CODER HOUSE

Presentación final Credit Risk Dataset

Equipo de Trabajo:

Matías Manero

Matias Alvarez

Mateo Maya Uparela

Profesor:

Octavio Lafourcade

Tutor:

Juan Felipe Gonzalez Sanmiguel

Comisión 29745 Data Science

Fecha 21/10/2022

Índice

Problemática

El problema del negocio es poder conocer si un cliente que viene a solicitar un problema es apto o no para el otorgamiento del mismo.

Objetivos Generales

En virtud de la amplia base de datos con la que trabajamos, y mediante la aplicación de un algoritmo de Machine Learning podremos predecir en base a las características del nuevo cliente, si es apto o no para otorgar el crédito.

Trabajamos con una base de 32.581 datos de clientes que actualmente tiene préstamos, y conocemos su edad, el destino del préstamo, si es dueño o no de su hogar, el grado del préstamo, ingreso anual, antigüedad en su empleo, monto del préstamo, tasa de interés del mismo, estado del préstamo (al día o en default), qué porcentaje del préstamo es de su ingreso, y el historial crediticio de dicha persona.

Objetivos de la investigación

Objetivo principal

La finalidad de este trabajo será conocer cuáles son las variables que más afectan al status(default o no) de un préstamo, y poder armar un modelo que detecte en base a las variables más relevantes si un potencial cliente entrará en default o no. Nuestro problema es la clasificación.

Objetivos secundarios

Realizar un completo análisis exploratorio de los datos, que nos permita extraer insights e información relevante en el comportamiento de los clientes y su correlación en el uso y obtención de préstamos.

Adquisición de datos (Data Acquisition)

Si bien originalmente habíamos realizado trabajos con un dataset con datos macroeconómicos, optamos por cambiar al análisis del sector crediticio, dado que nos pareció más interesante y tiene mucho uso en la práctica bancaria actual (uno de los integrantes del grupo es ex empleado bancario), por ejemplo, vemos que automáticamente nos ofrecen préstamos, tarjetas de crédito, apertura de cuentas, entre otros. Incluso, al asistir a una sucursal bancaria, simplemente con tu DNI, pueden por sistema ver si tienes alguna oferta disponible. Por ende, es una temática que nos toca a todos y nos pareció una buena opción.

El dataset fue extraído de Kaggle y posee 32581 filas y 12 columnas.
Fuente: https://www.kaggle.com/datasets/laotse/credit-risk-dataset

Detalle de cada columna:

person_age = Edad de la persona tomadora del préstamo.

person_income = Ingreso anual de la persona tomadora del préstamo.

person_home_ownership = Si la persona posee casa, alquila o tiene hipoteca.

person_emp_length = Cuántos años lleva la persona en su trabajo.

loan_intent = Destino del préstamo.

loan_grade = Grado del préstamo.

loan_amnt = Monto del préstamo otorgado.

loan_int_rate = Tasa de interés del préstamo.

loan_status = 0 es no default, 1 es default.

loan_percent_income = Cuanto porcentaje del ingreso anual del cliente representa el préstamo.

cb_person_default_on_file = Yes or no, default historico.

cb_person_cred_hist_length = Longitud de la historia crediticia de la persona.

Limpieza de datos (Data Wrangling)

Consiste en la manipulación, limpieza y unificación de conjuntos de datos complejos y desordenados para facilitar su acceso, análisis y modelado. El proceso incluye convertir y mapear los datos crudos, y dejarlos en un formato más adecuado para su uso.

Tenemos consideración especial es este apartado ya que entendemos que aproximadamente el 60% del esfuerzo de un trabajo de Data Science consiste en Data Wrangling.

Las etapas se componen en descubrimiento, estructuración, limpieza, enriquecimiento, validación, publicación.

Información y característica de los datos:

Vemos el formato y la cantidad de no nulos:

```
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  32581 non-null  int64
 1   person_income               32581 non-null  int64
 2   person_home_ownership       32581 non-null  object
 3   person_emp_length           31686 non-null  float64
 4   loan_intent                 32581 non-null  object
 5   loan_grade                  32581 non-null  object
 6   loan_amnt                   32581 non-null  int64
 7   loan_int_rate               29465 non-null  float64
 8   loan_status                 32581 non-null  int64
 9   loan_percent_income         32581 non-null  float64
 10  cb_person_default_on_file   32581 non-null  object
 11  cb_person_cred_hist_length  32581 non-null  int64
```

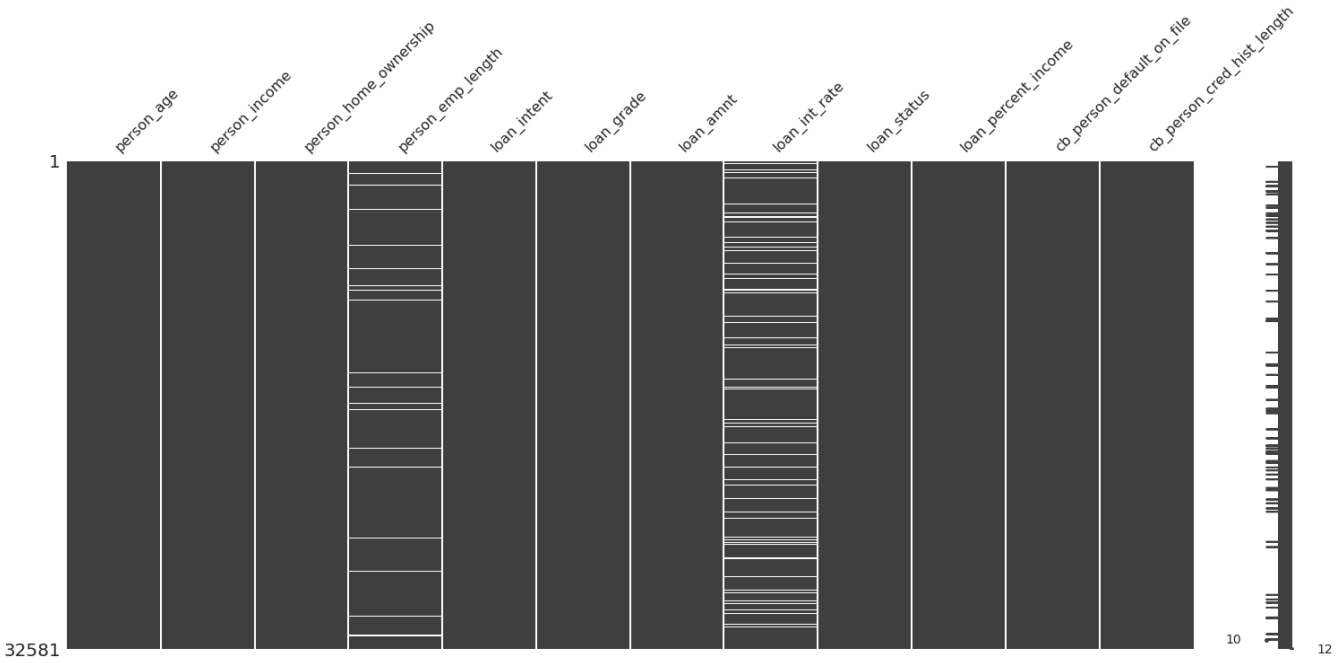
Usamos la función *describe* para tener una visión global de los datos numéricos:

	count	mean	std	min	25%	50%	75%	max
person_age	32581.00000	27.73460	6.34808	20.00000	23.00000	26.00000	30.00000	144.00000
person_income	32581.00000	66074.84847	61983.11917	4000.00000	38500.00000	55000.00000	79200.00000	6000000.00000
person_emp_length	31686.00000	4.78969	4.14263	0.00000	2.00000	4.00000	7.00000	123.00000
loan_amnt	32581.00000	9589.37111	6322.08665	500.00000	5000.00000	8000.00000	12200.00000	35000.00000
loan_int_rate	29465.00000	11.01169	3.24046	5.42000	7.90000	10.99000	13.47000	23.22000
loan_status	32581.00000	0.21816	0.41301	0.00000	0.00000	0.00000	0.00000	1.00000
loan_percent_income	32581.00000	0.17020	0.10678	0.00000	0.09000	0.15000	0.23000	0.83000
cb_person_cred_hist_length	32581.00000	5.80421	4.05500	2.00000	3.00000	4.00000	8.00000	30.00000

Vemos por el conteo de datos que algunas columnas tienen datos faltantes, realizamos el análisis de esos missing values:

```
person_age                    0
person_income                 0
person_home_ownership         0
person_emp_length           895
loan_intent                   0
loan_grade                    0
loan_amnt                     0
loan_int_rate              3116
loan_status                   0
loan_percent_income           0
cb_person_default_on_file     0
cb_person_cred_hist_length    0
```

Los visualizamos:



Luego, comprendemos que en el caso de la columna person_emp_length, los valores nulos corresponden a personas que están recién ingresando a su nuevo trabajo.

person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on_file	cb_person_cred_hist_length
22	12600	MORTGAGE	NaN	PERSONAL	A	2000	5.42000	1	0.16000	N	4
24	185000	MORTGAGE	NaN	EDUCATION	B	35000	12.42000	0	0.19000	N	2
24	16800	MORTGAGE	NaN	DEBTCONSOLIDATION	A	3900	NaN	1	0.23000	N	3
25	52000	RENT	NaN	PERSONAL	B	24000	10.74000	1	0.46000	N	2
22	17352	MORTGAGE	NaN	EDUCATION	C	2250	15.27000	0	0.13000	Y	3

En tanto que para los nulos de tasa de interés, calculamos la mediana del resto de la base de datos y aplicamos dicho número a los nulos optamos por utilizar la

mediana dado que cuando los datos no se ajustan a una distribución normal es más correcto utilizar la mediana. Esto es así porque la mediana es mucho más robusta, lo que quiere decir que se afecta menos por la presencia de sesgos en la distribución o de valores extremos.

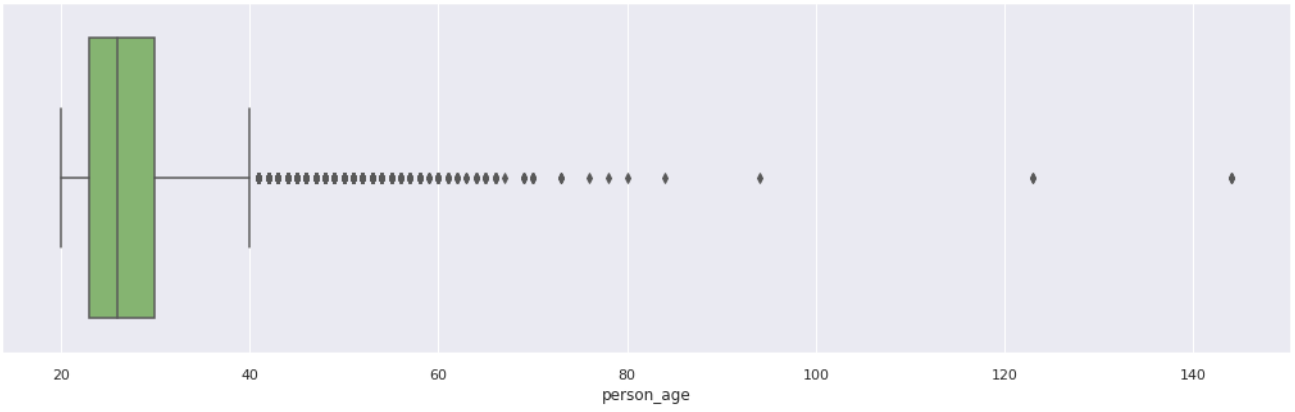
Graficamos un *describe* con colores gradientes para visualizar posibles outliers

	person_age	person_income	person_emp_length	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_cred_hist_length
count	32581.000000	32581.000000	32581.000000	32581.000000	32581.000000	32581.000000	32581.000000	32581.000000
mean	27.734600	66074.848470	4.658114	9589.371106	11.009620	0.218164	0.170203	5.804211
std	6.348078	61983.119168	4.159669	6322.086646	3.081611	0.413006	0.106782	4.055001
min	20.000000	4000.000000	0.000000	500.000000	5.420000	0.000000	0.000000	2.000000
25%	23.000000	38500.000000	2.000000	5000.000000	8.490000	0.000000	0.090000	3.000000
50%	26.000000	55000.000000	4.000000	8000.000000	10.990000	0.000000	0.150000	4.000000
75%	30.000000	79200.000000	7.000000	12200.000000	13.110000	0.000000	0.230000	8.000000
max	144.000000	6000000.000000	123.000000	35000.000000	23.220000	1.000000	0.830000	30.000000

Outliers

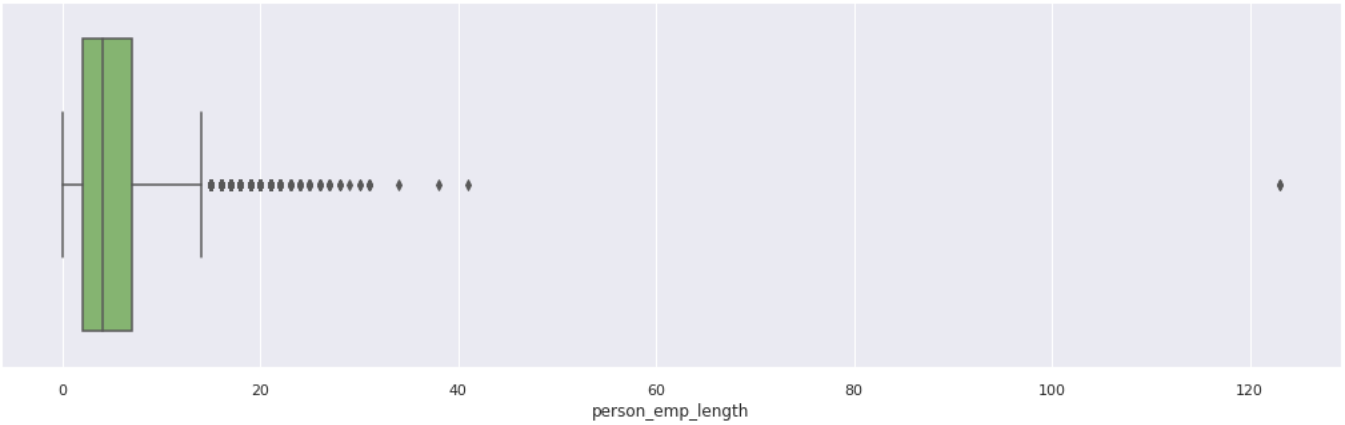
Comenzamos con el análisis y manejo de outliers. Realizamos algunos gráficos para tratar de detectarlos:

Edad:



Ingreso anual:

Duración de empleo:

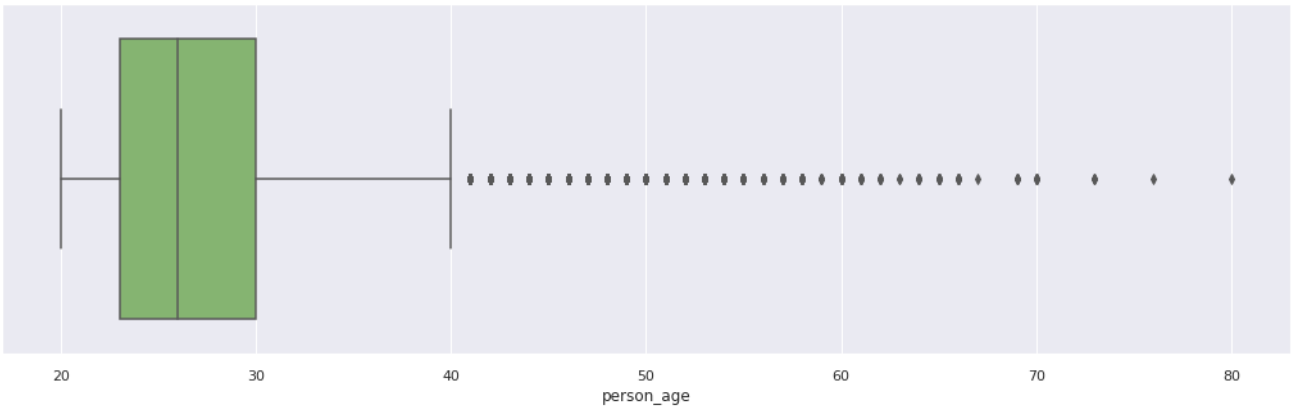


Tratamiento de outliers

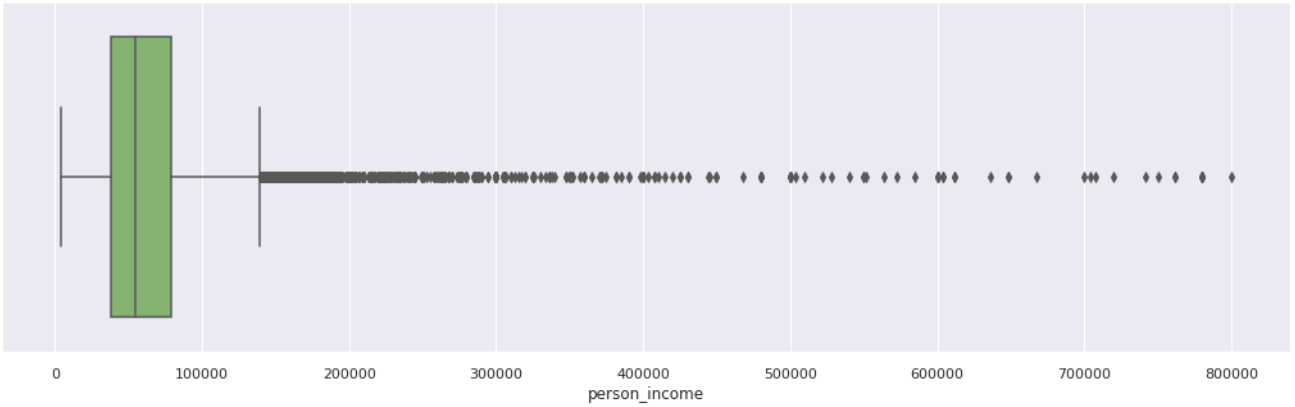
Debido a la presencia de outliers en edad, duración en el empleo e ingreso, le ponemos un máximo de 80 años para los solicitantes de crédito, de 800.000 anuales al ingreso y de 30 años de duración en el empleo.

Graficamos nuevamente para observar como quedan las variables tras el trabajo con los outliers:

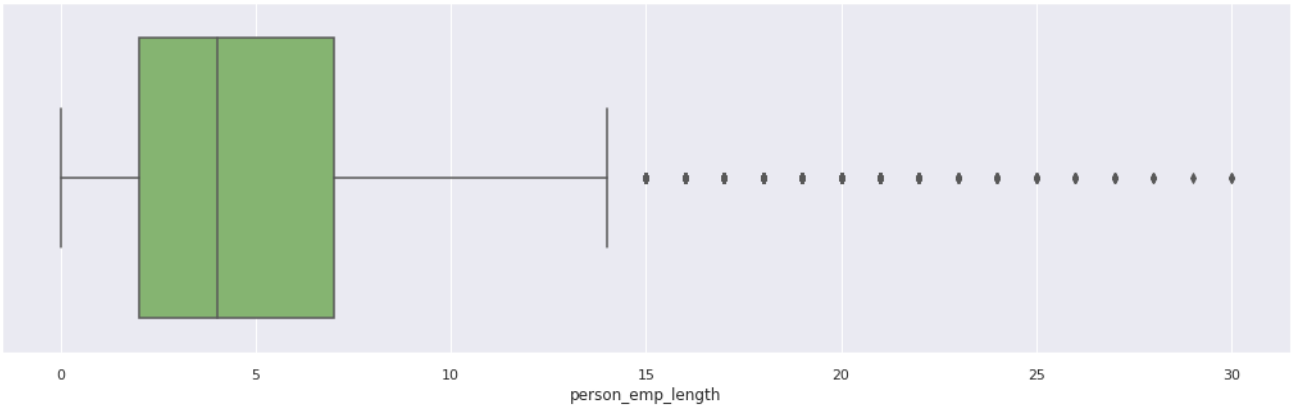
Edad:



Ingreso anual:



Duración de empleo:



Conclusiones del Data Wrangling

- Trabajamos con un dataset que cuenta 32051 registros y 12 columnas con atributos de cada cliente.

- Se detectan valores faltantes en las variables loan_int_rate (tasa de interés del crédito) y person_emp_lenght (antigüedad del cliente en su trabajo).

- En la tasa de interés del crédito, los valores faltantes ascienden a casi el 9.5% de la base. Dado que no tiene una distribución normal, optamos por calcular la mediana de la variable, y aplicarla a los missing values.

- Mientras que en la duración del cliente en su trabajo, reemplazamos por 0 dado que entendemos que son personas recién ingresadas en su trabajo. Por lo tanto la duración es de 0 años. En esta variable, los missing values ascendían a un poco más del 2% del total.

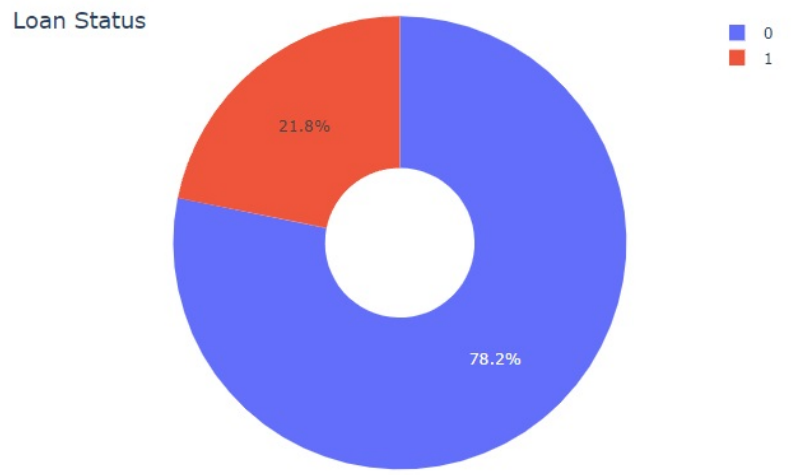
- Se detectan valores outliers en las variables person_age (edad del cliente), person_income (ingreso anual del cliente), y person_emp_lenght (la ya mencionada antigüedad del cliente en su trabajo).

- Tras visualizarlos en gráficos, y dado a los clientes que brindamos nuestro servicio, curamos el dataset con los siguientes ítems:
    - Limitamos la edad máxima a 80 años, dado que no nos interesan clientes mayores.
    - Colocamos un ingreso máximo de 800.000 USD anuales, avisando a los clientes que contraten nuestros servicios, que deberían otorgar un status premium (préstamo casi automático, se debería chequear su destino y su situación global financiera) a aquellos potenciales clientes que se presenten con ingresos demostrables mayor a este número.
    - Finalmente, para la antigüedad de una persona en su trabajo, dejamos un máximo de 30 años.

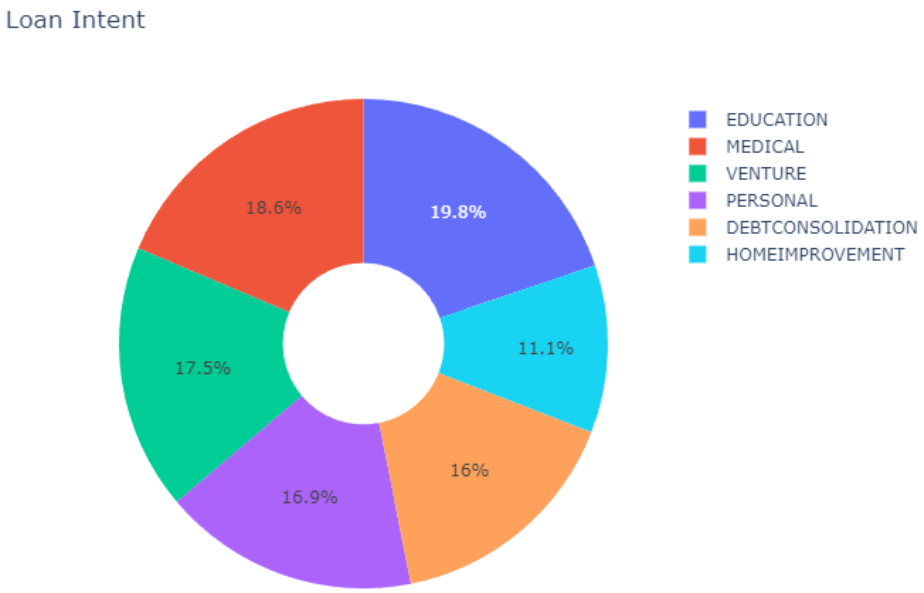
Análisis Exploratorio de los Datos (EDA) (Univariado - Bivariado - Multivariado)

Una vez que tenemos la base de datos limpia y curada, realizamos un segundo análisis de los datos, con foco en nuestra variable target (loan_status) dado que nuestro objetivo será conocer si un crédito entrará en default o no.

Conocemos el estado de los créditos y los graficamos:

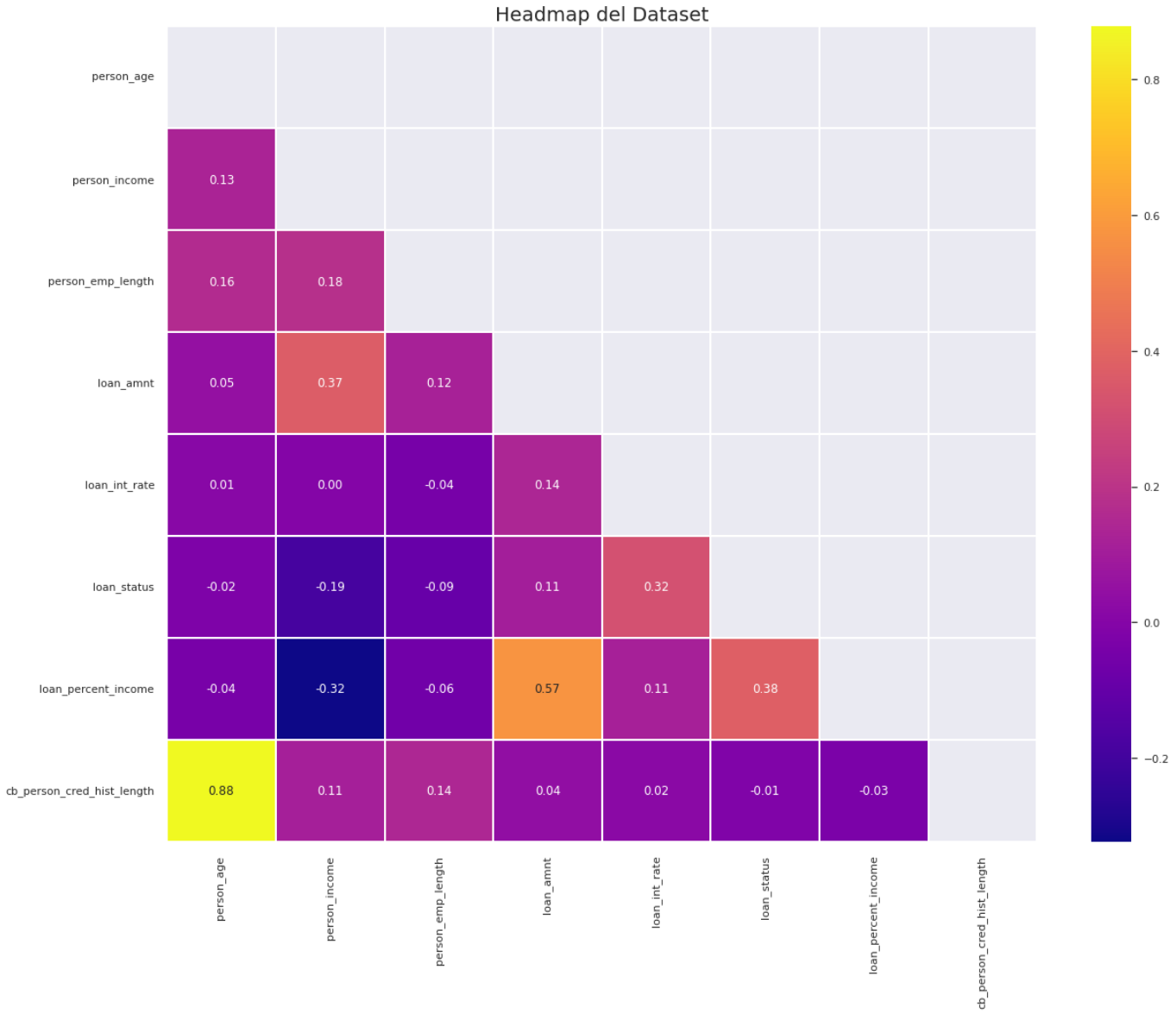


Graficamos el destino de los créditos:



Nuestro principal objetivo es conocer si un crédito entrará en default o no, así que veremos las principales correlaciones con el status del crédito y lo graficamos:

```
loan_status                   1.000000
loan_percent_income           0.379115
loan_int_rate                 0.319328
loan_amnt                     0.105439
cb_person_cred_hist_length   -0.014710
person_age                   -0.019749
person_emp_length            -0.090523
person_income                -0.188815
```



Realizamos distintos calculos exploratorios

Media de edad de los clientes: 27.701404036990382

Monto medio del crédito: 9586.811422777966

Porcentaje de créditos por calidad del mismo:

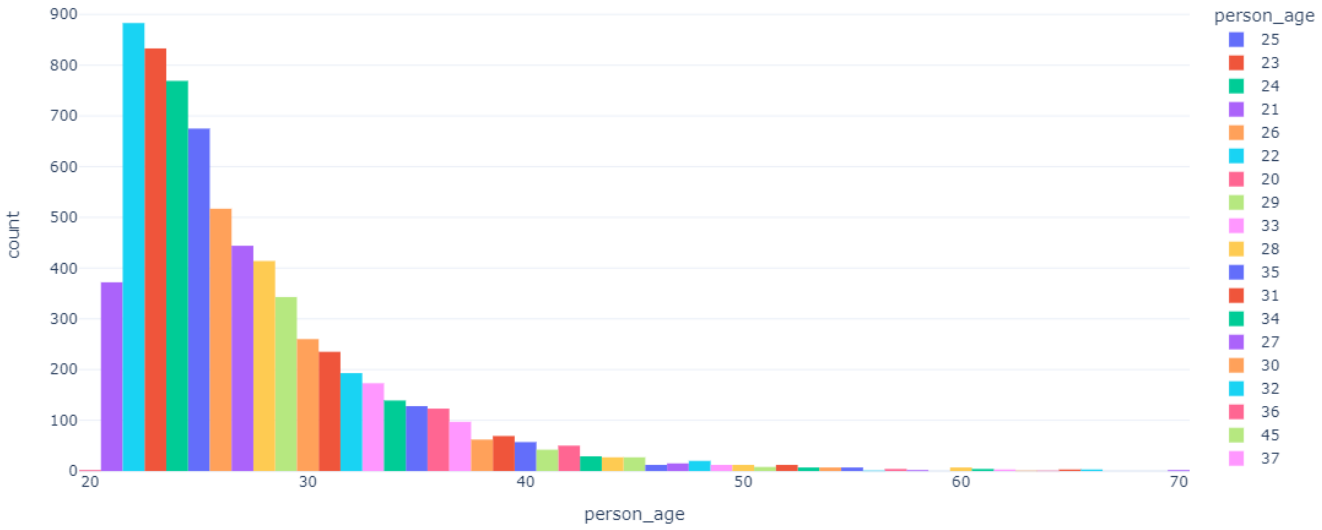
```
A    0.330794
B    0.320716
C    0.198193
D    0.111309
E    0.029617
F    0.007404
G    0.001966
```

Mostramos si los clientes son dueños de su casa, alquilan o tienen una hipoteca:

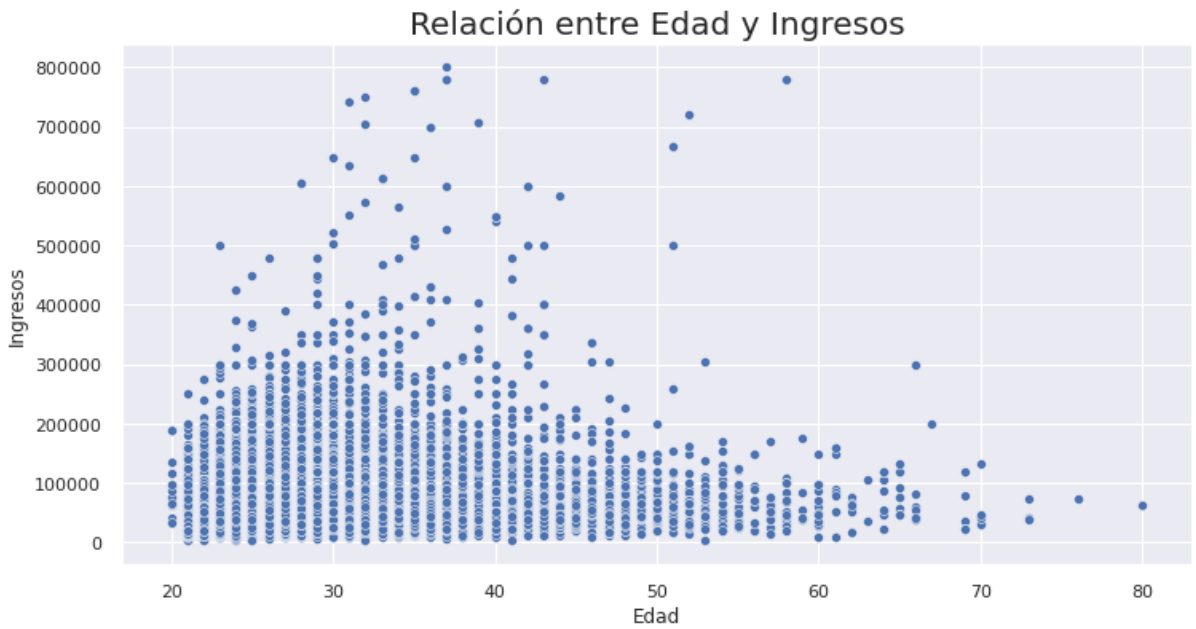
```
RENT        0.504962
MORTGAGE    0.412424
OWN         0.079327
OTHER       0.003287
```

Graficamos donde gastan el dinero obtenido aquellos clientes que no están al día (deudores):

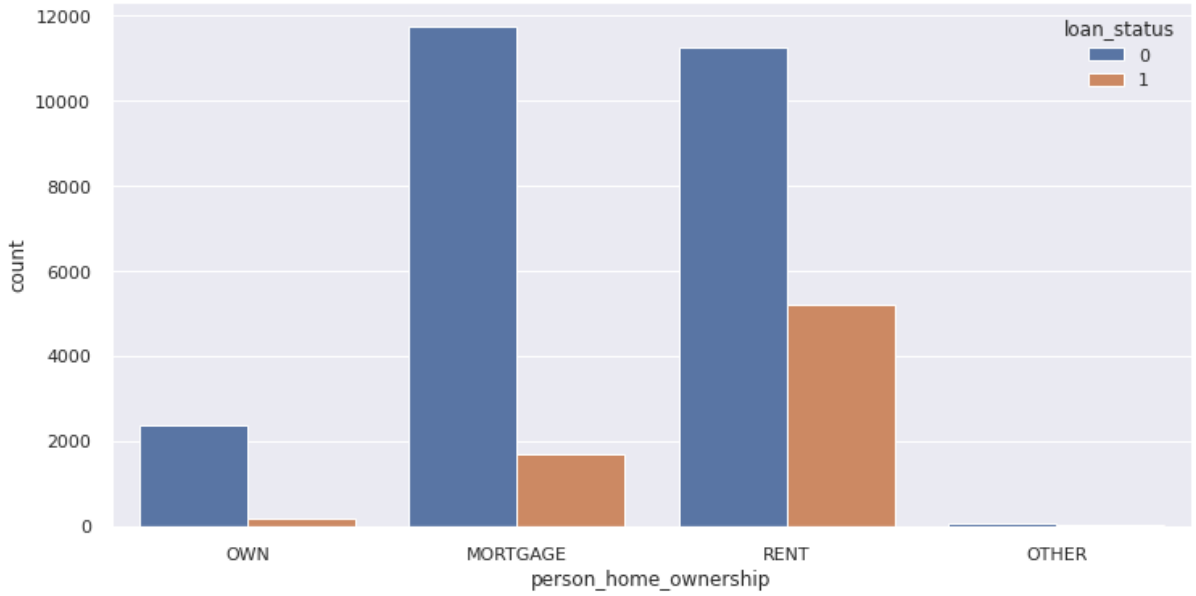
Observamos que la mayor cantidad de deudores es gente menor a 30 años



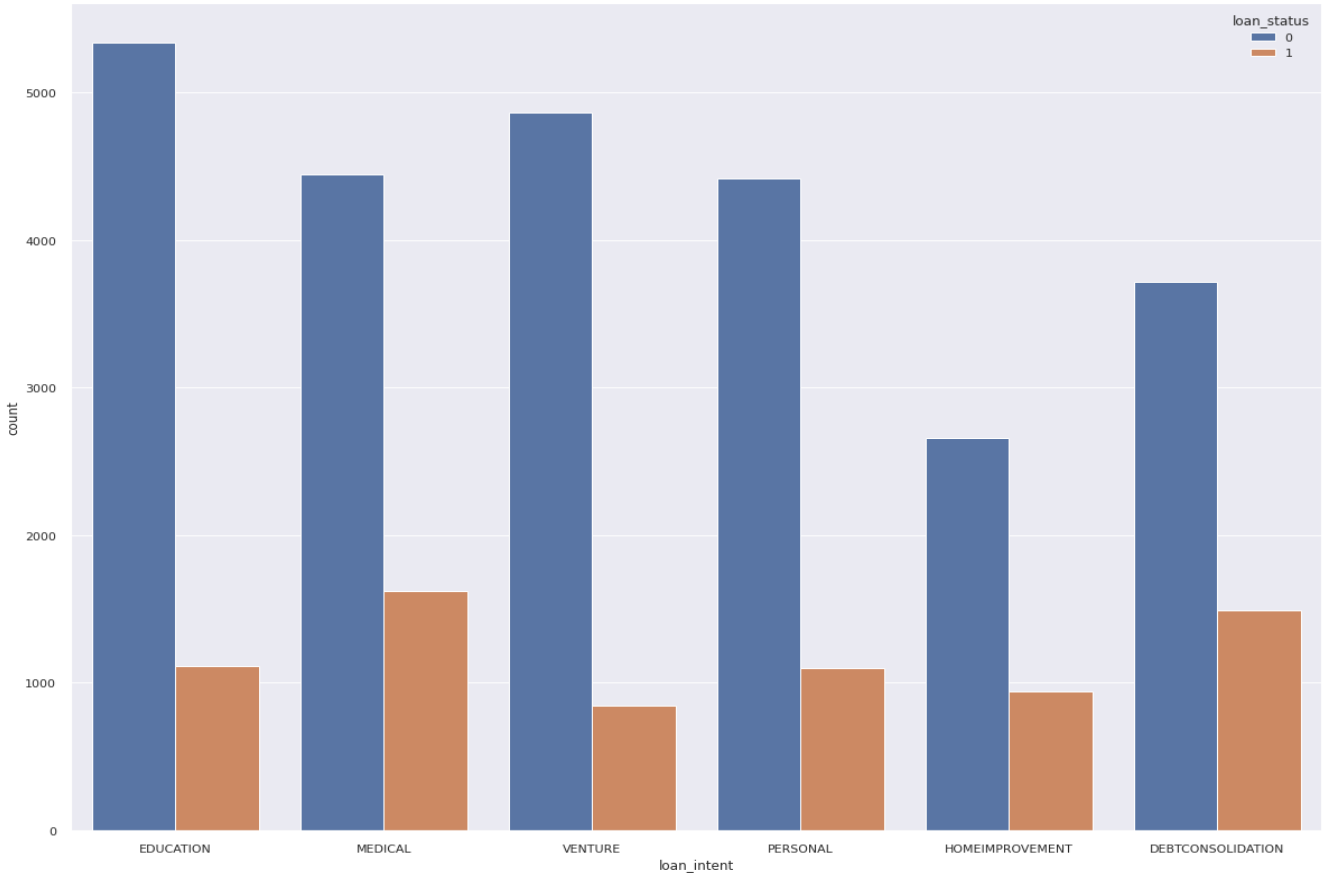
Graficamos la relación entre la edad y los ingresos:



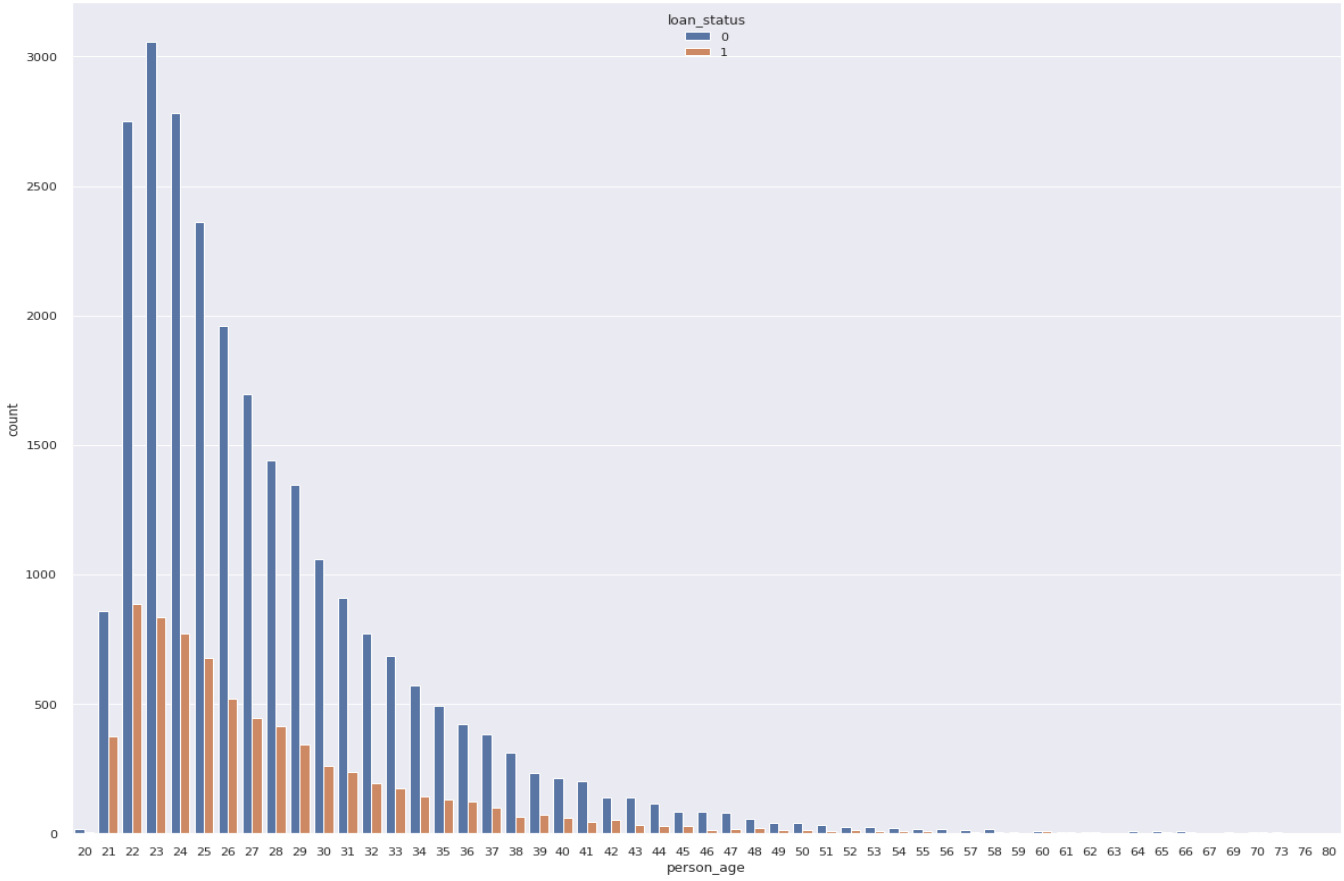
Vemos la relación entre el status de crédito y la propiedad de sus viviendas (dueños, hipotecas, alquila y otros):



Ahora vemos la relación del status de crédito respecto al destino gastado:



Continuamos relacionando, en este caso el status de crédito con la edad:



Conclusiones del EDA

Tras los distintos tipos de análisis aplicados (Univariado, Bivariado y Multivariado) aprendidos durante el curso, llegamos a las siguientes conclusiones sobre el mercado de créditos actual:

- Aproximadamente, el 78% de las personas mantiene sus créditos al día y el 22% presenta algún tipo de deuda (default).

- El destino de los créditos es bastante parejo, pero se destacan principalmente educación (20%), cuestiones médicas (19%) y emprendimiento (17.5%).

- Las variables person_income, person_emp_length, y person_age tienen correlación negativa en el estado del crédito, lo que significa que mientras mayores sean los ingresos, la duración en el empleo y la edad de la persona, menos riesgoso será otorgarle un préstamo al cliente.

- Las variables loan_percent_income, loan_int_rate, y loan_amnt tienen correlación positiva en el estado del crédito, lo que significa que mientras mayores sean dichas variables, más posibilidades de caer en default tiene el crédito.

- La edad media de los clientes es 27-28 años, lo cual tiene lógica dado que el mayor destino de los préstamos es la educación.

- El monto medio de los créditos es de 9587 USD.

- Considerando A,B como créditos de calidad alta, C,D calidad media y E,F,G calidad baja, tenemos 65% en calidad alta, 31% en calidad media y 4% en calidad baja.

- Más del 50% de nuestros clientes alquila su vivienda. Un 41% está pagando una hipoteca, y solo un 8% es dueño pleno de su hogar.

- Separamos a los clientes en deudores y no deudores para tener una mayor claridad sobre cómo actúan cada uno.

- Los clientes deudores destinan sus préstamos principalmente a cuestiones médicas y a consolidación de deuda. Puede ser que sean clientes sin seguro médico y ante un accidente, piden dinero prestado.

- La mayor cantidad de clientes deudores son jóvenes. Tiene sentido dado que hay una tendencia donde a mayor edad, mayor ingreso anual.

- Aquellos clientes que alquilan su vivienda o pagan una hipoteca, tienen mayor presencia en la situación de clientes deudores. Tiene lógica dado que aquellas personas que son dueñas de su hogar, no tienen un gasto importante en alquiler o hipoteca.

- Los clientes propietarios de su hogar, son excelentes cumplidores de sus préstamos.

- Los clientes cuyo destino es emprendimiento presenta el % de default más bajo (solo 14% de aquellos que solicitan préstamo para emprender no cumple con sus créditos en tiempo y forma).

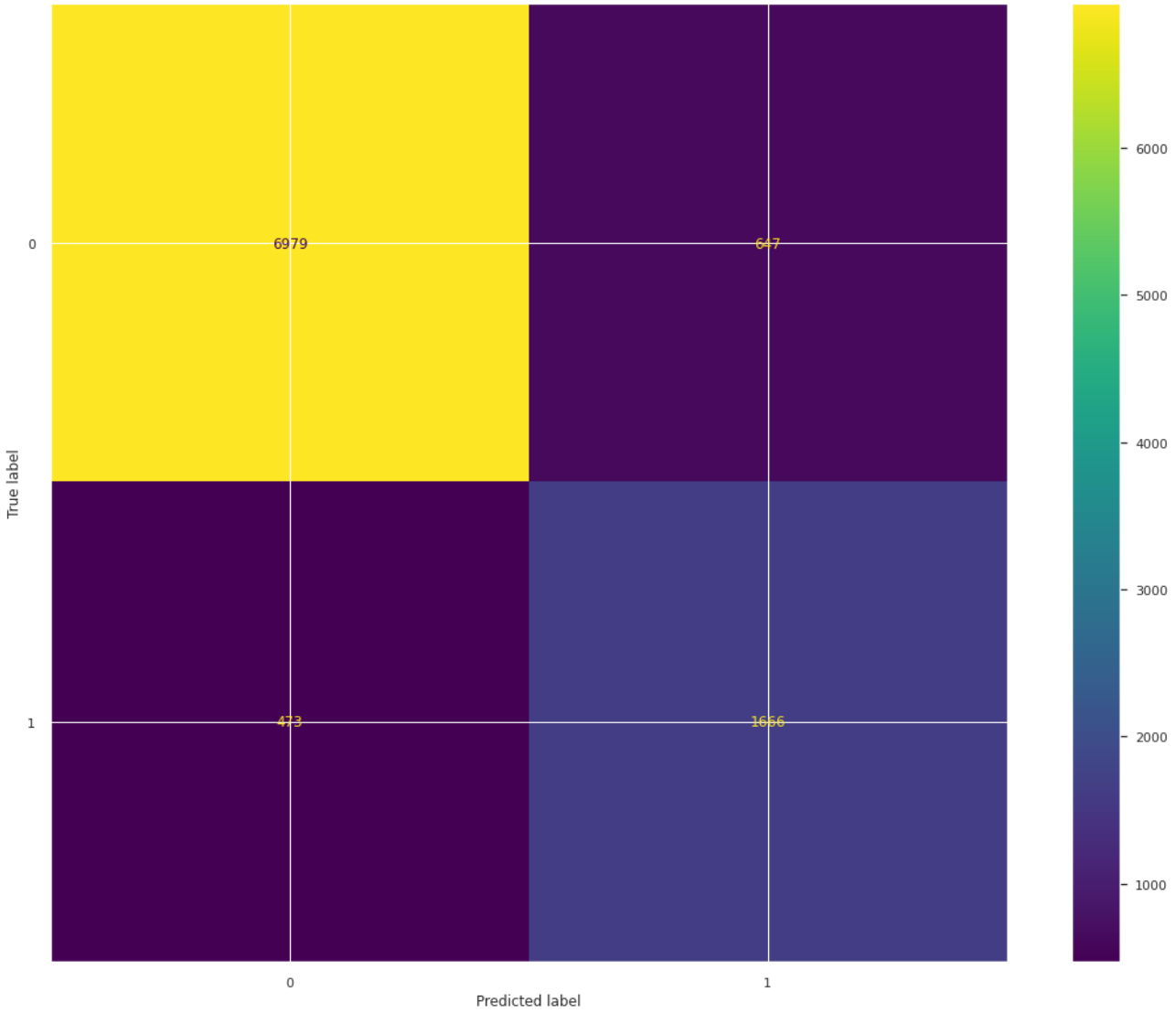
- Los clientes cuyo destino de préstamo es consolidación de deuda son los más propensos a ingresar en default (El 26% de los clientes que solicitan créditos para consolidación de deuda ingresa en default).

Modelos de Machine Learning

Modelo Árbol de Decisión

Los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones, es uno de los algoritmos de aprendizaje supervisado más utilizados en machine learning. La comprensión de su funcionamiento suele ser simple y a la vez muy potente. Los árboles de decisión tienen un primer nodo llamado raíz (root) y luego se descomponen el resto de atributos de entrada en dos ramas, planteando una condición que puede ser cierta o falsa. Se bifurca cada nodo en 2 y vuelven a subdividirse hasta llegar a las hojas que son los nodos finales y que equivalen a respuestas a la solución: En nuestro dataset será, 0 o 1 (non default or default).

Luego de crear y entrenar el modelo, ploteamos la exactitud del train y el test:



Hay cuatro términos importantes en una matriz de confusión:

- Verdaderos positivos (TP): estos son los casos en los que el "Sí" predicho en realidad pertenecía a la clase "Sí". 6979 datos

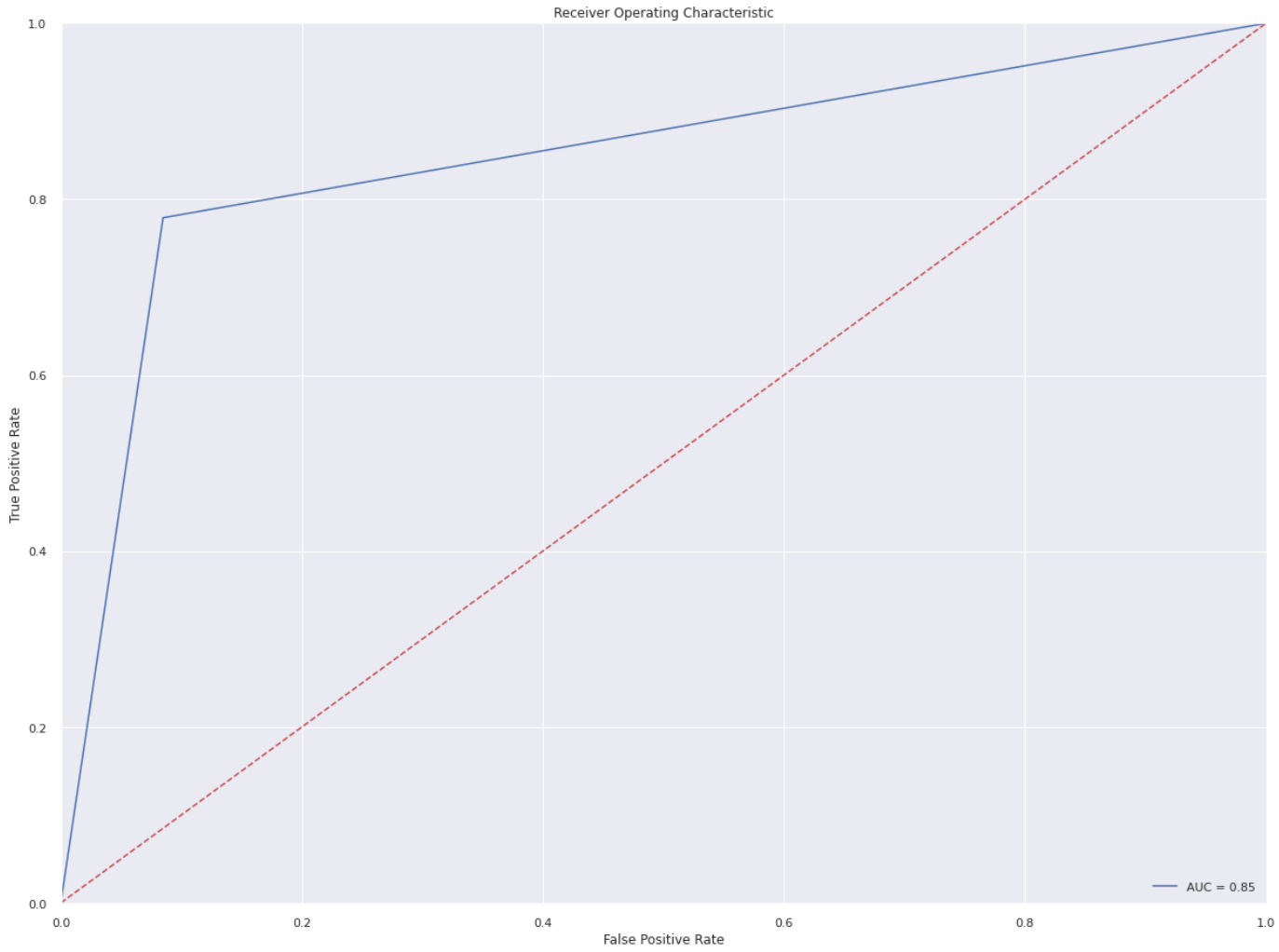
- Negativos verdaderos (TN): Estos son los casos en los que el “No” predicho en realidad pertenecía a la clase “No”. 1656 datos

- Falsos positivos (FP): estos son los casos en los que el "Sí" predicho en realidad pertenecía a la clase "No". Error tipo 1. 647 datos

- Falsos negativos (FN): estos son los casos en los que el "No" predicho en realidad pertenecía a la clase "Sí". Error tipo 2. 473 datos

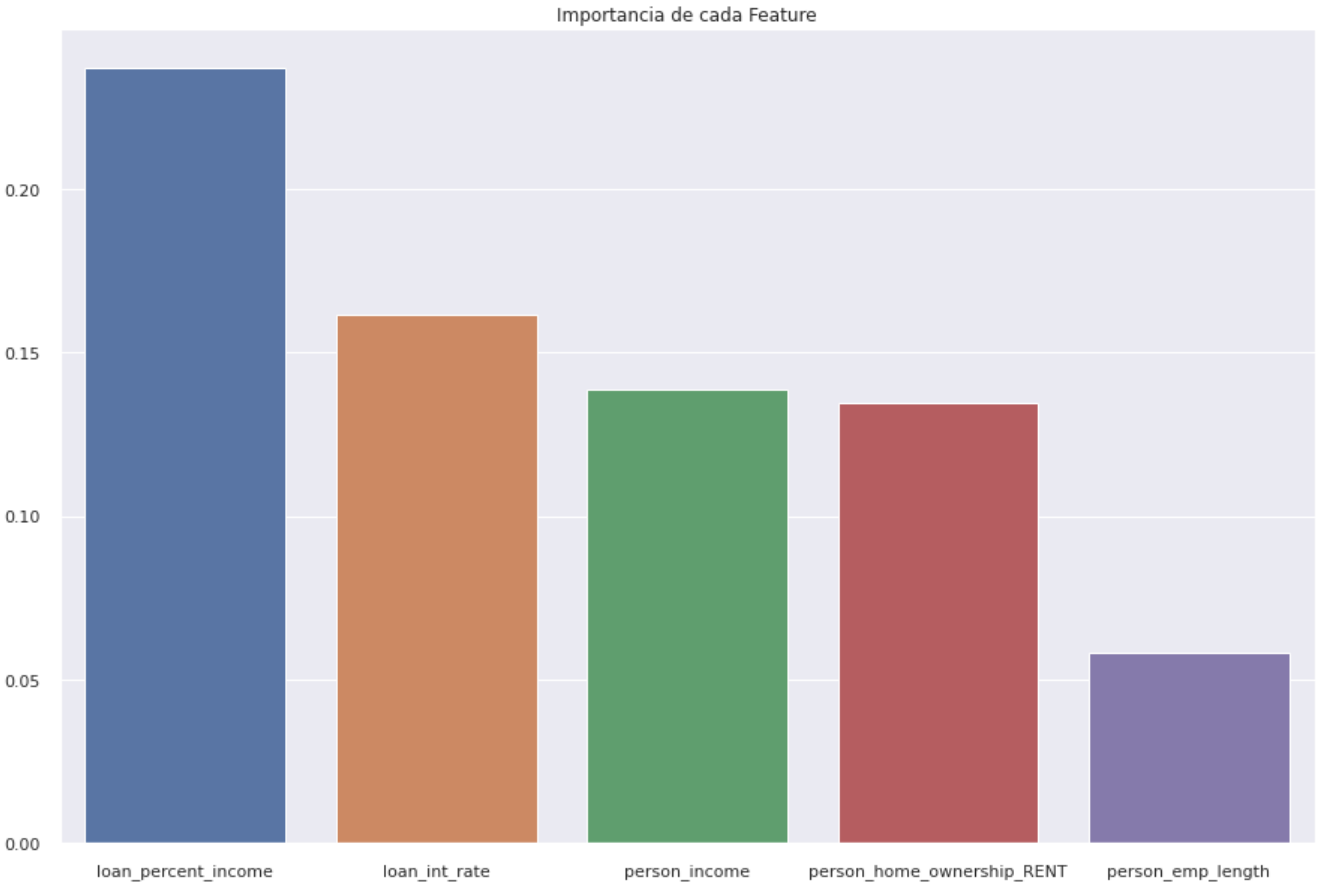
Evaluamos el modelo con los siguientes resultados y lo visualizamos con la curva ROC:

- ➢ Accuracy: 0.8853
- ➢ Precisión: 0.7203
- ➢ F1 Score: 0.7484
- ➢ Recall: 0.7789



Buscamos las variables que más influyen en el modelo y las visualizamos

Variables	Importancia
loan_percent_income	0.23712
loan_int_rate	0.16177
person_income	0.13884
person_home_ownership_RENT	0.13464
person_emp_length	0.05831



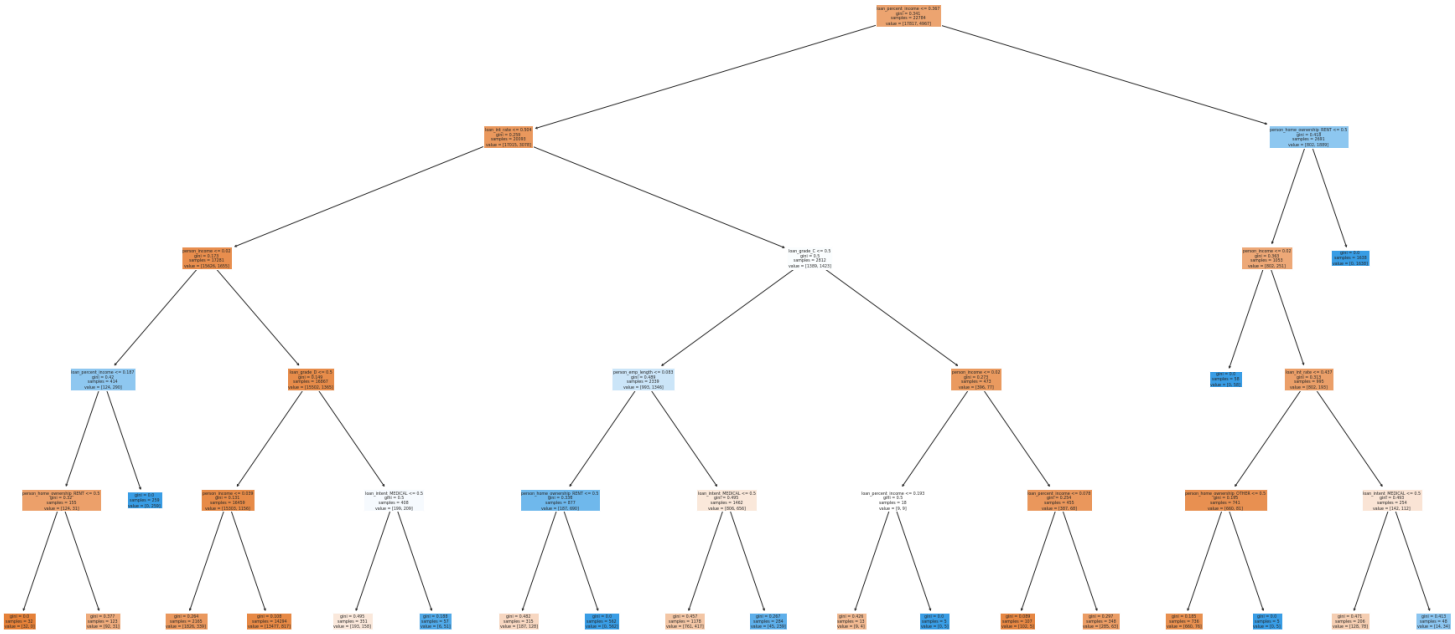
Probamos con una profundidad de 5, obteniendo los siguientes valores de exactitud para el test y train:

- % de aciertos sobre el set de entrenamiento: 0.9042749297752809
- % de aciertos sobre el set de evaluación: 0.907731694828469

Y los siguientes resultados en la evaluación del modelo:

- Accuracy: 0.9077
- Precisión: 0.9806
- F1 Score: 0.7371
- Recall: 0.5905

Visualizamos el arbol de desicion



Modelo KNN Vecinos más cercanos

K-Nearest-Neighbor es un algoritmo basado en instancia de tipo supervisado de Machine Learning. Sirve esencialmente para clasificar valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento y haciendo conjeturas de nuevos puntos basado en esa clasificación.

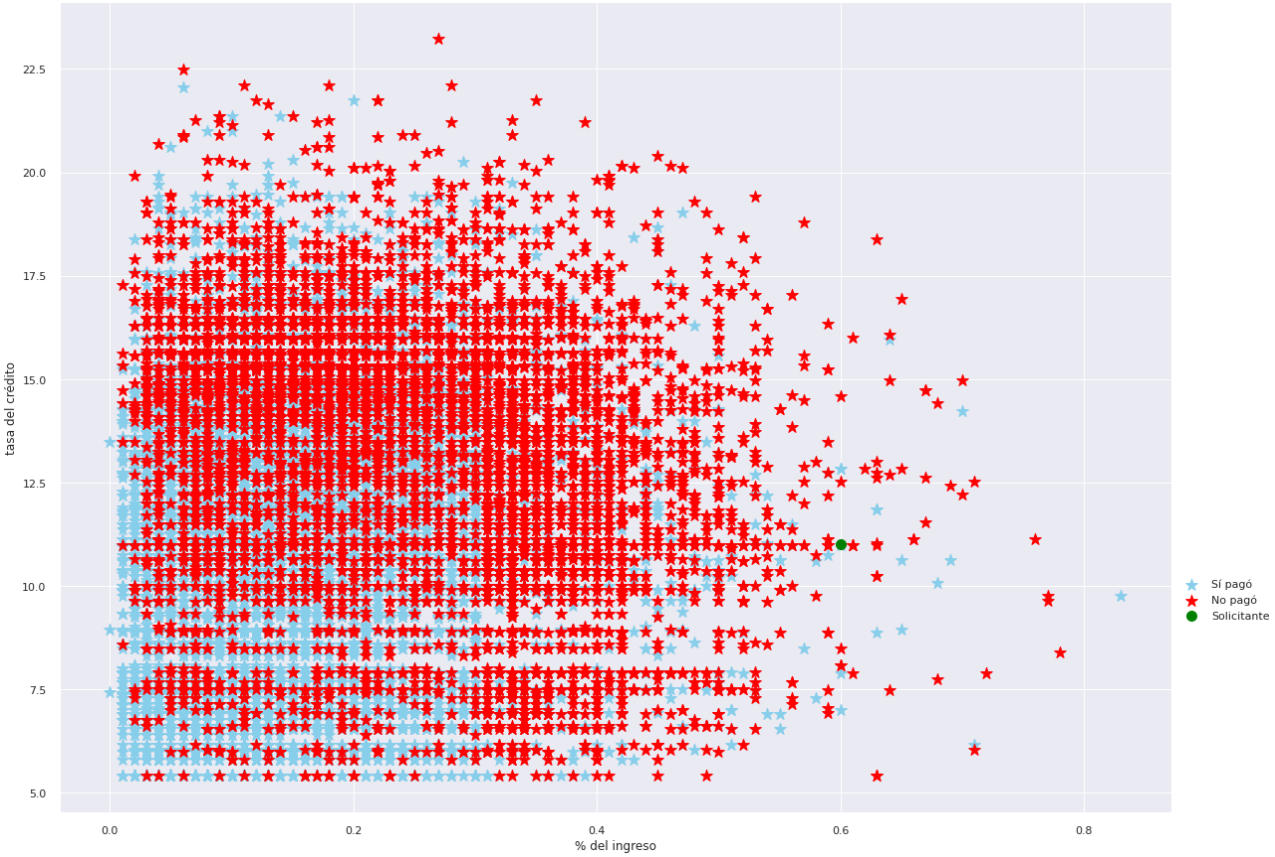
Obtenemos los siguientes resultados con 5 “vecinos”:

- Accuracy: 0.8922
- Precision: 0.8499
- F1 Score: 0.7147
- Recall: 0.6166

Predicción

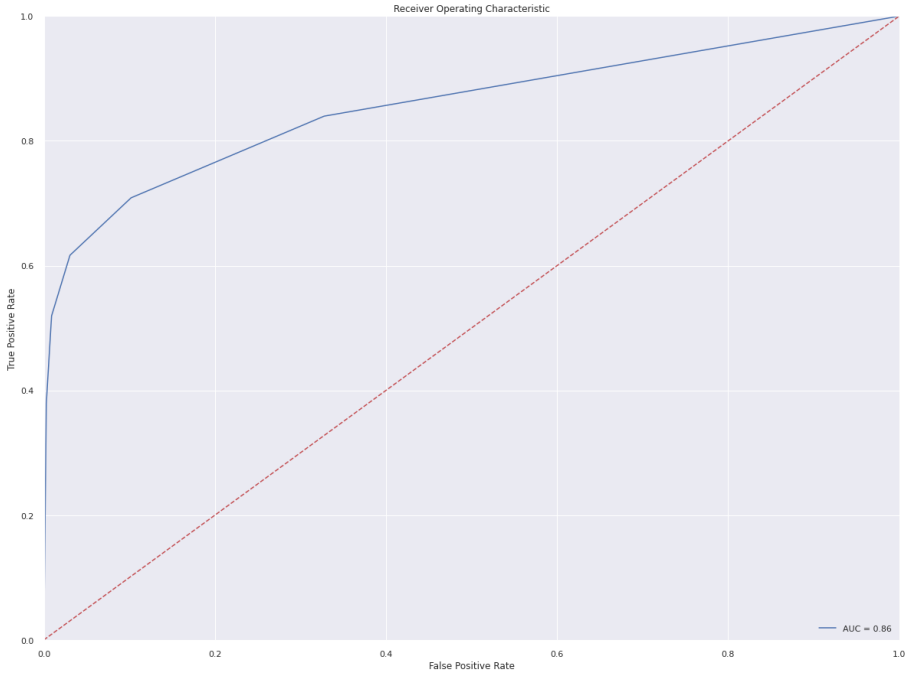
En base a los features más importantes, *loan_percent_income* y *loan_int_rate*, armamos un predictor.

Ejemplo de predicción: se acerca un cliente a solicitar un crédito, nos dice el monto a pedir, vemos sus ingresos, llegamos a que es el 60% de sus ingresos, le aplicamos la tasa vigente del 11%, escalamos los datos y lo graficamos para ver dónde se ubicaría:



Calculamos la clase y probabilidades, obteniendo un 20% de no caer en default y un 80% de entrar en en el.

Utilizamos nuevamente la curva ROC para visualizar los resultados del modelo:



Modelo Random Forest

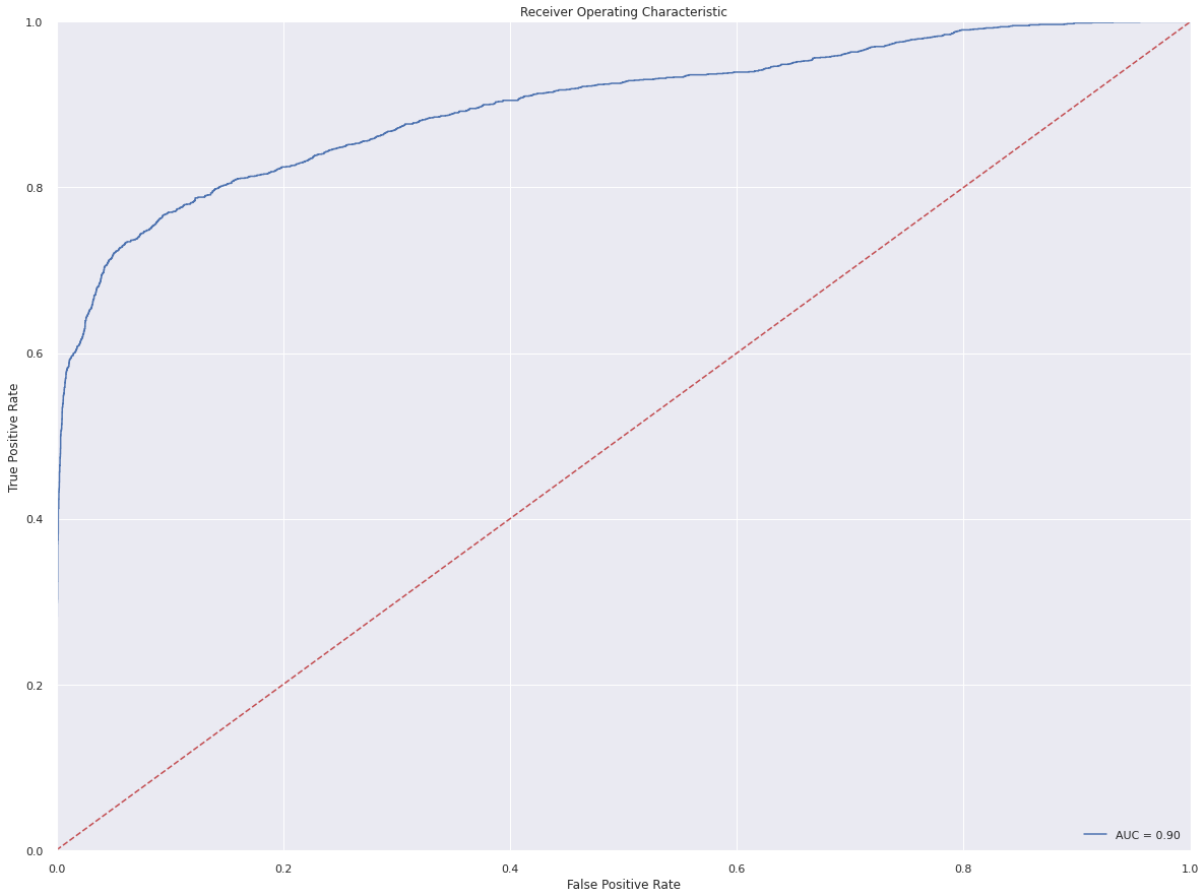
Random Forest es un modelo de aprendizaje supervisado para clasificación. Es un tipo de Ensamble en Machine Learning en donde combinaremos diversos árboles y la salida de cada uno se contará como “un voto” y la opción más votada será la respuesta del 'Bosque Aleatorio'.

Uno de los problemas que aparecía con la creación de un árbol de decisión es que si le damos la profundidad suficiente, el árbol tiende a “memorizar” las soluciones en vez de generalizar el aprendizaje. Es decir, a padecer de overfitting. La solución para evitar esto es la de crear muchos árboles y que trabajen en conjunto.

Tras crear un arbol de decision sencillo, evaluamos el modelo con los siguientes resultados:

- Accuracy: 0.8985
- Precision: 0.9541
- F1 Score: 0.7088
- Recall: 0.5638

Curva ROC para visualizar resultados:

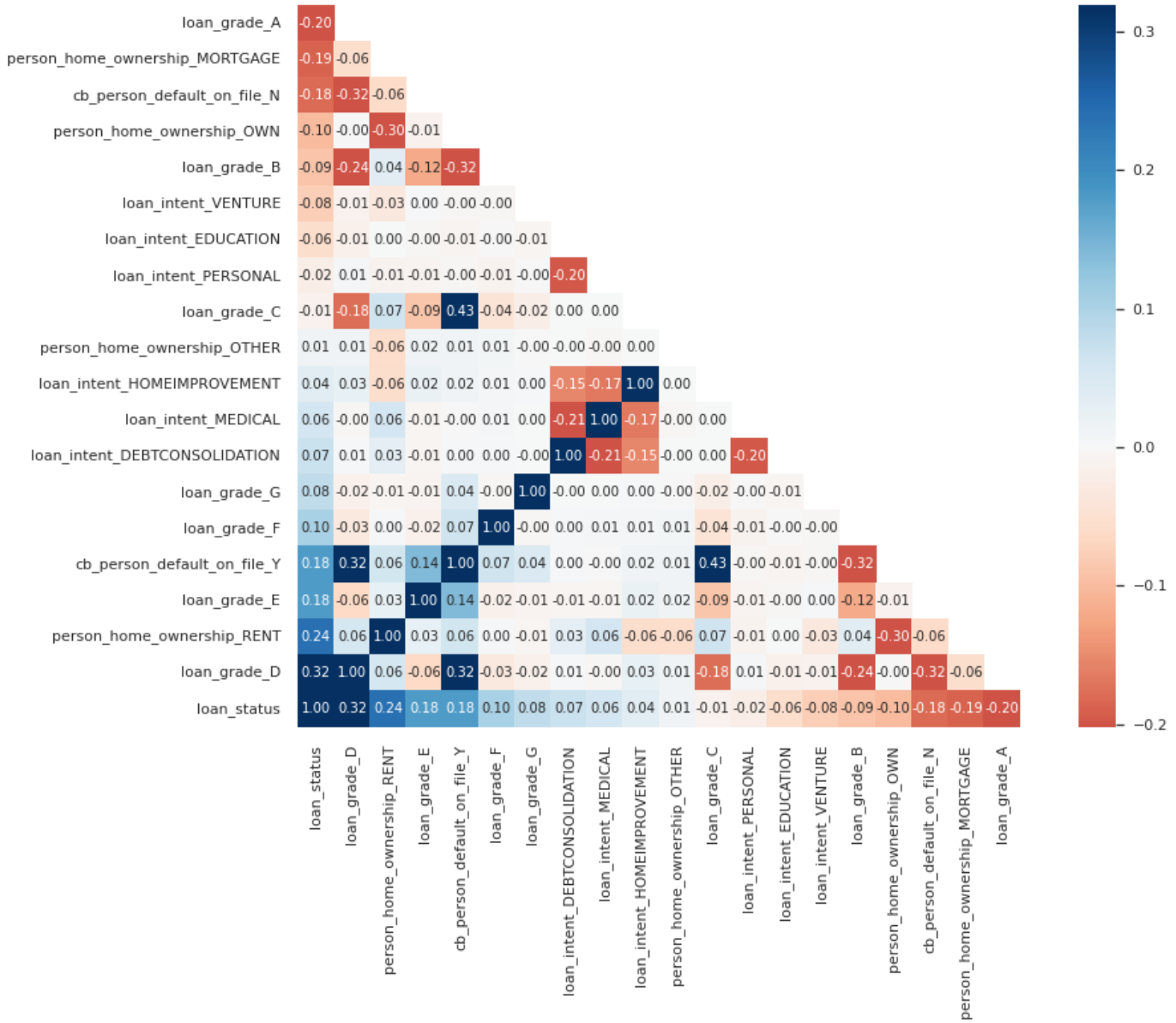


Modelo Regresión Logística

A pesar de su nombre, no es un algoritmo de regresión en el que se prediga un valor continuo, sino que es un método para problemas de clasificación binaria. El método de regresión logística es un método estadístico que se usa para resolver problemas de clasificación binaria, donde el resultado solo puede ser de naturaleza dicotómica, o sea, solo puede tomar dos valores posibles. Por ejemplo, se puede utilizar para detectar la probabilidad de que ocurra un evento.

En nuestro dataset, el evento será si el crédito caerá en default o no.

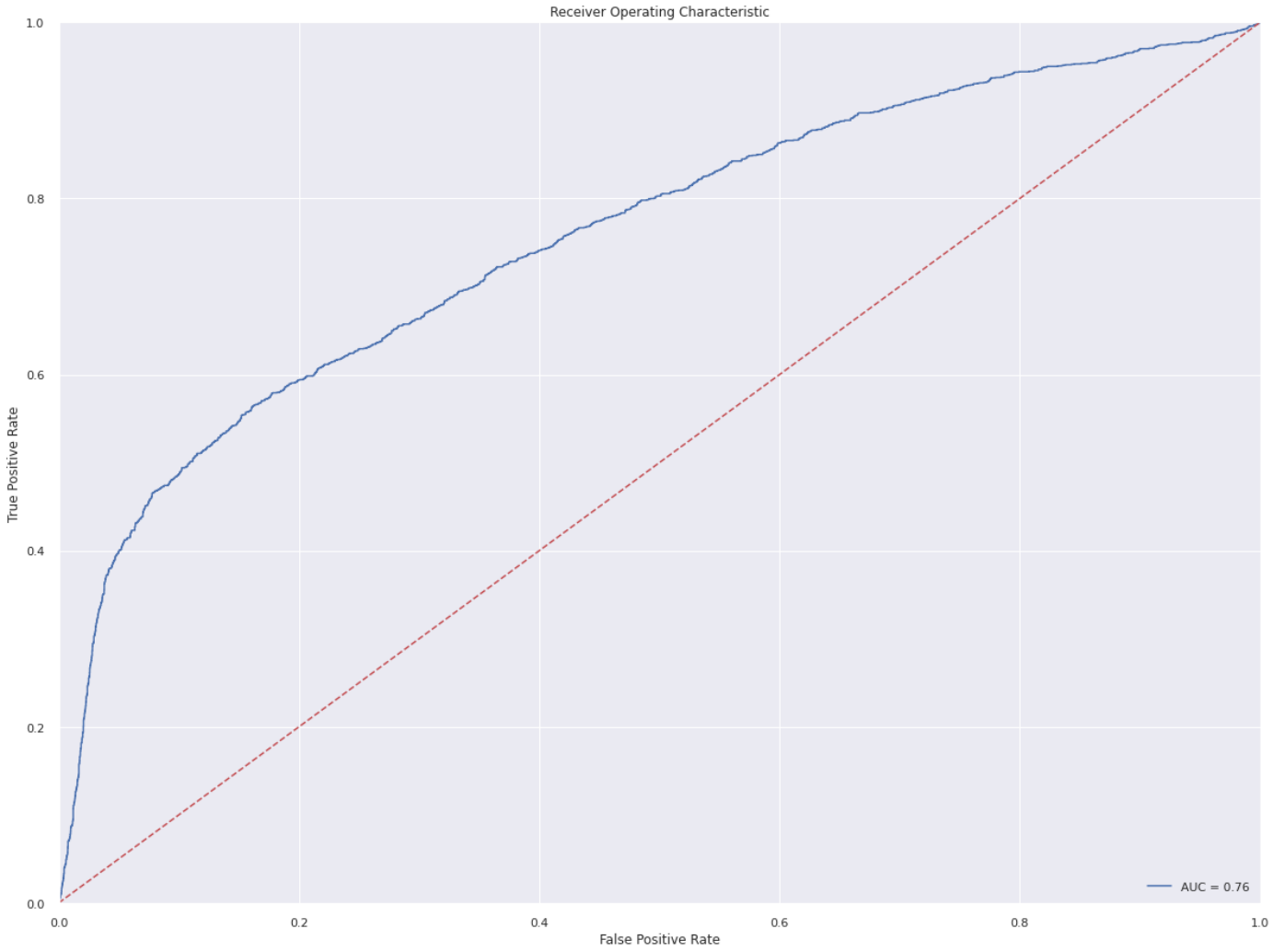
Codificamos las variables categóricas con el método One Hot Encoding:



Evaluamos el modelo y obtenemos los siguientes resultados:

- Accuracy predict: 80.36%
- Precision: 72.69%
- Recall: 16.55%
- F1_score: 26.96%

Curva ROC para visualización:



Comparación de Métricas de los 4 Modelos Utilizados

Creamos una tabla para mostrar la comparativa entre las métricas de cada modelo utilizado:

	accuracy	precision	f1 score	recall
Arbol de Decision	0.90770	0.98060	0.73710	0.59050
KNN Vecinos	0.89220	0.84990	0.71470	0.61660
Random Forest	0.89850	0.95410	0.70880	0.56380
Regresion Logistica	0.80358	0.72690	0.26961	0.16550

Conclusiones Parciales: Basando en los resultados, el algoritmo con mejor cantidad de aciertos es el árbol de decisión clásico (con profundidad de 5 niveles). Observamos como KNN vecinos más cercanos y random forest tienen un accuracy similar, por lo que son buenas alternativas. Regresión logística presenta un accuracy de 80%, bastante por debajo de los otros tres modelos. En tanto que en Precisión, qué es el número de elementos identificados correctamente como positivo de un total de elementos identificados como positivos. Árbol de Decisión y Random Forest tienen las mejores métricas.

Una vez asegurados que el mejor algoritmo para nuestro modelo sería el de Árbol de Decisión se proseguirá con la aplicación de algoritmos de optimización dentro del modelo.

Mejora a los Modelos de Machine Learning

KFOLD

Este proceso se repite *k* veces utilizando un grupo distinto como validación en cada iteración. Finalmente obtenemos criterios de evaluación donde tomamos un promedio. Resultados:

- Métrica del modelo 0.9042749297752809
- Metricas cross_validation [0.90542023 0.90037305 0.8975203 0.89817863 0.91022827]
- Media de cross_validation 0.9023440960399567
- Métrica en Test 0.907731694828469
- Accuracy: 0.9077
- Precision: 0.9806
- F1 Score: 0.7371
- Recall: 0.5905

La aplicación de KFOLD nos otorga ventajas respecto a los modelos previos ya que todos sus índices son superiores.

Random Grid Search

El método más común de ajuste de hiperparámetros es grid search. Este método crea una cuadrícula (*grid*) de ajuste con combinaciones únicas de valores de hiperparámetros y utiliza cross validation (*validación cruzada*) para evaluar su rendimiento. El objetivo del ajuste de hiperparámetros es encontrar la combinación óptima de valores para maximizar el rendimiento del modelo. El *random grid search* prueba combinaciones de valores al azar que se le proporcione en el grid de parámetros.

Obtenemos los siguientes resultados:

- Accuracy: 0.9208
- Precision: 0.9499
- F1 Score: 0.7886
- Recall: 0.6741

Este método nos aporta una mejora significativa respecto a lo obtenido previamente, aumentando todos los indicadores. Un accuracy del 92,08% es un buen candidato.

Modelos de Ensamble/Boosting

XGBoost

Implementa modelos de predicciones débiles, con el objetivo de que secuencialmente cada modelo débil le permita ir ganando más poder predictivo hasta llegar a un modelo más robusto con mayor estabilidad en sus resultados.

Obtenemos los siguientes resultados:

- ➢ Porcentaje de aciertos sobre el set de entrenamiento: 0.8828125
- ➢ Porcentaje de aciertos sobre el set de evaluación: 0.8794674859190988

Cuya evaluación arroja los valores a continuación:

- ➢ Accuracy: 0.9224
- ➢ Precision: 0.9297
- ➢ F1 Score: 0.7977
- ➢ Recall: 0.6985

Adaboost

Trabaja sobre una tasa de error, donde el primer modelo es un aprendiz débil, el segundo modelo será menos débil y así sucesivamente, hasta llegar a una tasa de error que sea próxima a 0.

Evaluación del modelo:

- ➢ Accuracy: 0.8857
- ➢ Precision: 0.7908
- ➢ F1 Score: 0.7137
- ➢ Recall: 0.6503

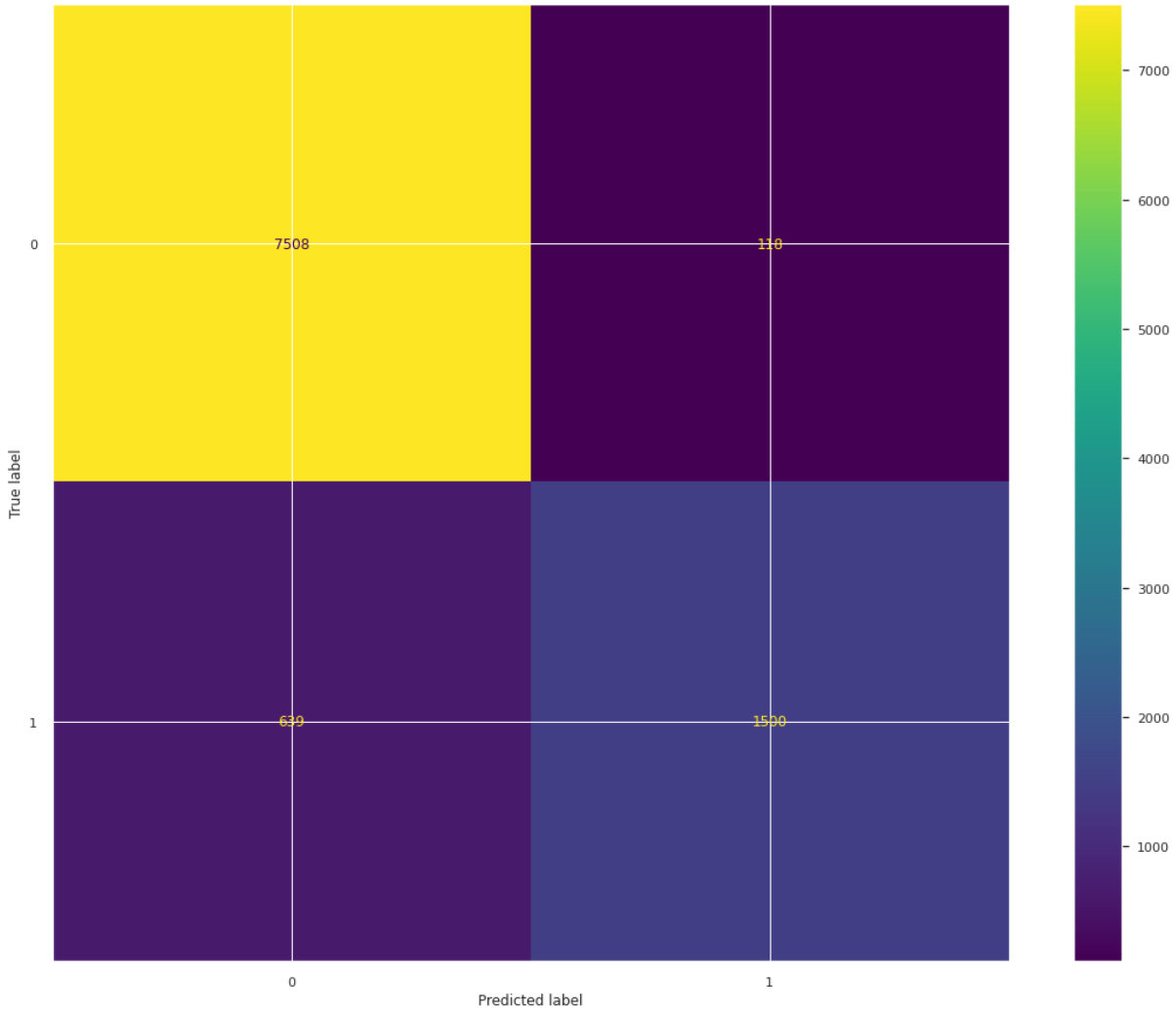
Gradient Boosting

Se basa en un principio de intuición. Implementa el concepto de aleatoriedad. En la iteración 1 entrena, en la iteración 2 adopta ese principio de intuición/aleatoriedad y luego en la iteración 3 lo hará nuevamente y así sucesivamente. Este factor permitirá darle al modelo una minimización del error de predicción general. Ese factor de aleatoriedad que se va agregando a las iteraciones, es lo que se conoce como Gradiente.

Resultados:

- Accuracy: 0.9225
- Precision: 0.9271
- F1 Score: 0.7985
- Recall: 0.7013

Debido al alto resultado, ploteamos la Matriz:

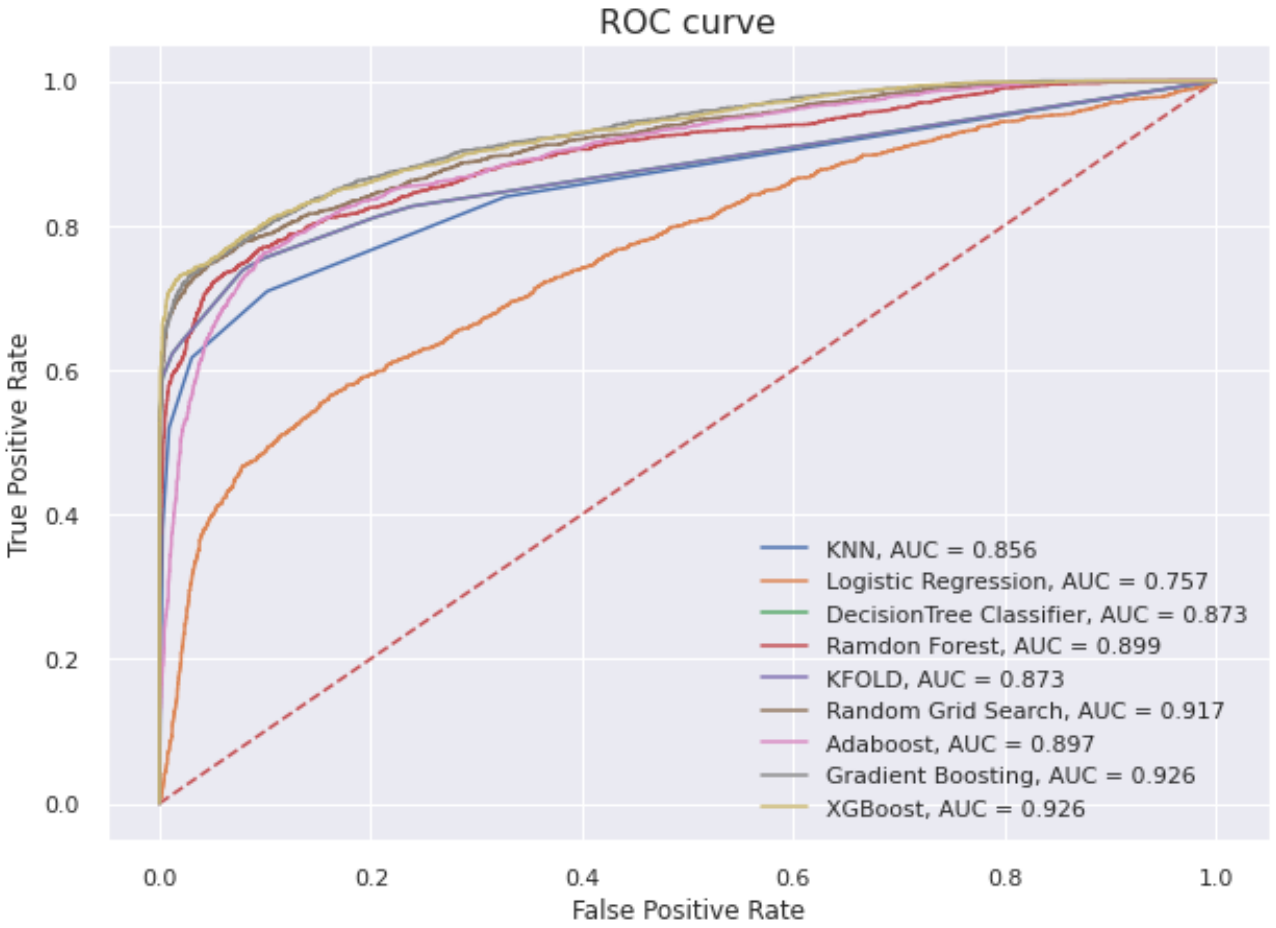


Comparación de Todos los Modelos

Generamos una tabla que muestra la comparativa entre las métricas de cada modelo utilizado:

	accuracy	precision	f1 score	recall
Arbol de Decision	0.90770	0.98060	0.73710	0.59050
KNN Vecinos	0.89220	0.84990	0.71470	0.61660
Random Forest	0.89850	0.95410	0.70880	0.56380
Regresion Logistica	0.80358	0.72690	0.26961	0.16550
KFOLD	0.90770	0.98060	0.73710	0.59050
Random Grid Search	0.92080	0.94990	0.78860	0.67410
XGBoost	0.92760	0.94200	0.81190	0.71340
Adaboost	0.88570	0.79080	0.71370	0.65030
Gradient Boosting	0.92250	0.92710	0.79850	0.70130

Por último, generamos una curva ROC con todos los modelos para compararlos visualmente:



Conclusiones Finales

Antes que nada, veamos la definición de las métricas principales que estamos usando para comparar:

Accuracy (exactitud): mide el porcentaje de casos que el modelo ha acertado. Es la medida más directa de la calidad de los clasificadores, los valores se encuentran entre 0 y 1, y mientras más alto mejor. Es una métrica de la cual no se debe confiar plenamente, ya que puede ocasionar problemas si las clases de variables de destino en los datos no están balanceadas.

Precision (Precisión): sirve para medir la calidad del modelo de machine learning en tareas de clasificación. Identifica qué porcentaje de valores que se han clasificado como positivos son realmente positivos.

Recall (Exhaustividad): nos va a informar sobre la cantidad que el modelo de machine learning es capaz de identificar.

F1 score (puntaje F1): se utiliza para combinar las medidas de *precision* y *recall* en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones.

Elección del modelo ganador

Ante los datos comparados, tenemos 3 candidatos con las mejores métricas:

Modelo	Exactitud	Precisión	Puntaje F1	Exhaustividad	Promedio
Random Grid Search	0.92080	0.94990	0.78860	0.67410	0.83335
XGBoost	0.92760	0.94200	0.81190	0.71340	0.848725
Gradient Boosting	0.92250	0.92710	0.79850	0.70130	0.83735

Pese a que en precisión es superior al modelo Random Forest con la aplicación del Random Grid Search, el modelo XGBoost es superior al resto en exactitud (*accuracy*), puntaje F1 (*F1 score*), y en exhaustividad (*recall*), dando un promedio superior al resto. Esto convierte al modelo de mejoras de machine learning XGBoost en la mejor opción para resolver nuestro problema inicial, la clasificación en default o no default. Además destacamos su velocidad de aplicación sobre el Random Forest con Random Grid Search.

Conclusiones Generales

El banco o institución financiera que comprenda la importancia del uso en su día a día de Data Science, será capaz de adelantarse a la competencia y ver incrementado sus beneficios. En este caso, gracias al análisis exploratorio de los datos y la aplicación de modelos de machine learning, las entidades podrán determinar con alta precisión, en base a pocos datos del cliente, la posibilidad de que un futuro préstamo sea cobrado correctamente o no. La determinación de posibles deudores, genera significativos ahorros en pérdidas por deudores incobrables y publicidad mal direccionada.

Conclusiones personales

Hemos recorrido un largo camino de aprendizaje y superación. La adquisición de conocimientos ha sido increíble y de gran ayuda en nuestra formación profesional. Queremos agradecer al profesor Octavio por sus excelentes y didácticas clases, y a nuestro tutor Juan, que estuvo para ayudarnos siempre, en todo lo que fuese necesario. Esperemos que la culminación de esta carrera, sea la puerta de entrada para futuras oportunidades laborales en el mundo de Data Science.