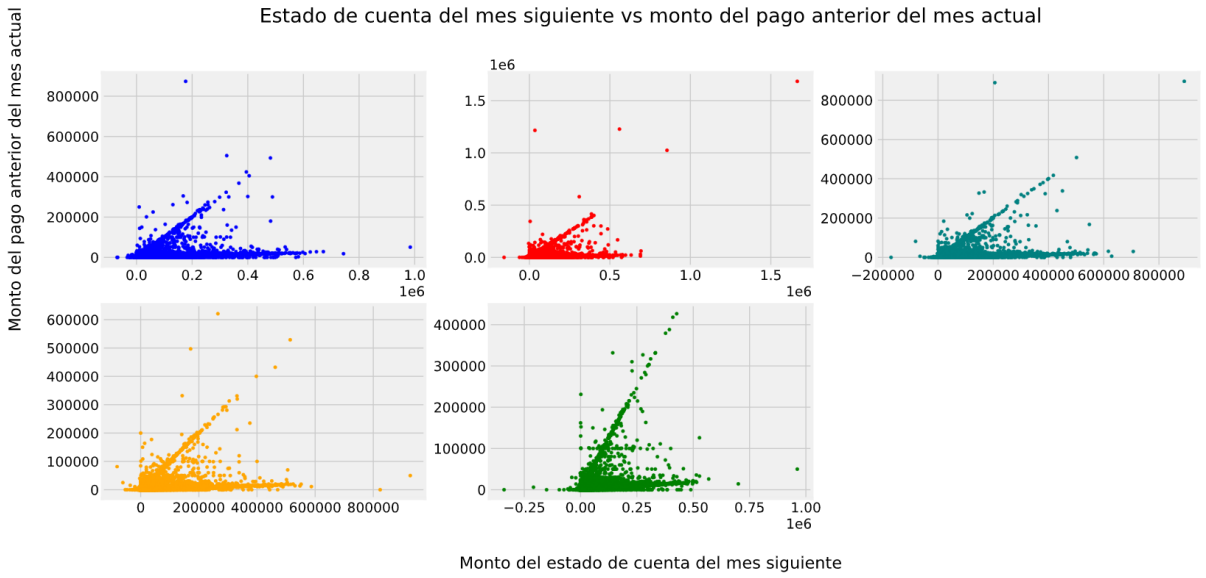
ID: se eliminó esta columna.
Educación: se tomó los valores que no eran parte de las categorías(0, 5 y 6) según la documentación y se cambió por 4(otros).
Estado civil: se cambió los valores que no pertenecen a las categorías definidas en la documentación (0 en este caso) y se cambió por 3.
Sexo: no fue necesario hacer un preprocesamiento adicional

Las decisiones que se tomaron fue considerando que las variables tenían dentro de sus categorías una variable “otros” por lo tanto los datos no pertenecientes a las categorías documentadas podrían ser asignadas a dicha categoría, además los datos no categorizados dentro de las variables era muy bajo con respecto al conjunto total de datos por su respectiva columna, es más la variable educación tenía dentro de sus datos solo un 1.15% de datos no categorizados según documentación.
Otras posibles soluciones con respecto al preprocesamiento de los datos pudo haber sido la eliminación de las tuplas completas en donde alguna de sus columnas no estaban bien

categorizadas o reemplazar dichos valores por la media de la columna debido a que esos valores no categorizados no eran los suficientes para afectar considerablemente la media o la mediana.

Diseño de experimentos

Se considero el experimento de relacionar 2 variables las cuales aparentemente tenían una correlación bastantante fuerte, las cuales no podrían ayudar a nuestros algoritmos a predecir con mayor exactitud
en los cuales se encontró un patrón entre el estado de cuenta del mes siguiente y el monto del pago anterior.



Revisión métodos relacionados

Primero

Predictive Analysis of Credit Score for Credit Card Defaulters
Nupura Torvekar, Pravin S. Game - Enero 2019

https://www.researchgate.net/profile/Pravin-Game/publication/332557433_Predictive_analysis_of_credit_score_for_credit_card_defaulters/links/5f44ec63458515b7294fc74c/Predictive-analysis-of-credit-score-for-credit-card-defaulters.pdf

Que proponen:

Los autores le dan énfasis al contexto en el que se encuentra el sector bancario afirmando que es uno de los más volátiles y vulnerables en el mundo con sus factores de riesgo cada vez mayores.
El riesgo de crédito continúa siendo un factor integral para que las instituciones bancarias sufran pérdidas del orden de cientos de millones de dólares debido a la imposibilidad de recuperar el dinero concedido a los clientes.

Que utilizaron:

Naive Bayes, Regresión logística, máquinas de vectores de soporte y bosque aleatorio. Los algoritmos anteriores se evalúan utilizando entorno weka para aprendizaje automático y la minería de datos, además se usa KNIME que es una plataforma de minería de datos que permite el desarrollo de modelos en un entorno visual.

Que encontraron:

Los autores afirman que el uso de técnicas de aprendizaje automático para la predicción de los morosos de tarjetas de crédito es esencial para la identificación de riesgo crediticio. Esto puede ayudar a las instituciones financieras a diseñar sus estrategias futuras.
En cuanto a los algoritmos utilizados los autores evaluaron los clasificadores utilizando la precisión de su predicción. Encontraron que el clasificador con la mayor precisión es el bosque aleatorio.

Segundo

Machine Learning Approaches to Predict Default of Credit Card Clients
Ruilin Liu - 2018.
https://www.scirp.org/html/7-7201927_88577.htm?pagespeed=noscript

Que proponen:
El autor afirma que la red neuronal puede explorar la relación entre las características de entrada y las etiquetas correspondientes, por lo que es adecuada para problemas complejos de aprendizaje automático. Por otro lado, otros modelos de aprendizaje automático como la regresión lineal o máquinas vectores de soporte pueden resolver problemas más simples de manera más eficiente. Por tanto, después de analizar problemas específicos, se debe responder a la pregunta de "¿es realmente necesaria la red neuronal en este caso?Además el autor menciona que existen varios tipos de redes neuronales.

Que utilizaron:
El autor utiliza varios modelos entre ellos k-vecinos cercanos, máquinas vector de soporte, árbol de decisión, bosques aleatorios, redes neuronales y redes neuronales recurrentes.

Que encontraron:
El autor afirma que los modelos tradicionales de aprendizaje automático solo pueden lograr una precisión de 0,8040, que se logra con SVM es decir mayor a los demás. La mayor precisión de la red neuronal es 0,8246, por lo tanto el autor concluye que las redes neuronales superan a los modelos tradicionales, excepto en situaciones en las que la investigación se centra fuertemente en predicciones positivas.

Tercero

Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithm
Hongya Lu, Haifeng Wang and Sang Won Yoon Department of Systems Science and Industrial Engineering State University of New York at Binghamton, Binghamton, NY- 2017

https://www.researchgate.net/profile/Haifeng-Wang-25/publication/319689046_Real_Time_Credit_Card_Default_Classification_Using_Adaptive_Boosting-Based_Online_Learning_Algorithm/links/59b991e1458515bb9c48a3f8/Real-Time-Credit-Card-Default-Classification-Using-Adaptive-Boosting-Based-Online-Learning-Algorithm.pdf

Que proponen:
Los investigadores proponen una aplicación de aprendizaje en línea al sistema de detección de incumplimiento de tarjetas de crédito que logra un ajuste del modelo en tiempo real con un mínimo esfuerzo computacional. Para los emisores de tarjetas de crédito, el

número de clientes de tarjetas de crédito, la cantidad de consumo y las tasas de incumplimiento son factores que influyen en la participación de los bancos en el mercado. Los autores además afirman que el banco sufrirá pérdidas debido a una mala gestión de la administración de tarjetas de créditos.

Que utilizaron:
Los autores utilizaron una técnica que se llama aprendizaje. Los investigadores explican que el aprendizaje en línea representa una familia de algoritmos eficientes y escalables en comparación con el aprendizaje por lotes tradicional, y aborda los problemas de la memoria. costes de consumo y reciclaje con nuevos datos entrantes.
Las técnicas exactas que utilizan los investigadores son métodos de aprendizaje extremo secuencial en línea (OS-ELM) y aumento de adaptación secuencial en línea (OS-AdaBoost). El OS-ELM está adaptado de la Extreme Learning Machine básica
(ELM) para permitir que el modelo aprenda uno a uno fragmento a fragmento . El ELM básico propone una red feedforward con una única capa oculta generalizada que no funciona estrictamente como neuronas.

Que encontraron:
Los autores mencionan que con las técnicas en línea a otras técnicas con determinadas medidas de rendimiento, se obtienen resultados experimentales en comparación a modelos tradicionales de aprendizaje automático, además los autores afirman que el aprendizaje en línea tiene un gran potencial para los problemas de identificación en tiempo real y que esa idea genera futuras discusiones y un gran potencial investigativo.

Cuarto

Estimation of Credit Card Customers Payment Status by Using kNN and MLP
Murat KOKLU, Kadir SABANCI - 2016
https://www.ijisae.org/IJISAE/article/view/969/546

Que proponen:
Los autores afirman que para los bancos, lo más importante durante la comercialización de tarjetas de crédito es la capacidad de pago de los clientes. Los investigadores en el estudio proponen una estimación del estado de pago para los clientes de tarjetas de crédito. Para ello han utilizado algoritmos de minería de datos. Los autores describen la minería de datos como un proceso computacional que revela patrones en conjuntos de datos utilizando métodos como inteligencia artificial, aprendizaje automático, estadísticas, etc. Los métodos utilizados en la minería de datos se investigan en dos grupos: predictivos y descriptivos

Que utilizaron:
Los autores utilizaron entorno Weka para el análisis de aprendizaje automático y recurrieron al uso de modelos predictivos como k-nn, mlp(percepción multicapa)

Que encontraron:
Utilizando el algoritmo knn, los autores obtuvieron tasas de éxito de la estimación de pago para diferentes valores de k. Se han alcanzado el error medio absoluto (MAE) y el error cuadrático medio (RMSE), el éxito de la estimación del método k-nn (en porcentaje), además en el estudio mencionan el impacto basado en MAE y RMSE según cuantos k vecinos se elija.

Descripción de algoritmos básicos

Árbol de decisión

Es un modelo analítico que a través de una representación esquemática de las alternativas disponibles facilita la toma de mejores decisiones para obtener un resultado. Su nombre deriva de la apariencia del modelo parecido a un árbol y su uso es amplio en el ámbito de la toma de decisiones bajo incertidumbre (Teoría de Decisiones).
Este modelo sirve para resolver problemas de clasificación.

La división entre valores de entrenamiento se hizo utilizando la técnica hold-out separando 30 test y 70 de entrenamiento.
En el problema que se busca solucionar se utilizó como criterio la entropía y de máxima profundidad un valor de 10.

Naive Bayes

En un sentido muy general, el modelo naive bayes es una clase especial de algoritmos de clasificación de Aprendizaje Automático el cual se basan en el teorema de bayes.
En este tipo de modelo se asume que las variables predictoras son independientes entre sí. En otras palabras, la presencia de una cierta característica en un conjunto de datos no está en absoluto relacionada con cualquier otra característica.

La división entre valores de entrenamiento se hizo utilizando la técnica hold-out separando 20 test y 80 de entrenamiento.
Para implementar el algoritmo se usó como parámetro binarización de 0.5.

Reporte de resultados

Después de que se hayan implementado ambos algoritmos el que obtuvo mejores resultados con una diferencia mínima y tomando en consideración el accuracy fue naive bayes con una métrica accuracy de 0.793 frente a la métrica del árbol de decisión que obtuvo 0.776.

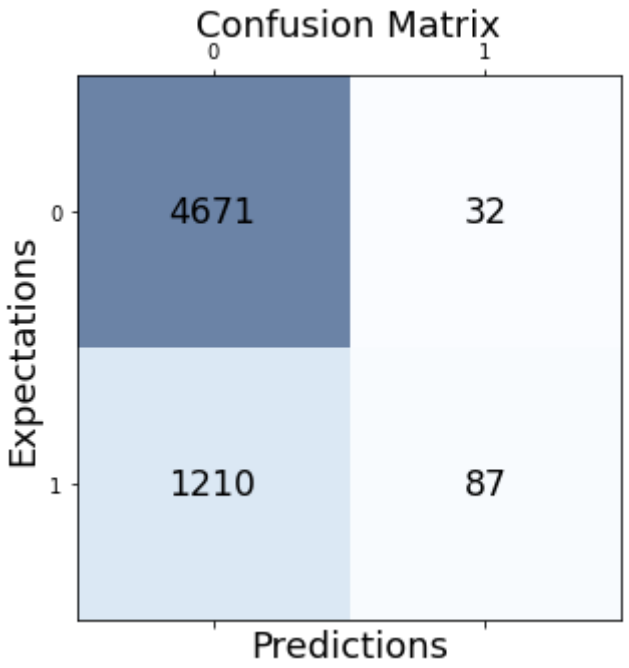


Tabla de Matriz de Confusión

Score de Accuracy	Score de Precision	Score de Recall	Score F1
0.793	0.7310924369747899	0.06707787201233616	0.12288135593220338

Tabla de métricas

Innovación

Revisión métodos relacionados

Primero

Machine Learning Approaches to Predict Default of Credit Card Clients

Ruilin Liu, University of Southern California, Los Angeles, CA, USA - 2018

https://www.scirp.org/html/7-7201927_88577.htm?pagespeed=noscript

Que proponen:
El autor menciona las grandes ventajas de las redes neuronales en explorar la relación entre las características de entrada y las etiquetas correspondientes, por lo que es adecuada para problemas complejos de aprendizaje automático, sin embargo también afirma que en problemas simples los métodos de machine learning tradicionales como regresiones lineales o svm tienen buenos resultados.
Otro punto que toma es diferenciar los distintos tipos de redes neuronales, explica generalmente el funcionamiento de las redes FeedForward y como se encarga esta de crear nuevas capas, además explica de forma muy general el funcionamiento de las redes neuronales recurrentes y su ventaja en datos secuenciales.

Que utilizaron:
El autor decide comparar los modelos tradicionales de aprendizaje automático, es decir, máquina de vectores de soporte, k-vecinos más cercanos, árbol de decisión y Random Forest, con la red neuronal Feedforward y la memoria a corto plazo.

Que encontraron:
El autor concluye que que las dos redes neuronales es decir FeedForward y recurrentes logran mayores precisiones que los modelos tradicionales anteriormente mencionados. Este documento también intenta averiguar si la deserción puede mejorar la precisión de las redes neuronales. El autor observa que para la red neuronal Feedforward, aplicar la deserción puede conducir a mejores rendimientos en ciertos casos pero peores rendimientos en otros. La influencia de la deserción en los modelos LSTM es pequeña. Por lo tanto, usar la omisión no garantiza una mayor precisión.

Segundo

Alternating Optimization of Decision Trees with Application to Learning Sparse Oblique Trees

Miguel A. Carreira Perpiñán, Pooya Tavallali - Dept. EECS, University of California, Merced- 2018
https://faculty.ucmerced.edu/mcarreira-perpinan/papers/neurips18.pdf

Que proponen:
Los autores realizan una pequeña reseña histórica de los árboles de decisión, además explica en términos generales cómo funcionan los árboles diciendo que el crecimiento de un árbol se basa en un crecimiento codicioso que va dividiendo recursivamente los nodos y posiblemente podando.
Además los autores también afirman que los árboles se encuentran entre los modelos estadísticos más utilizados en la práctica de decisión Están habitualmente en la parte superior de la lista en las encuestas anuales de KDnuggets.com de los mejores algoritmos de aprendizaje automático.

Que utilizaron:
Los autores tienen como propuesta un Tree Alternating Optimization (TAO), el cual es un algoritmo escalable que puede encontrar un óptimo de árboles oblicuos dada una estructura fija, en el sentido de disminuir repetidamente la pérdida de clasificación errónea hasta que no se pueda avanzar más.

Que encontraron:
En primera instancia los autores afirman que en ocasiones los árboles de decisión sean preferibles a modelos más precisos, como las redes neuronales, en algunas aplicaciones, como el diagnóstico médico o el análisis legal, debido a que los árboles de decisión se pueden intuir como una secuencia de reglas lo que es fácil de entender para los humanos, incluso los autores mencionan que un árbol se puede transformar en una base de datos de reglas específicas.
Otro punto que toman en cuenta es el propio de la investigación que realizan, el cual dice que TAO podría hacer que los árboles oblicuos se generalizaron en la práctica y reemplazaran en cierta medida los árboles alineados con ejes menos flexibles.
Incluso los investigadores afirman que los TAO pueden lograr un buen compromiso entre el modelado flexible de características (que involucran complejos locales correlaciones, como con los datos de imagen) y usando pocas características en cada nodo, por lo tanto produciendo un árbol precioso que es muy pequeño, rápido e interpretable.

Tercero

Counterfactual Explanations for Oblique Decision Trees: Exact, Efficient Algorithms

Miguel A. Carreira Perpiñán, Suryabhan Singh Hada - Dept. Computer Science & Engineering, University of California, Merced - 2021

https://www.aaai.org/AAAI21Papers/AAAI-6897.Carreira-PerpinanMA.pdf

Que proponen:
Los investigadores hacen parten haciendo un resumen general y una contextualización con respecto a la inteligencia artificial, ellos afirman que la implementación práctica de modelos de aprendizaje profundo y aprendizaje automático se ha generalizado en la última década, y Existe un enorme interés social en la IA como tecnología que puede proporcionar un procesamiento inteligente y automatizado de tareas que hace un tiempo atrás eran difíciles para las máquinas.. Al mismo tiempo, en algunas personas las preocupaciones sobre los sistemas de IA (éticos, de seguridad y otros) han surgido también. Uno es el problema de la interpretabilidad, es decir, explicar la funcionalidad de un sistema automatizado.
En términos específicos los autores se centran en una versión específica del segundo problema que, siguiendo a Wachter, Mittelstadt y Russell (2018), llamaremos explicación contrafáctica, que se refiere al hecho de que un evento no sucedió realmente.
Por ejemplo, “Se le negó un préstamo porque su ingreso anual era de $30,000. Si sus ingresos hubieran sido de $45,000,
se le hubiera ofrecido un préstamo ". La segunda declaración, o
contrafactual, ofrece un evento alternativo que resultaría
en el resultado deseado (aprobación del préstamo). Formalmente, una explicación contrafáctica busca el cambio mínimo en un determinado vector de características que cambiará la decisión de un clasificador en un forma prescrita, y haremos esto más preciso como un problema de optimización.

Que utilizaron:
Los autores buscaron optimizar y utilizar un árbol oblicuo, incluso en el paper de investigación se habla y se utiliza el anteriormente mencionado TAO.

Que encontraron:
Los investigadores concluyeron que los árboles de clasificación son muy importantes en aplicaciones como como negocios, derecho y medicina, donde las explicaciones hipotéticas son de particular relevancia. Los autores afirman que, algoritmo eficiente para calcular explicaciones contrafácticas para árboles alineados con ejes y oblicuos en problemas multiclase, con diferentes distancias y restricciones, y aplicable a tanto características continuas como categóricas. Finalmente los autores mencionan que el algoritmo es lo suficientemente rápido como para permitir incluso un uso interactivo.

Descripción de los algoritmos con innovación

Árbol oblicuo

El árbol de decisiones oblicuo es una opción popular en el campo del aprendizaje automático para mejorar el rendimiento de los algoritmos tradicionales del árbol de decisiones. A diferencia del árbol de decisión tradicional, que usa un punto de división de eje paralelo para determinar si un punto de datos debe asignarse a la rama izquierda o derecha de un árbol de decisión, el árbol de decisión oblicuo usa un hiperplano basado en todos los puntos de datos.

Ventajas:

- Posee mayor rendimiento que los árboles de decisión tradicionales.
- Existen varios paper que incluso mejoran el árbol oblicuo, en otras palabras tiene un gran potencial de escalabilidad.
- Tiene grandes aplicaciones en campos como medicina, negocios y derecho.

Desventajas:

- Si utilizamos sklearn podríamos tener dificultades para implementar soluciones con dicha documentación.
- Su rendimiento es bueno pero no logra superar a técnicas de deep learning

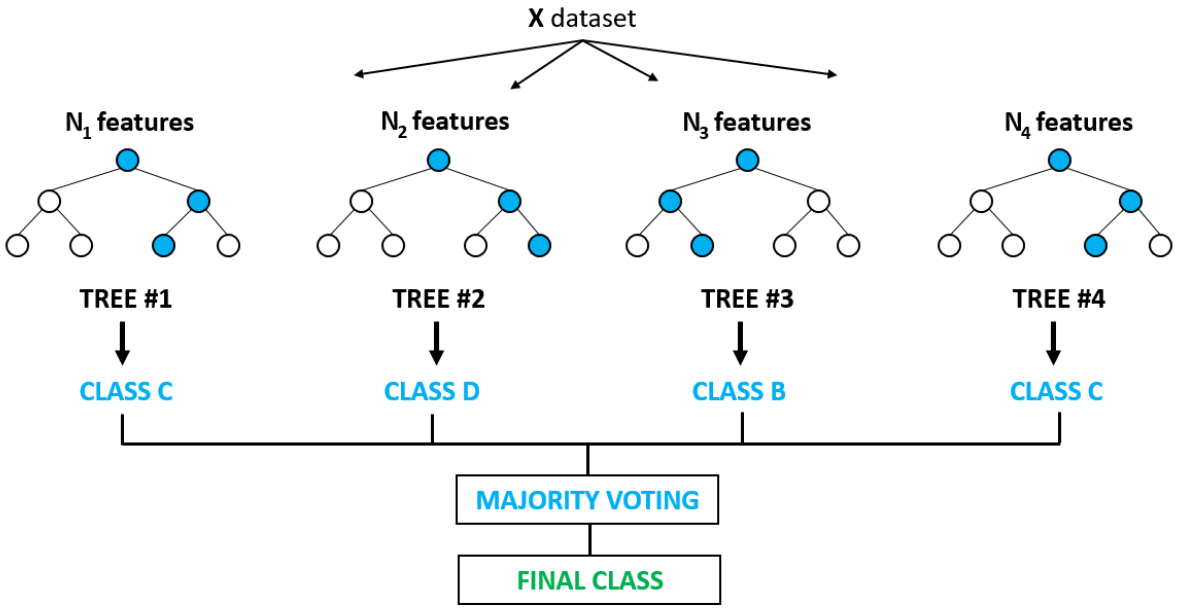
Random forest

Random Forest o bosques aleatorios son un método de aprendizaje conjunto para tareas de clasificación y regresión. En comparación con el árbol de decisiones, los bosques aleatorios garantizan robustez y son menos propensos a sufrir overfitting.
Los bosques aleatorios utilizan una técnica llamada bootstrapping el cual es un método de remuestreo que se usa frecuentemente para aproximar el sesgo o la varianza de un análisis estadístico.
En bosques aleatorios bootstrapping se usa para reducir la varianza de un método de aprendizaje estadístico. Los árboles en general son buenos candidatos para bootsrapping, ya que pueden obtener un sesgo bajo cuando crecen en profundidad. Este algoritmo tiende a tener una alta varianza que se puede reducir utilizando bootstrapping.
En términos generales un árbol aleatorio está formado por un conjunto de árboles de decisión individuales Esto implica que cada árbol de decisión se entrena con unos datos ligeramente distintos he aquí su ventaja frente a los árboles de decisión tradicionales.
Bajo la premisa “El mejor modelo es aquel que consigue un equilibrio óptimo entre sesgo y varianza” los árboles con pocas ramificaciones tienen poca varianza pero no consiguen bien representar las relaciones entre las variables en otras palabras baja varianza pero un sesgo alto, en cambio un árbol con muchas ramificaciones tienen la capacidad de ajustarse mucho a los datos es decir poco sesgo pero el problema es que tienen mucha varianza, esta problemática se puede solucionar como se mencionó anteriormente con bootstrapping, sin embargo esto es demasiado específico para hablar en términos generales por lo que la solución a esta problemática se divide en dos partes:

- Bagging: Se ajustan múltiples modelos cada uno con un subconjunto distinto de los de las observaciones. Entonces para realizar la predicción todos los modelos

generados participan aportando su respectiva predicción. Random forest se encuentra dentro de esta categoría.

- Boosting: Se ajustan secuencialmente múltiples modelos sencillos los cuales se llaman weak learners, de forma que cada modelo aprende de los errores del modelo anterior, los métodos más empleados dentro de esta categoría son AdaBoost, Gradient Boosting y Stochastic Gradient Boosting. Cabe destacar que dichas aplicaciones fueron mencionadas en algunos de los papers analizados durante este trabajo investigativo obteniendo buenos resultados y con aplicaciones en entornos en tiempo real.



Ventajas:

- No requiere de estandarización
- Requiere de un preprocesado mucho menos exhaustivo que otros algoritmos.
- Se ve poco influenciado por outliers
- Gracias al Out-of-Bag se puede estimar su error de validación sin necesidad de recurrir a operaciones computacionalmente costosas como cross-validation.
- Es un algoritmo bastante escalable por lo tanto permite tener un número elevado de observaciones.
- Es más robusto y menos propenso a sufrir overfitting en comparación a un árbol de decisión.

Desventajas:

- No logran extrapolar fuera del rango de los predictores observados en los datos de entrenamiento.

- Al ser conformado por múltiples árboles se pierde interpretabilidad en comparación a un árbol de decisión tradicional.

Implementación de los algoritmos con innovación

Árbol oblicuo

Aquí tomamos las características y las normalizamos

```
X_pre = RobustScaler().fit_transform( X.values )
```

En este bloque instalamos el modelo con un random state de 2020 y una iteración máxima de 1000.

```
stree = Stree(random_state=2020, C=.01, max_iter=1e3)
```

En esta parte separamos la data de testing y la de training

```
Xtrain, Xtest, ytrain, ytest = train_test_split(X_pre, y, train_size=.2, shuffle=True, random_state=2020, stratify=y)
```

El modelo. Aquí se entrena , se calcula la matriz de confusión, el f1 score.

```python
def try_model(model):

    now = time.time()
    model.fit(Xtrain, ytrain)
    spent = time.time() - now

    predict = model.predict(Xtrain)
    predictt = model.predict(Xtest)
    print(f"\n\n************************** Stree **********************")
    print(f"Train Model Stree took: {spent:.4} seconds")
    print(f"=========== Stree - Train {Xtrain.shape[0]:,} samples ============",)
    print(classification_report(ytrain, predict, digits=6))
    print(f"=========== Stree - Test {Xtest.shape[0]:,} samples ============")
    print(classification_report(ytest, predictt, digits=6))
    print("Confusion Matrix in Train")
    print(confusion_matrix(ytrain, predict))
    print("Confusion Matrix in Test")
    print(confusion_matrix(ytest, predictt))
    print(f"*********************************************************\n\n")
    return f1_score(ytest, predictt), spent
```

Random forest bajo esquema cross-validation

Se implementó usando one hot encoding, gridSearchCV y optimizando sus hiperparametros intentado que el tiempo computacional no creciera excesivamente.

Ajuste del modelo y optimización de los hiperparametros

```python
X_train, X_test, y_train, y_test = train_test_split(
                                    dfRF.drop(columns = 'default'),
                                    dfRF['default'],
                                    random_state = 123
                                )
```

Se utiliza GridSearchCV, RandomForest bootstrap activado y uso de cross-validation

```
param_grid = {'n_estimators': [10, 15, 20],
              'max_features': [4, 7, 9],
              'criterion'   : ['gini', 'entropy']
             }

# Búsqueda por grid search con validación cruzada y randomforest con tecnica bootstrapping

grid = GridSearchCV(
        estimator  = RandomForestClassifier(random_state = 123, bootstrap=True),
        param_grid = param_grid,
        scoring    = 'accuracy',
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = RepeatedKFold(n_splits=5, n_repeats=3, random_state=123),
        refit      = True,
        verbose    = 0,
        return_train_score = True
       )

grid.fit(X = X_train_prep, y = y_train)
```

Elegimos el mejor modelo en base a los hiperparametros

	param_criterion	param_max_features	param_n_estimators	mean_test_score	std_test_score	mean_train_score	std_train_score
17	entropy	9	20	0.812281	0.005089	0.992707	0.000649
14	entropy	7	20	0.810889	0.004796	0.992696	0.000611
5	gini	7	20	0.810459	0.005091	0.992867	0.000761
11	entropy	4	20	0.810119	0.004673	0.992800	0.000771

Mejor modelo al usar el parámetro refit = true del GridSearchCV automáticamente se guarda el mejor modelo disponible en best_estimator_

```
# Guardamos el modelo más optimo segun los hiperparametros
modelo_final_Random_Forest_CV = grid.best_estimator_
```

Realizamos las predicciones utilizando el test

```
#
predicciones_Random_Forest = modelo_final_Random_Forest_CV.predict(X = X_test_prep)
predicciones_Random_Forest[:20]
```

Reporte de resultados obtenidos

Árbol oblicuo

Medidas de rendimiento:
En entrenamiento:

```
Train Model Stree took: 3.371 seconds
========== Stree - Train 6,000 samples =============
              precision    recall  f1-score   support

           0   0.858092  0.912262  0.884348      4673
           1   0.602713  0.468726  0.527342      1327

    accuracy                       0.814167      6000
   macro avg   0.730402  0.690494  0.705845      6000
weighted avg   0.801611  0.814167  0.805390      6000
```

En Testing:

```
========== Stree - Test 24,000 samples =============
              precision    recall  f1-score   support

           0   0.853929  0.909850  0.881003     18691
           1   0.587515  0.452063  0.510964      5309

    accuracy                       0.808583     24000
   macro avg   0.720722  0.680956  0.695984     24000
weighted avg   0.794996  0.808583  0.799147     24000
```

El modelo demoró 3.37 segundos en ejecución y obtuvo un F1 Score de 0,51.
si tenemos en cuenta que el F1 score es

Recall = TP/TP+FN

Precision = TP/TP+FP

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Se puede decir que la precisión que tiene no es muy buena en este caso específicamente. Aunque cabe destacar que si lo comparamos con el árbol de decisión y el Random Forest, este modelo tuvo una gran mejoría (es el mejor).

```
Model:     Stree         Tiempo:    3.37 Segundos         f1: 0.5109644453906749
```

Matriz de confusión:

```
Confusion Matrix in Train
[[4263  410]
 [ 705  622]]
Confusion Matrix in Test
[[17006  1685]
 [ 2909  2400]]
```

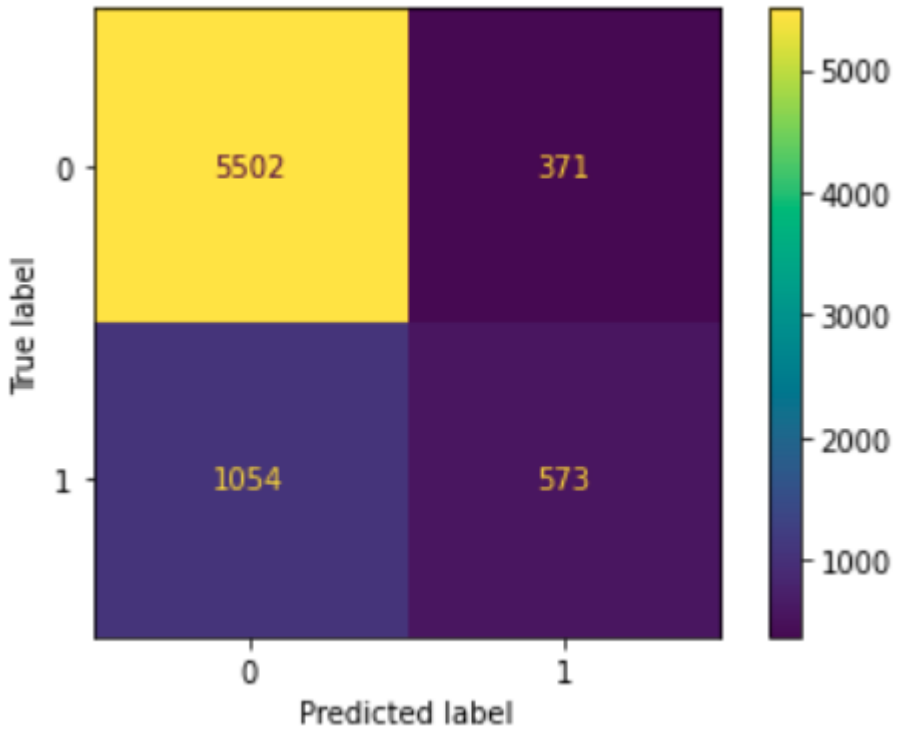
Random Forest enfoque cross-validation

Medidas de rendimiento:
En entrenamiento se obtuvieron los siguientes hiperparametros óptimos y accuracy:

```
Mejores hiperparámetros encontrados (cross validation)
{'criterion': 'entropy', 'max_features': 9, 'n_estimators': 20} : 0.8122814814814814 accuracy
```

En Test se obtuvo los siguientes resultados:

Matriz de confusión



Medidas de rendimiento en Test:

```
reporte_clasificacion_Random_forest= classification_report(
        y_true = y_test,
        y_pred = predicciones_Random_Forest
    )
```

	precision	recall	f1-score	support
0	0.84	0.94	0.89	5873
1	0.61	0.35	0.45	1627
accuracy			0.81	7500
macro avg	0.72	0.64	0.67	7500
weighted avg	0.79	0.81	0.79	7500

Alcanzando un accuracy de 81.0% en Test, es decir un excelente rendimiento en cuanto al entrenamiento.

Conclusiones de resultados

Con respecto al Random Forest con enfoque cross-validation podemos decir que es un muy buen algoritmo el cual es bastante escalable y presenta considerables mejoras con respecto a un árbol de decisión simple, es más en una de las fuentes bibliográficas se mencionaba que los árboles aleatorios suelen ser la panacea si de modelos de clasificación hablamos, esta afirmación era sin considerar al deep learning.
Con Random Forest obtuvimos en test un 81% de accuracy lo que es un excelente resultado.
En cuanto a el árbol oblicuo si comparamos el modelos con los demás modelos, podríamos decir que es bueno si nos basamos en el accuracy, ya que este obtuvo uno de 80,8%.
El problema es cuando vemos el f1 score (el cual es de 0.51) porque si recordamos, la precisión es la relación entre las observaciones positivas predichas correctamente y el total de observaciones positivas predichas y el f1 es promedio ponderado de precisión y Sensibilidad, Por lo que esta puntuación tiene en cuenta tanto los falsos positivos como los falsos negativos. Entonces logra acertar bastante positivamente pero también se equivoca en varios negativos, por lo que en el caso de ser un caso real para tomar una decisión importante, este quizas no seria el mejor, aunque también hay que recalcar que tampoco es el peor, de hecho es mejor de los quemos probado por lo que desde ese punto de vista es bastante bueno.
Quizás tocando algunos de los parámetros se podría obtener un mejor resultado.

Conclusión

Finalmente podemos considerar que existen 3 posibles caminos si hablamos de preprocesamiento los cuales eran eliminar, reemplazar por media/mediana y re asignar los valores mal categorizados a la categoría que correspondan (otros).
En cuanto a las correlaciones que se pudieron encontrar en los diagramas 2D fueron el monto crédito otorgado vs el estado de cuenta donde existía una correlación positiva un tanto débil pero considerable, entre otras variables con relaciones interesantes. Cabe destacar que en esta primera etapa se puede evidenciar que existen relaciones entre las variables y que esto nos permitirá a futuro entrenar modelos predictivos de manera idónea.
Otro punto importante que podemos concluir en base a los paper mencionados en este trabajo, es que existen múltiples caminos para dar solución a este problema, algunos caminos priorizan la eficiencia en función del tiempo para eventualmente poder implementar la solución en entornos de tiempo real, otras formas utilizan algoritmos más tradicionales los cuales prácticamente son un estándar cuando de clasificar hablamos, sin embargo el método que se considera como mejor solución actual en condiciones normales es el uso de redes neuronales en otras palabras deep learning, este método es catalogado como la mejor solución en dos de los cuatro paper, y en la página oficial de ics uci.
Actualmente el deep learning, nlp y la inteligencia artificial nos ha llevado a avanzar a pasos agigantados en múltiples problemas que anteriormente la solución no era la mejor o simplemente no era óptima.

Bibliografía

- https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
- https://pandas.pydata.org/docs/
- https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py
- https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html

- https://www.researchgate.net/profile/Pravin-Game/publication/332557433_Predictive_analysis_of_credit_score_for_credit_card_defaulters/links/5f44ec63458515b7294fc74c/Predictive-analysis-of-credit-score-for-credit-card-defaulters.pdf
- https://www.ijisae.org/IJISAE/article/view/969/546
- https://www.researchgate.net/profile/Haifeng-Wang-25/publication/319689046_Real_Time_Credit_Card_Default_Classification_Using_Adaptive_Boosting-Based_Online_Learning_Algorithm/links/59b991e1458515bb9c48a3f8/Real-Time-Credit-Card-Default-Classification-Using-Adaptive-Boosting-Based-Online-Learning-Algorithm.pdf
- https://www.scirp.org/html/7-7201927_88577.htm?pagespeed=noscript
- https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://scikit-learn.org/stable/modules/grid_search.html#grid-search
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://www.aaai.org/AAAI21Papers/AAAI-6897.CarreiraPerpinanMA.pdf
- https://faculty.ucmerced.edu/mcarreira-perpinan/papers/neurips18.pdf
- https://www.scirp.org/html/7-7201927_88577.htm?pagespeed=noscript
- https://rpubs.com/Avalos42/randomforest
- https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=F1%20score%20%2D%20F1%20Score%20is,have%20an%20uneven%20class%20distribution.
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
- https://github.com/Doctorado-ML/STree
- https://www.kdnuggets.com/