

# Modelo Previsión Ventas Productos Financieros

# Memoria

Trabajo Final Master Data Science

Manuel Gonzalez Prados

TFMDS 2020 - 2021

## TABLA DE CONTENIDO

### 1 INTRODUCCION

- 1.1. Motivación
- 1.2. Prologo
- 1.3. Objetivos
- 1.4. Finalidad
- 1.5. Interés

### 2. ESTADO DEL ARTE

- 2.1. Marco
- 2.2. Seguros en España- Cifras
- 2.3. Tipos de seguros contratados en España. Evolución anual
- 2.4. Tipos de servicio del seguro en España
- 2.5. Nivel de contratación de seguros por edades
- 2.6. Contratación de seguros por ingresos del hogar
- 2.7. Niveles de contratación del seguro de hogar en Europa
- 2.8. Conclusiones

### 3. CASO DE USO

- 3.1. Planteamiento
- 3.2. Importancia del modelo a resolver
- 3.3. Datos
- 3.4. Variables utilizadas
- 3.5. Modelos

### 4. DESARROLLO Y CONSTRUCCION DEL MODELO PREDICTIVO

- 4.1. Información del Repositorio
- 4.2. Requisitos técnicos
- 4.3. Carga de la Base de Datos
- 4.4. Información y extracción de la base de datos
- 4.5. Objetivo
- 4.6. Limpieza y Unión
- 4.7. EDA Análisis Exploratorio

4.8. Importancia de las variables

4.9. Preprocesado

4.10. Construcción del modelo

5 Conclusiones

.....

## 1 INTRODUCCION

### 1.1 MOTIVACION

Desde 1998 hasta la actualidad, he desarrollado mi vida profesional en una entidad financiera, principalmente en Banca de particulares. Durante estos 23 años he pasado por todas las categorías laborales posibles dentro de una oficina comercial abierta al consumidor. Desde comercial de caja y de mesa, a subdirector y director de oficina. En 2018, motivado por la búsqueda de nuevas habilidades, reciclaje laboral y personal, la adaptación a la nueva realidad de transformación digital y la necesidad de construir un plan alternativo debido a las inciertas perspectivas laborales, huyendo de mi zona de confort decido cursar un Master en Bussines Analytics con la intención de aprender nuevas formas de análisis de negocio y poder ponerlas en práctica. Durante el curso me doy cuenta que aun sin ninguna base de programación o informática, estadística o matemáticas, procediendo de una licenciatura de letras, había encontrado una motivación, una nueva parcela de estudio y un nuevo reto. Decido continuar la formación con el Master en Data Science de K-School, recomendado por un antiguo profesor y siempre avisado de la dificultad técnica del mismo. El resultado lo puedo definir en una frase. ***Intenso pero entusiasmado y con ganas de continuar mi formación.***

### 1.2 PROLOGO

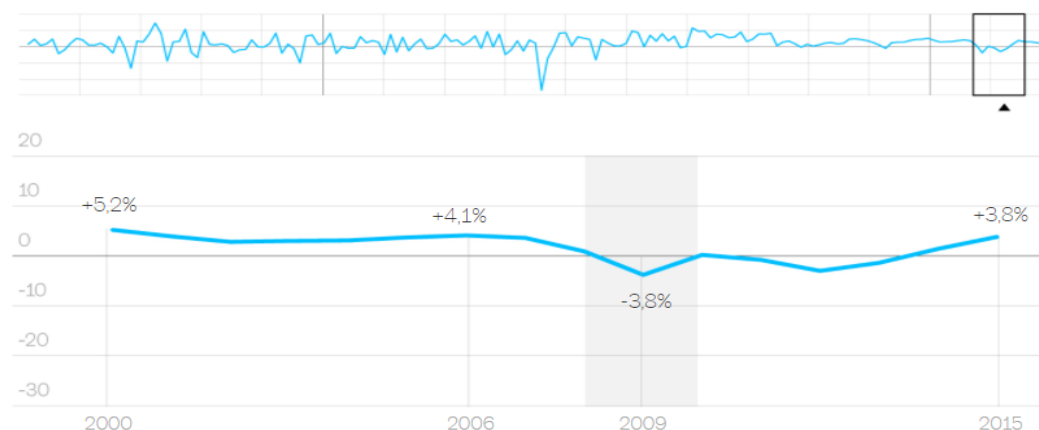
Los bancos son entidades importantes. Canalizan la riqueza entre los distintos actores de la sociedad. Ese ha sido siempre su principal objetivo y razón de ser. Captar recursos de aquellos que los tienen, remunerarles por esa captación y posteriormente prestarlos recibiendo a cambio un tipo de interés. Se trata de una transacción sencilla y fácil. Las entidades financieras pagan por captar y cobran por prestar. De esta forma todas las partes salen beneficiadas. Quien pone a disposición del banco un dinero que no necesita cobra intereses, el banco cobra intereses al prestarlo y aquellos que reciben el préstamo pueden destinar el dinero para aquello que necesitan. Una empresa, una compra de un vehículo...la compra de una casa... etc. Resumiendo, el negocio tradicional de las entidades financieras ha sido prestar a más tipo de interés el dinero que ha captado.

En España, hasta 2008 vivimos en un gran periodo de crecimiento económico impulsado principalmente por un modelo basado en el mercado inmobiliario. Este crecimiento se vio bruscamente interrumpido por la crisis de las hipotecas subprime. Se inicio un largo

periodo de 6 años de crisis profunda y recesión en la que los principales indicadores económicos de la sociedad sufrieron importantes caídas. Las entidades financieras no fueron ajenas a este sufrimiento viendo sus cuentas de resultados mermadas y reducidas considerablemente. El negocio tradicional bancario se había paralizado , se había acabado.

### La Gran Recesión (2008-2013)

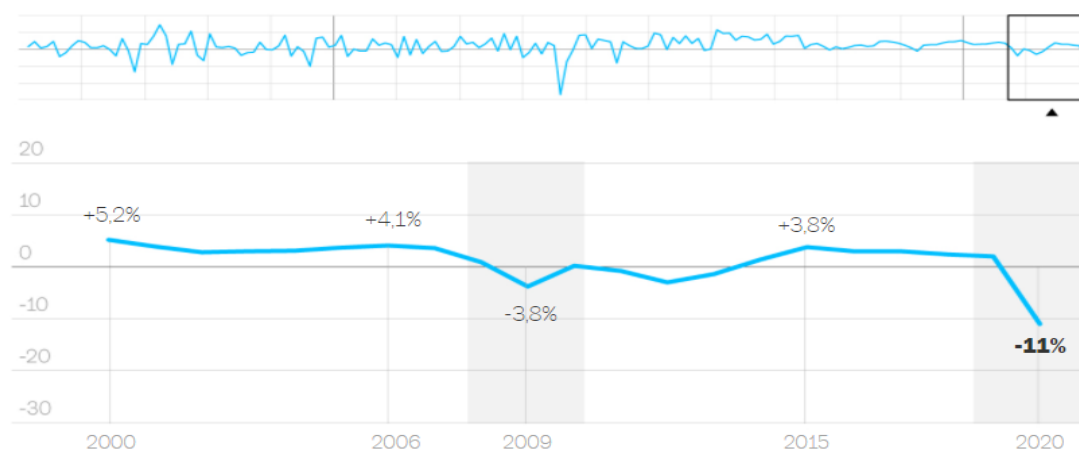
Variación anual del PIB en %



En 2014 se inició la recuperación y cuando parecía que recuperábamos todas las sensaciones, actividad y evidencias económicas de haber olvidado la crisis , en 2020 nos encontramos con una de las peores crisis sanitarias de la historia de nuestra edad moderna generando todavía mayores dificultades y retrocesos que las anteriormente vividas.

### Pandemia de la covid (2020)

Variación anual del PIB en %



Como consecuencia de las diferentes crisis vividas, la reducción de los tipos de interés y de los beneficios obtenidos por el puro negocio tradicional de captar y prestar dinero, este modelo de negocio había dejado de ser el motor principal de la cuenta de resultados de una entidad. La venta de otros productos financieros como los Fondos de Inversión, Planes de Pensión y los seguros de riesgo entre otros, consiguieron acaparar toda la importancia. Hoy en día, la generación de comisiones adheridas a la comercialización de estos productos ha supuesto un vuelco en la estrategia comercial, especializándose y poniendo el foco en su venta mediante el asesoramiento especializado a los clientes por parte de los empleados de las sucursales.

### **1.3 OBJETIVOS**

A través de un conjunto de datos pertenecientes a 450.000 clientes he querido desarrollar un modelo predictivo de potencial compra de estos productos financieros, concretando en el Seguro de Hogar. Un modelo predictivo de clasificación que ayude a toda la fuerza comercial de las sucursales a orientar la comercialización, a optimizar los tiempos, metodologías y sistemáticas utilizadas. Todo ello en búsqueda de un mayor éxito de ventas, satisfacción de los clientes y generación de margen para la entidad financiera.

Mi objetivo final será implementar una aplicación ( CallorNot) donde mediante un cuestionario, incorporando una serie de características de un cliente podamos informar al gestor si el cliente es susceptible de contratar o no un seguro de hogar, sugiriéndole finalmente si llamar o no llamar al cliente.”

### **1.4 FINALIDAD**

Búsqueda de mayores ventas, mejores oportunidades, mayores márgenes y satisfacción final de los clientes

### **1.5 INTERES**

El interés de esta investigación se centra en tres niveles; Empresarial, Comercial y Optimización Comercial.

- Intereses Empresariales: El principal objetivo de una entidad financiera, como cualquier otra empresa privada es maximizar el beneficio que sus socios han invertido, y por los cuales, esperan un retorno a través del reparto de dividendos. La generación de comisiones por la venta de seguros es una de las vías más

importantes para generar beneficios y en consecuencia una mayor cuenta de resultados.

- Desde el punto de vista comercial, la competencia a nivel de mediación y venta de seguros es grandísima. No solo las propias entidades aseguradoras, sino que todas las entidades financieras de hoy en día venden seguros. El hecho de asesorar, vender y captar a un nuevo cliente a medio largo plazo es un hito importante que hay que potenciar.
- Por último, hablar de la optimización del trabajo. No solo hay que trabajar duro, sino trabajar de forma eficaz y eficiente. La jornada laboral de un empleado de banca se resume en Asesorar y vender. Hay que vender y hay que llegar a cuantos más clientes mejor. Si todo esto lo hacemos centrando la llamada, optimizando el tiempo, sabiendo a quien llamamos, a quien nos dirigimos y porqué, mejoraremos en todos los sentidos esa eficiencia y eficacia buscada.

## **2. ESTADO DEL ARTE**

### **2.1 MARCO**

Las Entidades Financieras y los seguros, son el principal sustento de nuestro sistema financiero y asegurador. La economía de cualquier país tiene como pilares principales a estos dos sectores. Tanto la Banca como las aseguradoras están en constante desarrollo tecnológico y organizativos. Ambos sectores jugarán siempre un papel principal en cualquier crisis presente o futura ya que en ellos están depositados planteamientos de recuperación solvencia y liquidez para minimizar el impacto económico en empresas y hogares.

Algunos datos de 2020. En los primeros meses lógicamente influenciada por la gran crisis sanitaria que vivimos, se redujo la siniestralidad en ramos como autos o viajes y aumentó en ramos de decesos, impago de alquileres, vida y salud.

A nivel empresarial se ha evidenciado una alta tasa de anulación de pólizas. Pymes y autónomos son sectores que han padecido con mayor intensidad esta pandemia teniendo como consecuencia cierre de comercios y empresas.

Motivado por todo este descenso asegurador, las primas han subido en las renovaciones y se han incrementado riesgos que antes no había. Ej. riesgos cibernéticos y medioambientales.

Predicciones para seguros en 2021. La **escalada de precios en el mercado global de seguros** se hará evidente. Daños Materiales, Responsabilidad Civil, D&O y Ciberriesgos serán las líneas de negocio más afectadas. **La carga de siniestros relacionados con la pandemia se reducirá** progresivamente debido a la exclusión de las coberturas pandémicas en los acuerdos objeto de renovación

## 2.2 SEGUROS EN ESPAÑA – CIFRAS

Todos los años el INE publica un detalle de los gastos y presupuestos de las Familias españolas. Entre ellos están lo que se gasta en seguros detallándolo por familia exceptuando los de vida que no se informan.

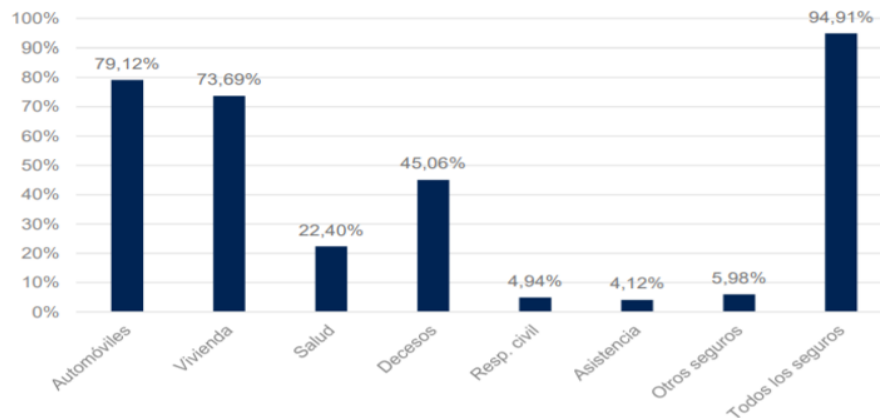
## 2.3 TIPOS DE SEGUROS CONTRATADOS EN ESPAÑA. EVOLUCION ANUAL

	2015	2016	2017	2018
Vida – asegurados por fallecimiento o invalidez	19.098.701	19.650.180	20.170.771	20.466.207
Vida- ahorradores	9.510.371	9.904.511	9.511.544	9.449.896
Salud - prestación de servicios médicos	9.238.717	9.568.054	9.906.084	10.268.012
Salud - subsidios por enfermedad	1.512.863	1.576.477	1.603.951	1.805.347
Decesos	21.090.080	21.260.669	21.537.456	21.763.397
Responsabilidad civil	2.487.013	1.907.115	1.966.052	1.997.918
Automóviles	29.107.481	29.597.454	30.295.290	31.018.517
Viviendas	18.186.862	18.407.931	18.792.044	19.209.473
Comercios	1.270.051	1.291.342	1.312.171	1.324.124
Empresas	1.893.329	2.174.852	2.389.887	2.478.655

Actualmente estamos censados en España 46.722.980 habitantes, contando con un parque de seguros que supera los 119.000.000 de seguros a 2018. El número de seguros de Hogar-Vivienda asciende a más de 19.000.000. Un 15% del total.

El aseguramiento de las viviendas españolas tiene un crecimiento continuo. Mas de 1MM de seguros de hogar se han contratado en el periodo de 2015 a 2018. Seguros de Auto y de Vivienda son los que más crecimiento han tenido.





Fuente: <https://www.unespa.es/que-hacemos/publicaciones/>

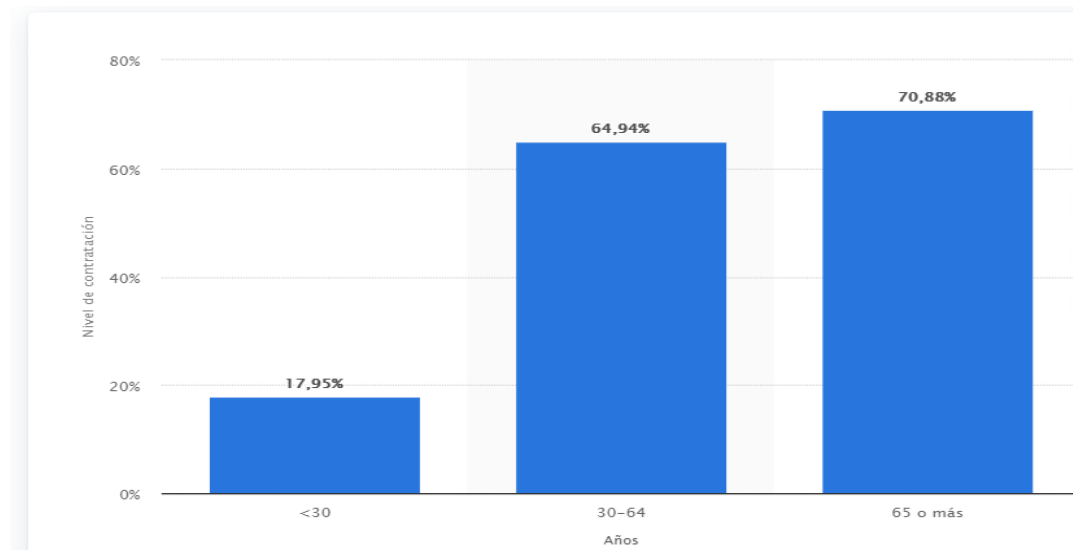
## 2.4 TIPOS DE SERVICIO DEL SEGURO ESPAÑOL

Tipo de percance	Servicios
Percances en viviendas y comunidades	1.003
Percances en comercios	55
Percances del automóvil	1.263
De los cuales: vehículos reparados	487
De los cuales: asistencias en carretera	459
De los cuales: golpes de chapa	230
De los cuales: accidentes graves	60
De los cuales: víctimas en accidentes de tráfico	33
Actos médicos	12.246
De los cuales: visitas al especialista	5.860
Sepelios de decesos	28
Percances de responsabilidad civil	38
Otros percances de particulares	298
Percances de empresas	1.266

Fuente: <https://www.unespa.es/que-hacemos/publicaciones/>

Quitando los servicios médicos, supone el 10% de la demanda de servicios aseguradores.

## 2.5 NIVEL DE CONTRATACION DE SEGUROS POR EDADES



Fuente: <https://es.statista.com/estadisticas/967761/nivel-de-contratacion-de-seguros-de-hogar-por-edad-espana/>

Esta grafica visualiza como el seguro de hogar es contratado a partir de los 30 años. Claramente influenciado por el modelo de vida y dificultad de acceso a la vivienda de las personas jóvenes. Esto lo podremos ver en nuestro modelo predictivo.

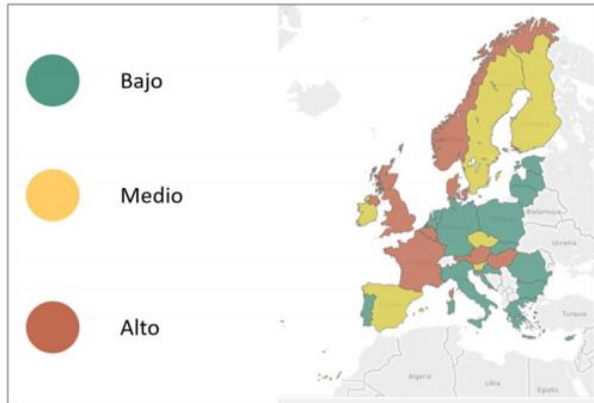
## 2.6 CONTRATACION DE LOS DISTINTOS SEGUROS POR INGRESOS DEL HOGAR

	Automóviles	Vivienda	Salud	Decesos	Resp. civil	Asistencia	Otros seguros	Todos los seguros
Menos de 500 €	54,85%	40,97%	8,92%	44,13%	4,74%	4,06%	4,85%	78,44%
De 500 a 1.000 €	58,74%	58,45%	9,72%	54,56%	3,62%	3,02%	4,22%	89,65%
De 1.000 a 1.500 €	82,20%	73,30%	18,77%	48,37%	4,86%	4,09%	5,91%	96,32%
De 1.500 a 2.000 €	89,16%	83,36%	26,36%	41,40%	4,70%	3,99%	6,56%	98,59%
De 2.000 a 2.500 €	91,58%	88,94%	34,39%	39,25%	5,40%	4,78%	6,70%	99,12%
De 2.500 a 3.000 €	90,83%	89,45%	42,00%	32,30%	6,40%	5,12%	7,36%	99,57%
3.000 € o más	92,18%	89,58%	51,36%	26,71%	8,90%	5,97%	8,69%	99,02%

Las contrataciones de los seguros de hogar dependen en gran medida de los ingresos familiares. Se evidencia un gran corte en los hogares. Las contrataciones despegan con los hogares donde se ingresan mínimo 1000€.

Por último, visualizar a nivel europeo las diferencias de gasto en los seguros de hogar.

## 2.7 NIVELES DE CONTRATACION SEGURO HOGAR EN EUROPA



## 2.8 CONCLUSIONES

El seguro de hogar está muy arraigado en la sociedad española. La cantidad del presupuesto del hogar destinada a los seguros como en el caso de cualquier otro componente de dicho gasto, depende de muchos factores estructurales y contextuales de cada hogar, como el lugar de residencia, la situación laboral de la familia, la escala salarial de sus integrantes, etc.

**Modelo.** Con los datos y variables disponibles intentaremos entrenar una solución comercial que facilite la venta de este producto a los asesores de las entidades financieras.

Dentro del sector financiero existen múltiples modelos de clasificación relacionados con las distintas necesidades.

- Modelos de Riesgo
- Modelos de Morosidad
- Modelos de RRHH
- Modelos de Segmentación Clientes
- Modelos de ventas FFII – PP – Seguros – Servicios Financieros
- Modelos de clientes digitales

### 3 CASO DE USO

#### 3.1 Planteamiento de Caso de Uso.

Modelo predictivo contratación de seguros de hogar

#### 3.2 Importancia del problema a resolver

La generación de comisiones es fundamental para la cuenta de resultados de la oficina y del banco. Los seguros de hogar se quedan en cartera durante un periodo medio de 5 años. Cada seguro de hogar contratado deja una comisión directa del 15 %. Esto sobre un seguro de hogar de prima media de 300€ supone 45€ de comisión anual por cliente. Nuestra base de datos correspondiente únicamente a 162 oficinas y 450.000 clientes podría llegar a generar unas comisiones anuales de más de 17mm€. Solo con esta cifra y extrapolándola a un colectivo de 3 – 4 millones de clientes, queda más que explicado la evidente y clara la necesidad de identificar potenciales clientes que sean susceptibles de contratar el seguro de hogar.

#### 3.3 Datos utilizados.

Los datos provienen de la suma de diferentes conjuntos de información obtenidos directamente de la entidad financiera. TODOS LOS DATOS HAN SIDO ANONIMIZADOS. SE HAN ELIMINADO LOS NOMBRES Y EL NÚMERO DE CLIENTES INTERNOS, SE HAN ELIMINADO EL NÚMERO Y LAS ESPECIFICACIONES DE LAS DIRECCIONES DE ÁREA Y DE OFICINA Y, POR ÚLTIMO, SE HAN ELIMINADO LOS NOMBRES DE LOS ASESORES COMERCIALES. A todos estos datos se les ha asignado un número de identificación ficticio y secuencial, quedando únicamente los datos de tenencia o no de producto por parte de los clientes.

#### 3.4 Variables utilizadas

- ❖ **DZ.** Identificación Dirección de Zona a la que pertenece la oficina. Una Dirección de Zona engloba varias oficinas. Total 11 Direcciones de Zona
- ❖ **OFICINA.** Numero de Oficina / Sucursal de banco
- ❖ **CLIENTE.** Numero de cliente
- ❖ **EDAD.** Edad del cliente

- ❖ **ESTA\_CARTERIZADO.** Identifica si el cliente pertenece o no a una cartera.
- ❖ **CARTERA\_PATRON.** Tipo de cartera a la que pertenece el cliente.
  - Asesoramiento Financiero
  - Tutela. Familiar de cliente Asesoramiento Financiero
- ❖ **CLIENTE\_BBP.** Cliente con saldos superiores a 500.000€ identificado como colectivo Banca Privada.
- ❖ **GESTOR.** Numero identificación del gestor/ Asesor Financiero de la sucursal
- ❖ **TIP\_GESTOR.** Tipo de gestor. Figura de una oficina que ofrece el asesoramiento.
- ❖ **CODIGO\_CARTERA.** Numero identificación cartera a la que pertenece el cliente.
- ❖ **MARCA\_AF\_CCTE.** Identifica si el tipo de gestor es de oficina u Online.
  - AF- Asesor Financiero (Oficina)
  - CCTE – Gestor Online
- ❖ **MARCA\_BANCA\_PERSONAL.** Cliente perteneciente a cartera Asesoramiento Financiero e identificado como colectivo Banca Personal.
- ❖ **SEGMENTO\_RECORRIDO.** Identifica el potencial recorrido comercial del cliente para una mayor vinculación.
  - Alto Recorrido
  - Medio Recorrido
  - Bajo Recorrido
- ❖ **SEGMENTO\_VALOR.** Valor del cliente
  - Alto ( Alto nivel Patrimonial o alto nivel de vinculación )
  - Medio ( Clientes que sin cumplir el anterior requisito tienen un alto nivel de fidelización.
  - Bajo ( Resto de clientes )
- ❖ **CAMINO\_DIGITAL.** Se diferencian 4 tipo de clientes según la utilización de canales digitales.
  - Comprador
  - Consultivo
  - Transaccional
  - Poco uso
- ❖ **DIGITAL\_3\_MESES.** Identifica si el cliente ha utilizado medios digitales durante los últimos 3 meses.
- ❖ **LP\_DOMIC\_INGRESOS.** Tiene o no tiene ingresos domiciliados

- ❖ ***LP\_OFIC\_INTERNET.*** Tiene o no tiene servicio internet
- ❖ ***LP\_REC\_LTGA\_OTR.*** Tiene o no tiene recibos domiciliados
- ❖ ***LP\_SEG\_ACCIDENT.*** Tiene o no tiene seguro accidentes contratado
- ❖ ***LP\_SEG\_AUTO.*** Tiene o no tiene seguro automóvil contratado
- ❖ ***LP\_SEG\_MEDICOS.*** Tiene o no tiene seguro salud privado contratado
- ❖ ***LP\_SEG\_MULTIRRIES.*** Tiene o no tiene seguro hogar contratado
- ❖ ***LP\_SEG\_VIDA.*** Tiene o no tiene seguro vida contratado
- ❖ ***LP\_TARJ\_CREDITO.*** Tiene o no tiene tarjeta crédito pago fin de mes contratada
- ❖ ***LP\_TARJ\_REVOLVING.*** Tiene o no tiene tarjeta crédito pago fraccionado contratada
- ❖ ***SF\_AH\_CAPTACION\_TT.*** Saldo en cuenta de ahorro
- ❖ ***SF\_FINANCIACION\_TT.*** Importe financiación en activo.
- ❖ ***SF\_FONDOS\_INVER.*** Saldo en Fondo de Inversión
- ❖ ***SF\_PLAN\_PENSION.*** Saldo Plan de Pensión

### **3.5 Modelos Machine Learning Utilizados.**

- Modelo Clasificación Regresión Logística
- Modelo Clasificación K-Nearest Neighbor
- Modelo Clasificación TREE
- Modelo Clasificación Xgboost
- Modelo Clasificación Random Forest - MODELO FINAL ELEGIDO

## **4 DESARROLLO Y CONSTRUCCION DEL MODELO PREDICTIVO**

### **4.1 Información del Repositorio**

Toda la información de este TFM ha quedado recogida en un repositorio de GitHub al cual se accede a través de la siguiente dirección.

<https://github.com/romayana/Financial-Product-Sales-Forecast-Model.git>

El repositorio se estructura en 7 carpetas y 3 archivos. ( Según posición en repositorio )

Carpeta 1 – Códigos Python Limpieza y Unión

Carpeta 2 – Códigos Python EDA Análisis Exploratorio

Carpeta 3 – Frontend. Aplicación creada para nuestro modelo (APP CallorNot.)

Carpeta 4 – Imágenes .png guardadas de cada una de las gráficas construidas

Carpeta 5 – Memoria. Documentos y notebooks memoria TFM

Carpeta 6 – Códigos Python Modelos clasificación utilizados.

Carpeta 7 – Códigos Python Preprocesado

Archivo 1 – archivo .gitignore. Archivos descartados en las actualizaciones del repositorio

Archivo 2 – Diccionario e información del significado de las variables

Archivo 3 – Readme con primera información del Trabajo y comunicación de expectativas

main
1 branch
0 tags
Go to file
Add file
Code

Manuel 'updated'		8fcbf38 yesterday	265 commits
Cleanning & Merging	'updating'	13 days ago	
Exploratory Data Analysis	'updated'	2 days ago	
Frontend	'updated'	yesterday	
Images	'updated'	yesterday	
Memory	'updated'	yesterday	
Models	'updated'	yesterday	
Preprocessing	'updated'	2 days ago	
.gitignore	'updating'	20 days ago	
Data Set Codes	updated	6 days ago	
Readme.md	Update Readme.md	yesterday	

Readme.md

## Modelo Previsión Ventas Productos Financieros

## 4.2 Requisitos Técnicos

Para ejecutar los códigos es necesario tener instalado Python versión 3.8 así como distintos paquetes o librerías. Se recomienda tener instalada la Suite Anaconda donde se encontrarán preinstalados la mayoría de los paquetes y librerías que son necesarias.

Librerías utilizadas. La mayoría ya precargadas en Suite Anaconda

```
# Librerías manejo y análisis de estructuras de datos.  
import pandas as pd
```

```
# Librerías especializada en el cálculo numérico y el análisis de datos  
import numpy as np
```

```
# Librerías de Métricas  
from sklearn.metrics import f1_score, recall_score, precision_score, accuracy_score  
from sklearn.metrics import roc_auc_score, roc_curve  
from sklearn.metrics import confusion_matrix  
from sklearn.metrics import classification_report  
from sklearn.metrics import auc  
from sklearn.model_selection import KFold  
from sklearn.model_selection import cross_val_score
```

```
# Librerías de Visualización  
import matplotlib.pyplot as plt  
import pylab as pl  
import seaborn as sns  
from pylab import rcParams  
from matplotlib import pyplot
```

```
# Librerías de Modelos  
from sklearn.linear_model import LogisticRegression  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.neighbors import KNeighborsRegressor  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.model_selection import train_test_split  
from collections import Counter  
from imblearn.over_sampling import SMOTE  
from imblearn.under_sampling import NearMiss
```



Adicionalmente será necesaria la instalación de las siguientes librerías.

- Imbalanced learn – proporciona herramientas cuando se trata de la clasificación con clases desequilibradas. (Instalación mediante consola - pip install imbalanced learn )
- pydotplus - Visualización de árboles de decisión en Python con PyDotPlus. ( instalación mediante consola - pip install pydotplus )
- streamlit – creación e intercambio de aplicaciones web personalizadas para el aprendizaje automático y la ciencia de datos. ( instalación mediante consola - pip install setreamlit )

#### 4.3 Guía de Ejecución y Carga de la base de Datos

1. Clonar repositorio GitHub <https://github.com/romayana/Financial-Product-Sales-Forecast-Model.git> en carpeta local elegida.

2. Descargar base de datos:

A pesar de haberse **Anonimizado** toda la base de datos, se ha decidido que la misma no estará disponible en el repositorio de GitHub. Para acceder a la base de datos ubicada en el Google Drive del propietario del TFM, se tendrá que solicitar permiso y acceso a la misma dirigiendo correo electrónico a [manuelgonzalezprados@gmail.com](mailto:manuelgonzalezprados@gmail.com) el cual previa valoración de los fines y objetivos perseguidos podrá compartir el enlace con la persona solicitante.

Una vez compartido el acceso, descargar y ubicar la carpeta entera llamada **Origin\_Data** dentro de la carpeta local donde se ha clonado el repositorio junto con el resto de carpetas.

3. Ejecutar código con la siguiente secuencia y orden. Los archivos csv se irán guardando en cada una de las carpetas.
  - 1º Carpeta Leasing & Merging
    - \_merging\_data.ipynb
    - \_cleanning\_data.ipynb
  - 2º Carpeta Exploratory Data Analysis
    - EDA.ipynb

- 3º Carpeta Preprocessing
  - Preprocessing.ipynb
- 4º Carpeta Models
  - Ejecutar los modelos.
- 5º Carpeta Frontend
  - Aplicación Callornot.

Dentro de la carpeta Memory existe la posibilidad de ejecutar toda la memoria y código a través del notebook memoria & código. El modelo final al que se hace referencia es Random Forest.

#### **4.4 Información y extracción de la base de datos**

La información y explicación detallada de las variables que conforman la base de datos se encuentra dentro de una de las carpetas de este repositorio con el nombre de Data-Set-Codes. También ha quedado explicada dentro de esta memoria en la sección **3.4 Variables utilizadas**.

La extracción de los datos se ha realizado desde un sistema de información de gestión de una entidad financiera. Sistema de información que guarda millones de datos de tipo financiero y económicos.

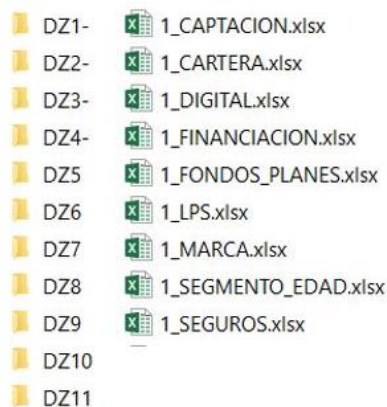
Toda la información y datos necesarios para el estudio del modelo de clasificación se han obtenido de forma directa y con permisos limitados de una entidad financiera real. Permisos limitados ya que no se ha podido disponer de mucha información que hubiese mejorado el modelo. Información como por ejemplo género, estado civil, hijos, clase económica, renta disponible, importe de nómina, detalle de compras realizadas, detalle de llamadas comerciales realizadas y otras muchas variables.

Para la construcción de la base de datos final se han ido descargando de este sistema de información de gestión y de forma manual, archivos individuales extensión xlsx, relacionados con distintos epígrafes como saldos en cuenta, saldos en fondos de inversión o planes de pensión, líneas de producto, tarjetas, seguros o tipo de segmentación. En total han sido 99 archivos Excel descargados. 9 archivos por cada una de las 11 Direcciones de Zona disponibles lo que ha generado finalmente una base de datos de 15 millones de datos de 450.000 clientes.

El peso total de los datos originales es de 112 MB (117.706.752 bytes) los cuales después de haber sido limpiados se han quedado en 70,3 MB (73.742.213 bytes).

**Nuestra carpeta Raiz, desde donde comienza todo, llamada origin\_data está compuesta por las distintas carpetas y archivos Excel descargados de este sistema de información de gestión, guardados ya anonimizados.** Numero de identificador de cliente ha sido cambiado por una secuencia desde 1 a 450.000. Números y códigos distintivos de las Direcciones de Zona han sido cambiados por una secuencia del 1 al 11. Nombres, direcciones y números de identificación fiscal de los clientes han sido eliminados.

Detalle Carpeta con Datos Originales. 11 carpetas x 9 archivos.



#### 4.5 Objetivos

A través de un conjunto de datos pertenecientes a 450.000 clientes he querido desarrollar un modelo predictivo de potencial compra de estos productos financieros, concretando en el Seguro de Hogar. Un modelo predictivo de clasificación que ayude a toda la fuerza comercial de las sucursales a orientar la comercialización, a optimizar los tiempos, metodologías y sistemáticas utilizadas. Todo ello en búsqueda de un mayor éxito de ventas satisfacción de los clientes y generación de margen para la entidad financiera.

Mi objetivo final será implementar una aplicación ( CallorNot) donde mediante un cuestionario, incorporando una serie de características de un cliente podamos predecir la posibilidad de que ese cliente sea susceptible de contratar o no un seguro de hogar, sugiriendo finalmente si llamar o no llamar al cliente.

#### 4.6 Unión de archivos y Preparación de los Datos

- Unión. 11 carpetas correspondientes a 11 Direcciones de Zona y 9 archivos Excel cada uno se fusionan en un solo Data Frame.

De forma secuencial se han ido leyendo los archivos de cada una de las Direcciones de Zona creando una única lista agregada por DZ y finalmente uniendo en una sola base de datos la totalidad de los 99 archivos Excel originales individuales.

Todo el proceso de unión está ubicado en el notebook `_merging_data.ipynb` al cual se puede acceder dentro de la carpeta `Cleanning & Merging` del repositorio.

- Preprocesado y Limpieza de los datos. Esta tarea de preprocesado de los datos o sencillamente de preparación de los datos, la iremos realizando a lo largo de nuestro estudio y en distintas secciones . Detectaremos Nans o valores nulos, se corregirán, se buscarán posibles outliers en variables, se buscarán las mejores o más importantes variables mediante técnicas de feature selection, se estudiará la posible reducción de dimensionalidad, convertiremos variables categóricas en numéricas y normalizaremos las variables a una escala común. Todo ello con el fin de construcción un data set de calidad y así poder trabajar con la mejor información de datos para la construcción definitiva del modelo.

La primera decisión que tomamos de preprocesado con el data set recién construido y unido en la fase de merging, es estudiar y visualizar posibles deficiencias con la detección de valores Nans. De 30 variables, encontramos 10 que contienen Nans. Dado el conocimiento que tenemos de la base de datos, estas se rellenan con distintos valores acordes a la categoría y segmento del dato.

Este primer proceso de limpieza está ubicado en el notebook `_cleanning_data.ipynb` al que se puede acceder dentro de la carpeta `Cleanning & Merging` de este repositorio.

#### 4.7 EDA. Análisis Exploratorio.

Comenzamos el análisis de nuestra base de datos analizando primero el Marco de Datos. Nuestro objetivo es realizar un análisis exploratorio, estudiando los datos, buscando posibles patrones, visualizando los datos estadísticos y encontrando posibles relaciones que serán útiles para entender el contexto del Marco de Datos y posteriormente para nuestro modelo de clasificación.

Esperamos obtener información básica del marco de datos y una información más profunda sobre nuestro Objetivo Seguro de Hogar

Disponemos de una base de datos consistente en 451.374 filas, 30 columnas - variables y 13.541.220 datos. Total memoria utilizada 103.3MB. El nombre de las variables son las siguientes:

'cliente', 'saldo captación', 'esta\_carterizado', 'cliente\_bbp', 'tipo\_gestor', 'gestor', 'cartera\_patron', 'codigo\_cartera', 'digital\_3\_meses', 'camino\_digital', 'saldo\_financiacion', 'saldo\_ffii', 'saldo\_plp', 'lp\_dom\_ingresos', 'lp\_tjta\_cto', 'lp\_tjt\_rev', 'lp\_rbos', 'lp\_of\_int', 'marca\_bp', 'marca\_ccte', 'edad', 'seg\_valor', 'seg\_recorrido', 'dz', 'oficina', 'lp\_seg\_vida', 'lp\_seg\_acc', 'lp\_seg\_salud', 'lp\_seg\_hogar', 'lp\_seg\_auto'

Estas variables las hemos juntado por categorías y segmentado en 6 bloques distintos para poder estudiarlas mejor.

- a. Bloque Unidades de Negocio
- b. Bloque Edad de los clientes
- c. Bloque Ahorro y Financiación
- d. Bloque Servicios
- e. Bloque Seguros de Riesgo
- f. Bloque Segmentación

#### # Contexto.

Para poder entender la importancia de nuestro modelo, el impacto económico y repercusión que puede llegar a tener el hecho de saber diferenciar a los clientes susceptibles de contratar el seguro de hogar vamos a explicar haciendo un retrato piramidal, cómo está estructurada la entidad financiera.

1. Dirección Banca Particulares. Destinada a la atención de clientes particulares / personas físicas.
2. Direcciones Territoriales. División en un número determinado de Direcciones Territoriales según distribución nacional.
3. Direcciones de Zona o de negocio. Cada una de las Direcciones Territoriales está dividida en Direcciones de Zona dando cobertura a cada una de las zonas geográficas de ese territorio.
4. Oficinas. Cada Dirección de Zona está dividida en un número concreto de Oficinas atendiendo a situación geográfica vinculada con esa Dirección de Zona.
5. Asesores Financieros- Cada una de estas oficinas dispone de un número determinado de Asesores Financieros. Entre 1 y 4. Dependiendo el volumen de clientes.
6. Clientes de cada una de las Oficinas.

**Nuestra base de datos hace referencia a una de esas territoriales y su estructura interna con un total de 450.000 clientes.**

Teniendo en cuenta este volumen de negocio de clientes, el impacto y ganancia económica de una entidad financiera con 2000 - 3000 oficinas puede ser enorme.



### **# Bloque 1 Variables de Unidad de Negocio.**

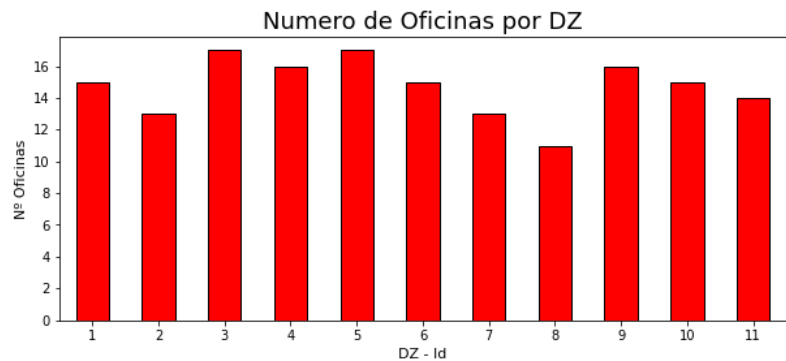
Informamos de inicio de cómo queda la estructura de negocio de este retrato piramidal. Se trata de unidades de negocio por lo que no estarán presentes en nuestro modelo de clasificación. Queremos clasificar a los clientes por su propensión al consumo y no por su ubicación territorial.

- Encontramos 11 Direcciones de Zona con 162 oficinas y 451.374 clientes. Estos clientes están asesorados comercialmente por 458 Asesores Financieros. El 90 % de los clientes pertenecen a una cartera de negocio.

En general igualdad de oficinas por DZ.

DZ nº8 11 oficinas. Corresponde a una DZ con pueblos.

```
Numero de Oficinas por DZ: 162
dz
1    15
2    13
3    17
4    16
5    17
6    15
7    13
8    11
9    16
10   15
11   14
Name: oficina, dtype: int64
```



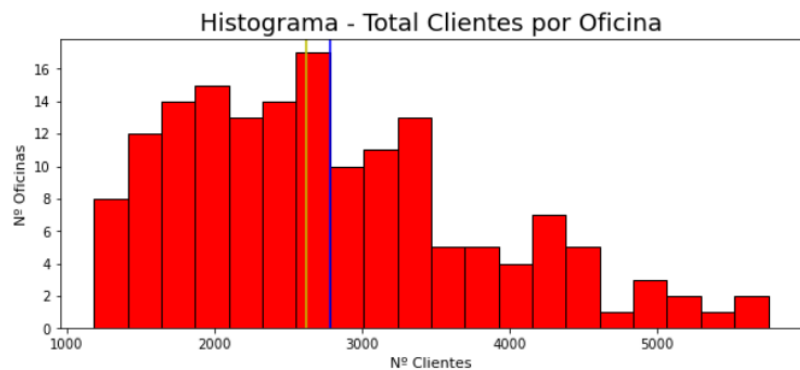
- Colectivo 451.000 Clientes.

Grueso de oficinas entre 1500 y 3500 clientes

Oficinas con menos de 1500 clientes y más de 5000 son escasas.

Total Clientes por Oficina: 451374

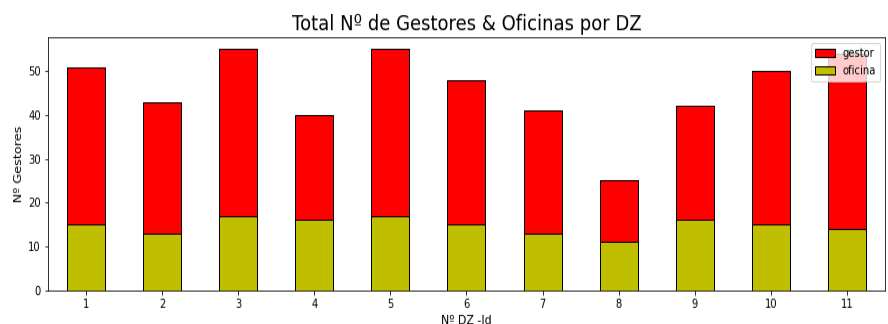
```
count    162.000000
mean     2786.259259
std       1025.420335
min       1185.000000
25%       2000.500000
50%       2619.500000
75%       3357.000000
max       5755.000000
Name: cliente, dtype: float64
```



- Servicio Asesoramiento Especializado
- Colectivo de 458 Empleados
- No hay mucha descompensación entre DZs.
- Media de 48 Asesores – Gestores \*DZ8

Total Nº de Gestores: 458

```
count    11.000000
mean     45.818182
std       8.931049
min       25.000000
25%       41.500000
50%       48.000000
75%       52.500000
max       55.000000
Name: gestor, dtype: float64
```



## # Bloque 2 Edad de los clientes.

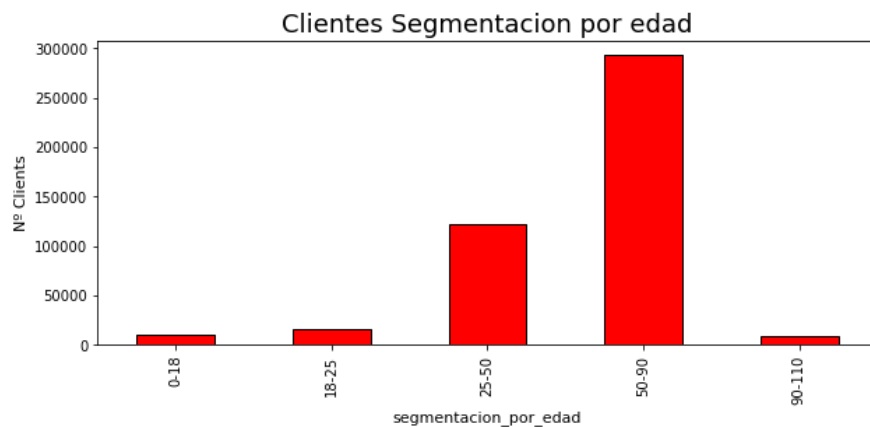
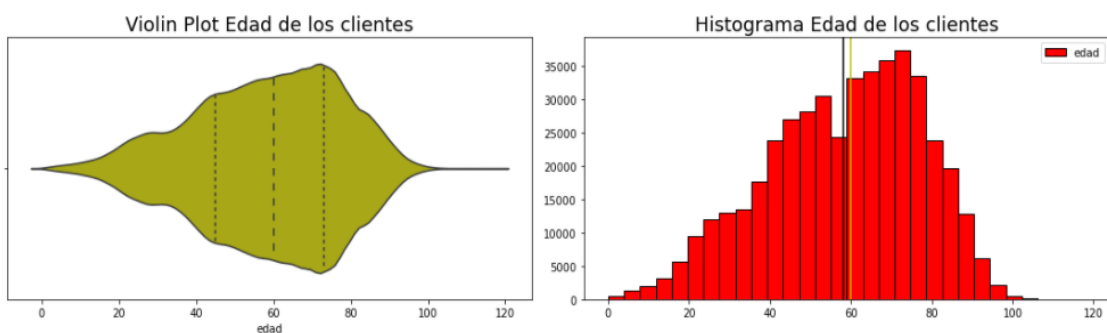
Visualizamos mediante un Box Plot la variable edad de los clientes. Esta visualización nos da la oportunidad de identificar la edad mínima de los clientes, edad máxima, media, así como los cuartiles 25% y 75%.

Adicionalmente realizamos una segmentación de clientes por edad para ver en que rango de edad hay más clientes.

Las gráficas nos dicen que la mayoría de clientes se encuentra en un rango de edad de 50 a 90 años. Se trata de una base de datos con clientes de mediana y avanzada edad.

- La edad mínima que reflejan los datos son 0 años. (Valor incluido en estudio Nans)
- La edad máxima son 118 años.
- La media de edad está en 58.2 años.

count	451374.000000
mean	58.268941
std	19.018368
min	0.000000
25%	45.000000
50%	60.000000
75%	73.000000
max	118.000000



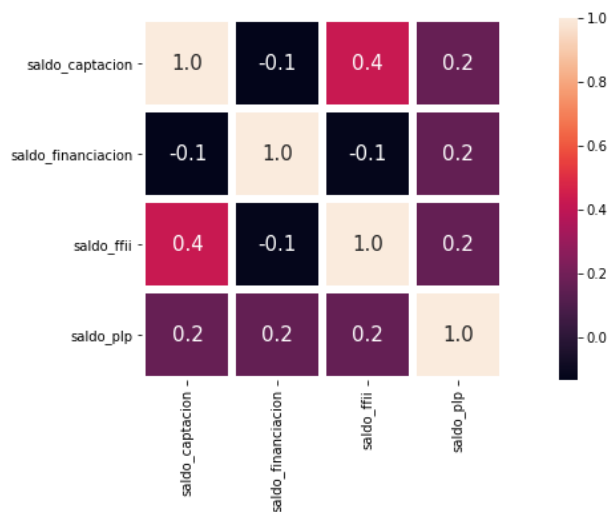


De cara a nuestro modelo, además de no resultar significativos por número total eliminaremos a los clientes entre los rangos de edad de 0 a 25 años y de 90 en adelante. La realidad a la hora de contratar un seguro de hogar es que los menores y jóvenes hasta los 25 años no contratan un seguro porque sencillamente no tienen vivienda propia. En cuanto a los clientes mayores de 90 años, dada su avanzada edad, normalmente son sus hijos los responsables de los seguros de hogar, por lo que vamos a eliminarlos del modelo.

## # Bloque 3 Variables de Ahorro y Financiación.

Dentro de este bloque estudiamos las variables saldo captación, saldo financiación, saldos en fondos de inversión y saldos en planes de pensión.

Iniciamos el estudio de este bloque mediante la visualización de posibles correlaciones entre las propias variables por si pudiésemos eliminar alguna de ellas por alta correlación.



La grafica de correlaciones muestra que no existe grandes relaciones por lo que, a falta del estudio de cada una de las variables de forma independiente, en principio se serán mantenidas todas ellas de cara al modelo de clasificación.

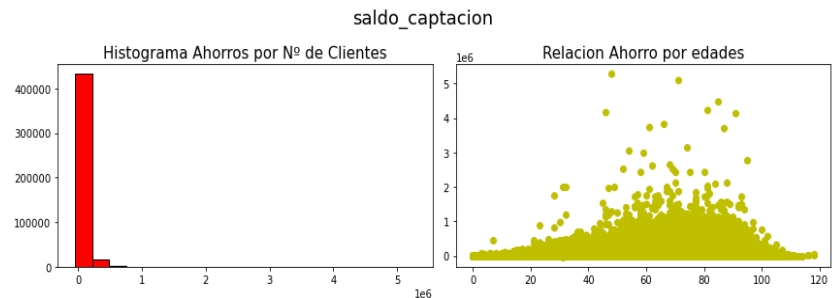
Vemos una a una cada variable.

- Saldos de ahorro.

Conseguimos entender cuál es la realidad de los ahorros de los clientes mediante la visualización por histogramas y Scatter Plot, y mediante descripción estadística de la variable. Según arroja la función describe() la media de saldo en cuenta de los clientes es de menos de 5.000€. Si bien existen clientes con saldos importantes situándose el máximo en 5.2mm€ el 75% de los clientes no llega a los 6.500€ y los ahorros se distribuyen claramente entre los clientes a partir de 60 años hasta los 90. Solo 215 clientes disponen de mas de 1mm€ en cuenta.

```
Nº Clientes con saldos en cuenta: 436942
Nº Clientes con saldos > 1mm€: 215

count    4.513740e+05
mean     4.971455e+04
std      8.792364e+04
min     -5.104332e+04
25%      1.813182e+03
50%      1.520828e+04
75%      6.446964e+04
max       5.290216e+06
Name: saldo_captacion, dtype: float64
```



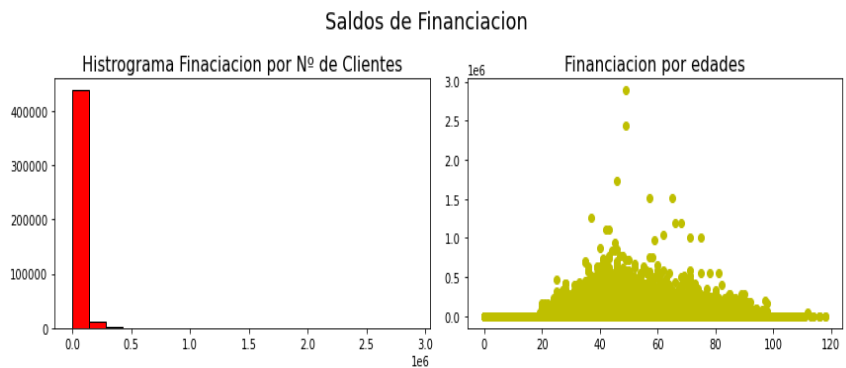
- Saldos Financiación.

La mitad de los clientes de nuestra base de datos tienen préstamos, pero estos son de pequeño importe. Solo el 10% de los clientes tienen préstamos superiores a 50.000€, cantidades que pueden dar a pensar que la financiación corresponde a préstamos hipotecarios de adquisición de vivienda. El préstamo medio es de 13.000€ seguramente préstamos destinados a consumos familiares, vehículos, reformas, etc. El rango de edad donde mas préstamos tienen son de 30 a 75 años, edades que casan con la realidad del rango de edad en los que se solicitan préstamos al consumo. Por último, vemos que hay clientes con deudas.

Se trata de una variable importante que mantendremos en el modelo. Si bien daremos valor 0 a los clientes que tienen saldo negativo / deudas para no distorsionar.

Nº Clientes con Financiación: 223894  
Nº Clientes con Financiación > 50.000€: 42860

```
count    4.513740e+05
mean     1.380668e+04
std      4.296438e+04
min     -4.900000e+03
25%      0.000000e+00
50%      0.000000e+00
75%      6.841775e+02
max      2.896317e+06
Name: saldo_financiacion, dtype: float64
```

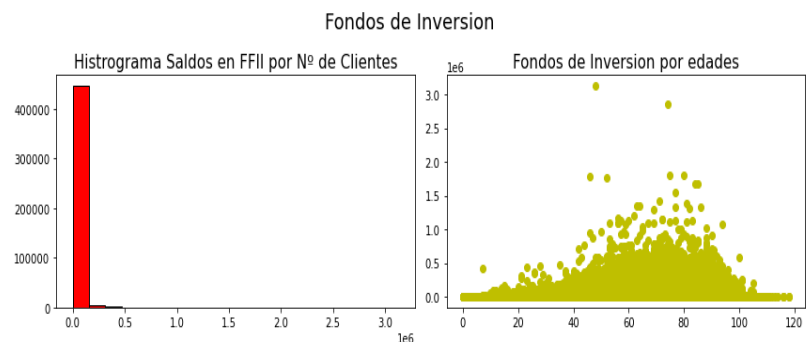


- Fondos de Inversión.

Se trata de una variable que mide el nivel de ahorro de los clientes y también de cultura financiera. No todos los clientes son susceptibles de tener este tipo de ahorro. Bien por desconfianza o por desconocimiento de su funcionamiento. Aquellos clientes que si disponen de Fondos de Inversión están dentro de un rango de edad de 50 a 90 años. Con la descripción estadística vemos claramente que la mayoría del cliente no tienen fondos. La media se sitúa en 8.000€.

Nº Clientes con Fondos Inversión: 60932  
Nº Clientes con Fondos Inversión > 50.000€ : 23479

```
count    4.513740e+05
mean     8.483663e+03
std      3.844671e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      0.000000e+00
max      3.130940e+06
Name: saldo_ffii, dtype: float64
```



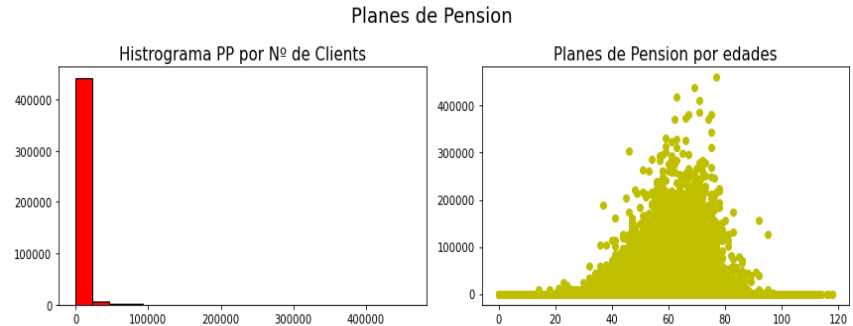
- Planes de Pensión.

Se trata de un producto dirigido a la jubilación. Se trata de ahorrar durante la vida activa para cuando se llegue a la jubilación se pueda complementar la pensión que se recibirá de la seguridad social. En la misma línea de los fondos de inversión, se trata de un producto que no todo el mundo tiene a pesar de ser muy buena opción de ahorro aunque no la única. Los clientes que tienen planes de pensión se sitúan en rango de edad de 40 años hasta llegar al máximo con 65 años donde empieza a descender motivado por los rescates de los planes una vez jubilados los clientes.

```

Nº Clientes sin Planes Pension 404808
Nº Clientes sin Planes Pension > 100.000€: 1262

count    451374.000000
mean      1728.478326
std       10780.622435
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max      459937.460000
Name: saldo_plp, dtype: float64
    
```



La realidad comercial es que el hecho de tener saldos en Fondos de Inversión o Planes de Pension no es una de las condiciones que llevan a los clientes a contratar o no un Seguro de Hogar. Estas dos variables se convertirán en variables de categoría booleana. Los valores asignados serán 0 si no se tiene y 1 si se tiene. De esta forma simplificaremos el estudio.

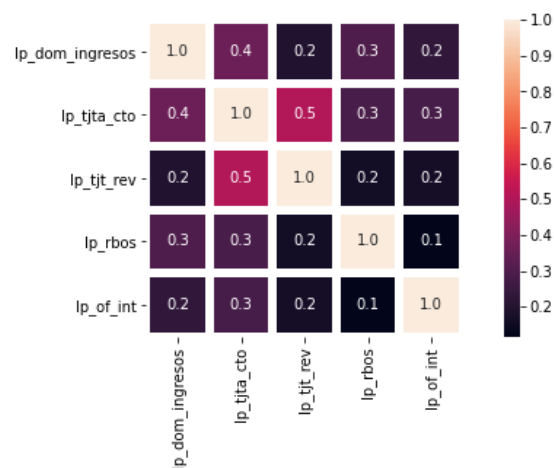
En cuanto a la variable de saldo en cuenta. Son valores reales y si pueden tener efecto en la contratación del seguro. Sólo modificaremos los negativos y les daremos el valor de la media.

Por último, vemos que el 50% de los clientes sí tienen financiación. Mantenemos la variable que puede ser útil para el modelo. A los clientes con saldo deudor les damos un valor de 0€.

## # Bloque 4 Servicios

Dentro de este bloque estudiamos los distintos servicios que disponen los clientes. Nóminas, domiciliación de recibos, tarjetas de crédito y App internet.

Visualizamos en primer lugar las correlaciones que puedan tener entre sí.



La mayor correlación que existe 0.5 es de tarjeta de crédito con tarjeta revolving. Se trata de dos tarjetas de crédito, pero de distinto tipo de uso. La primera el tipo de pago es 100% a fin de mes, mientras que la tarjeta revolving es de pago fraccionado mediante cuota definida fija.

La alta correlación de estas tarjetas responde a una comercialización conjunta de ambas tarjetas en un mismo acto. A pesar de esta correlación del 0.5 ambas tarjetas serán mantenidas en nuestro modelo ya que se trata claramente de dos realidades de financiación distintas.

Estudiamos una a una nuestras variables de servicio.

- Domiciliación de Nomina.

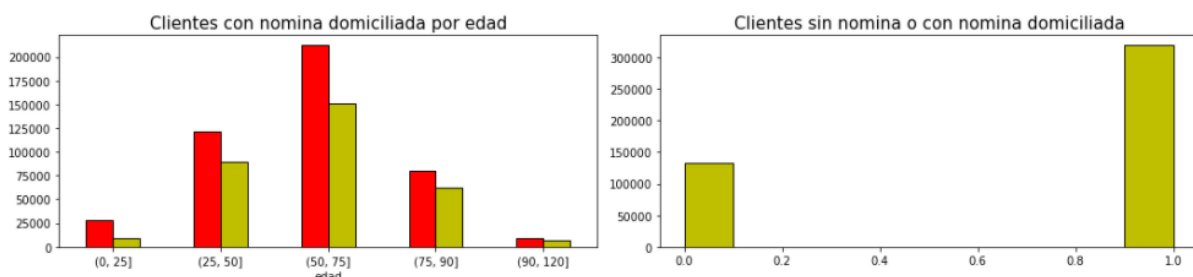
Variable muy importante. Quizás el servicio más demandado por una entidad financiera. La domiciliación de nómina hace referencia a la nómina del trabajo y a la nómina por pensión. El hecho de tener la nomina domiciliada en un banco supone una serie de beneficios mutuos para consumidor y entidad. El consumidor puede obtener mejores precios de préstamos, menos comisiones y mejores primas en seguros. Por otro lado, la entidad financiera a pesar de cobrar menos comisión a este tipo de clientes se beneficia en la propia comisión generada por la comercialización de los seguros y de servicios asociados que pueden conllevar.

Tenemos a un amplísimo porcentaje de clientes que tienen la nomina domiciliada y entre todo tipo de segmento de edad. Si bien se agudiza en edades de mas de 25 años manteniéndose hasta la propia jubilación con la domiciliación de la pensión.

% Clientes con Ingresos 0.71

edad	lp_dom_ingresos count	sum
(0, 25]	27661	9013
(25, 50]	121987	89581
(50, 75]	212315	150816
(75, 90]	80580	62504
(90, 120]	8818	7264

Domiciliacion de Nomina

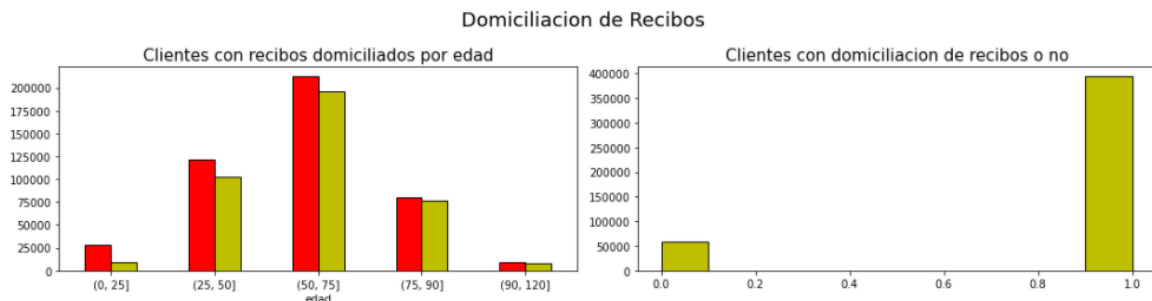


- Domiciliación de Recibos.

Este tipo de servicio suele ir de la mano a la domiciliación de la nómina . Esto se ve en las siguientes graficas que son muy parecidas a las del servicio nómina. El 87% de los clientes tienen el servicio. Por supuesto este tipo de servicio también deja importantes beneficios a un banco ya que para tener domiciliado un recibo es necesario disponer de saldo en cuenta y el saldo en cuenta es sencillamente margen de beneficio para la entidad.

% Clientes con recibos domiciliados 0.87

edad	lp_rbos count	sum
(0, 25]	27661	9382
(25, 50]	121987	103084
(50, 75]	212315	196207
(75, 90]	80580	76691
(90, 120]	8818	8169

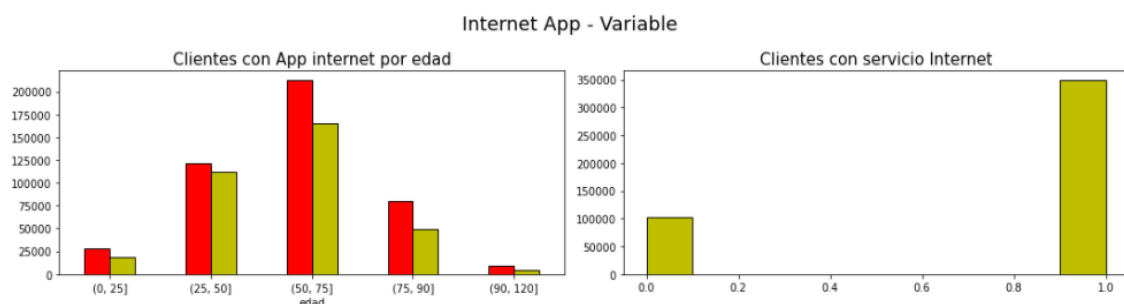


- App Internet.

Servicio destinado a la consulta y operativa 24 horas de las posiciones de un cliente. A través de este servicio es posible realizar compras, transacciones y consultas sin que la oficina del banco este abierta. Es un servicio que tienen la mayoría de los clientes, el 77% en este caso. Los beneficios de este servicio son para ambos lados. Cliente puede realizar en cualquier momento una operación y la entidad financiera ahorra costes de estructura y esfuerzos logísticos.

% Clientes con App Internet 0.77

edad	lp_of_int count	sum
(0, 25]	27661	18593
(25, 50]	121987	112287
(50, 75]	212315	164809
(75, 90]	80580	49159
(90, 120]	8818	4138



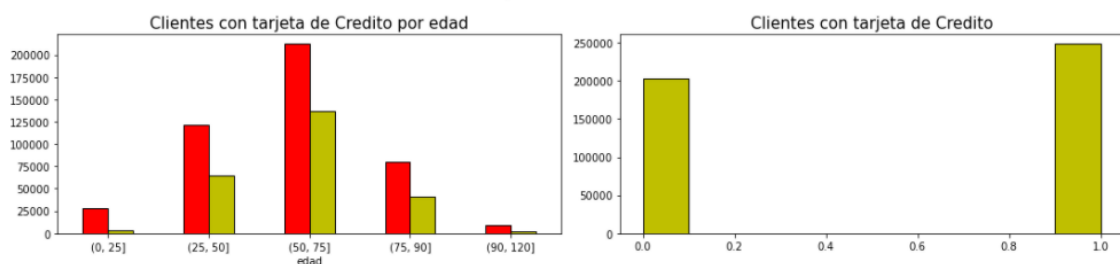
- Tarjetas de crédito y Revolving.

Servicio que mantienen el 55% de los clientes en el caso de Tjta e Crédito pago 100% y el 24 % de los clientes en el caso de Revolving.

% Clientes con Tarjeta de Credito 0.55

edad	lp_tjta_cto	count	sum
(0, 25]		27661	2864
(25, 50]		121987	64390
(50, 75]		212315	137133
(75, 90]		80580	41694
(90, 120]		8818	2391

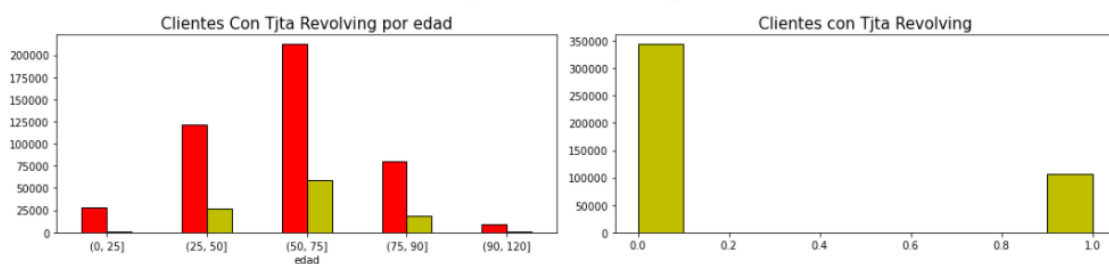
Tarjetas de Credito



% Clients with Revolving Credit Card 0.24

edad	lp_tjt_rev	count	sum
(0, 25]		27661	994
(25, 50]		121987	26939
(50, 75]		212315	59209
(75, 90]		80580	19130
(90, 120]		8818	964

Tarjeta Credito Revolving



## RESUMEN VARIABLES SERVICIOS

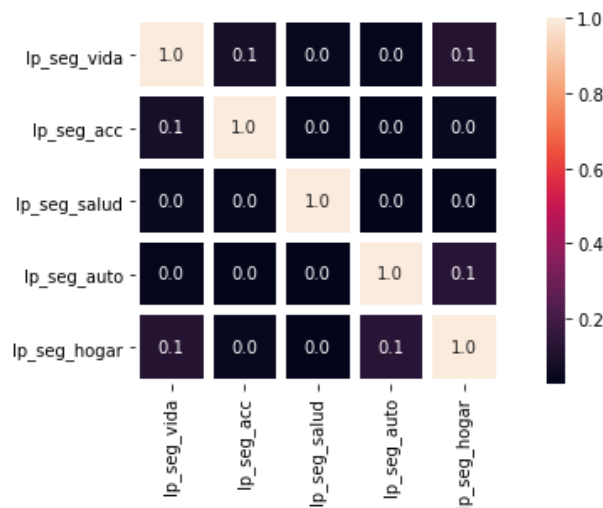
- Clientes con Ingresos 0.71%
- Clientes con Tita Cto 0.55%
- Clientes con Tjta Rev 0.24%
- Clientes con Rbos domiciliados 0.87%
- Clientes con Servicio Internet 0.77%

Como conclusión de este bloque n 4 # Servicios, podemos asegurar que el data set esta formado por clientes con media - alta vinculación con la entidad financiera. Aun así, se evidencia un amplio potencial de desarrollo comercial. Las cinco variables que componen este bloque van casi de la mano tanto en tenencia como en el segmento de edad por utilización. Son clientes entre 25 y 75 años los que mas utilizan estos servicios.

Todas las variables serán mantenidas para el modelo ya que consideramos que son útiles para estimar si el cliente puede contratar el seguro de hogar.

## # Bloque 5 Seguros de Riesgo

En primer lugar, vemos como las variables no tienen ninguna correlación entre sí.



- Seguros de Vida, Accidentes, Salud y Auto.

Comprobaremos como estos seguros que atienden a contingencias de vida, accidentes y salud no tienen una buena penetración en los clientes. El seguro más contratado por los clientes es el seguro de vida y solo se trata de un 8%. Los porcentajes de tenencia de los otros seguros descienden hasta sencillamente el 0% en el caso de Salud. Anteriormente en el bloque de servicios hemos comentado como el hecho de tener nómina y recibos domiciliados puede ser una ventaja para los clientes por ahorrarse y beneficiarse de bajadas de prima en seguros. Estos datos reflejan el altísimo potencial comercial y de mejora en este segmento. Se trata de una oportunidad importante de generación de comisiones que no hay



que dejar pasar. Estos datos cruzados con el servicio de recibos pueden ser importante ya que es posible detectar clientes con recibos de seguros en otras aseguradoras. Esto es una verdadera oportunidad comercial ya que la realidad es que en cualquier casa o familia lo normal es que haya uno o dos vehículos, el cabeza de familia tenga un seguro de vida o la familia tenga un seguro de salud. Este porcentaje tan bajo de posesión de seguros indica que probablemente los tengan en otra entidad financiera o compañía de seguros.

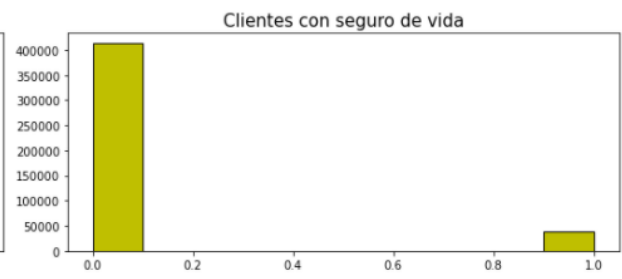
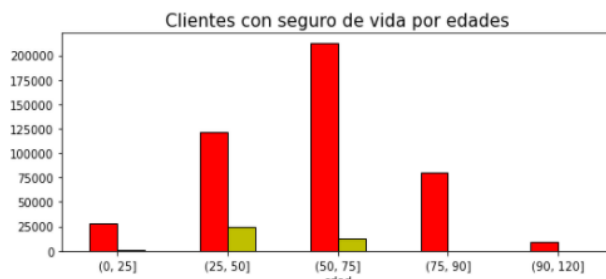
Resumimos los % de tenencia de seguros

- % Clientes con Seguro Vida 0.08
- % Clientes con Seguro Accidentes 0.02
- % Clientes con Seguro Salud 0.0
- % Clientes con Seguro Auto 0.01

% Clientes con seguro de vida 0.08

edad	lp_seg_vida count	sum
(0, 25]	27661	967
(25, 50]	121987	24373
(50, 75]	212315	12515
(75, 90]	80580	1
(90, 120]	8818	0

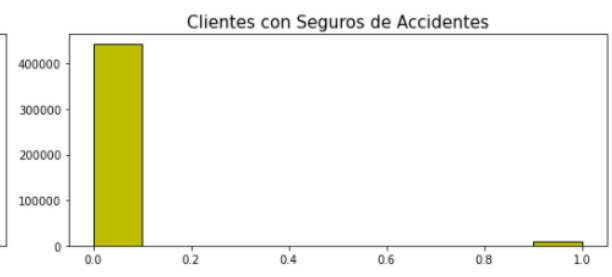
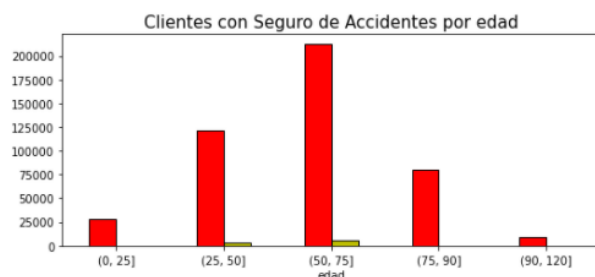
## Seguros de Vida



% Clientes con Seguro de Accidentes 0.02

edad	lp_seg_acc count	sum
(0, 25]	27661	86
(25, 50]	121987	2867
(50, 75]	212315	6035
(75, 90]	80580	38
(90, 120]	8818	0

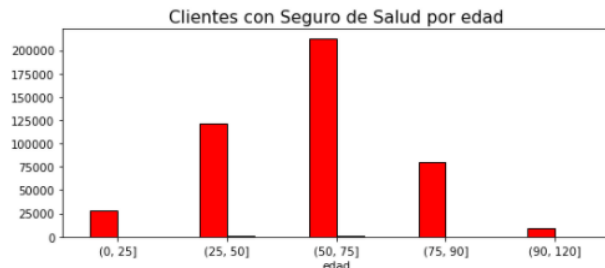
## Seguros de Accidentes



% Clientes con Seguro de Salud 0.0

edad	lp_seg_salud	count	sum
(0, 25]		27661	53
(25, 50]		121987	793
(50, 75]		212315	1003
(75, 90]		80580	75
(90, 120]		8818	0

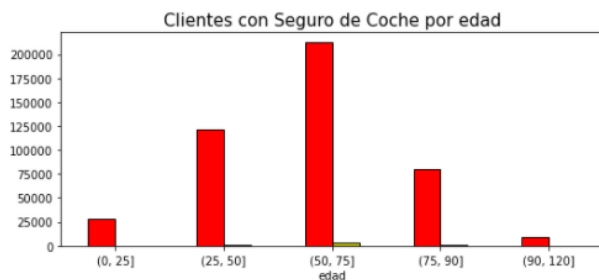
Seguros de Salud



% Clients with Car insurance 0.01

edad	lp_seg_auto	count	sum
(0, 25]		27661	9
(25, 50]		121987	1482
(50, 75]		212315	3762
(75, 90]		80580	974
(90, 120]		8818	5

Seguro de Coche



- Seguro de Hogar **TARGET**.

Estudiamos mas en profundidad nuestra Target. ¿Cuál es su estructura con respecto a nuestro marco general, que variables son importantes , podemos descartar alguna de ellas?

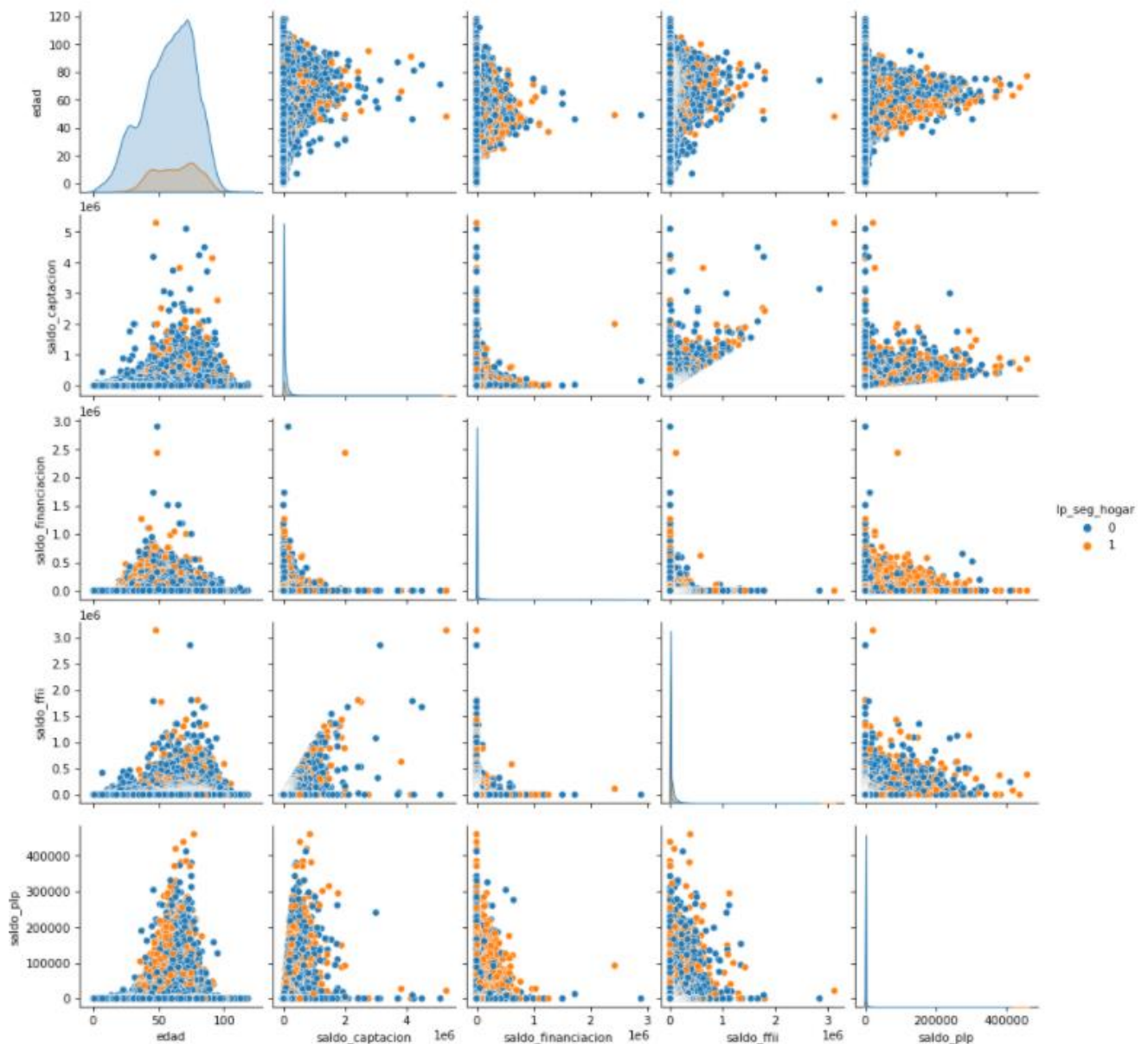
Visualizamos relaciones mediante pairplot. Exclusivamente variables monetarias y edad

Visualizamos cuantos clientes tienen el seguro de hogar y como están distribuidos en nuestras direcciones de zona.

La DZ que más clientes tiene con seguro de hogar contratado es la DZ 3 con 7233 clientes mientras que la que menos tiene es la DZ 8 con 2.274 clientes. En todo caso, las cifras son bajas y la media se posiciona en 5.346 clientes. Estos datos vs un data set de 450.000 clientes reflejan un gran potencial de comercialización.

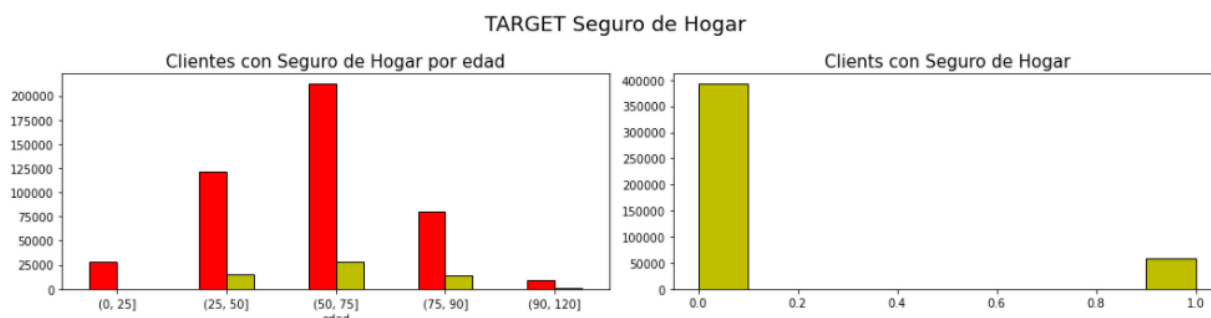
PairPlot. Relacionamos exclusivamente las variables monetarias y edad con el seguro de hogar para ver posibles diferencias y relaciones entre sí.

En general amplio potencial de contratación de seguro de hogar.



% Clientes con Seguro Hogar 0.13 --TARGET

lp_seg_hogar	count	sum
edad		
(0, 25]	27661	78
(25, 50]	121987	15337
(50, 75]	212315	27874
(75, 90]	80580	14217
(90, 120]	8818	1305

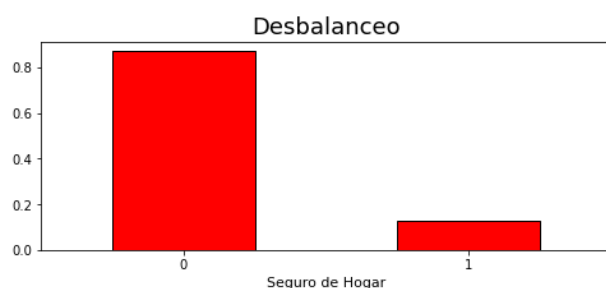


Vemos como es un seguro que tampoco tiene mucha penetración en los clientes. Solo el 13 % de los clientes de la base de datos tiene este seguro. El segmento de clientes con más contrataciones de seguros de hogar es entre 25 y 90. Ninguno de los extremos tienen contratación de seguros relevantes. En este caso, atendiendo a estas magnitudes y confirmando lo estudiado en la variable edad, eliminaremos de nuestro modelo a clientes que de 0 – 25 años y de 90 – 120 años.

Esta cifra refleja otra circunstancia importante y es el gran desbalanceo que tiene. Esto tendremos que tenerlo en cuenta en la construcción del modelo.

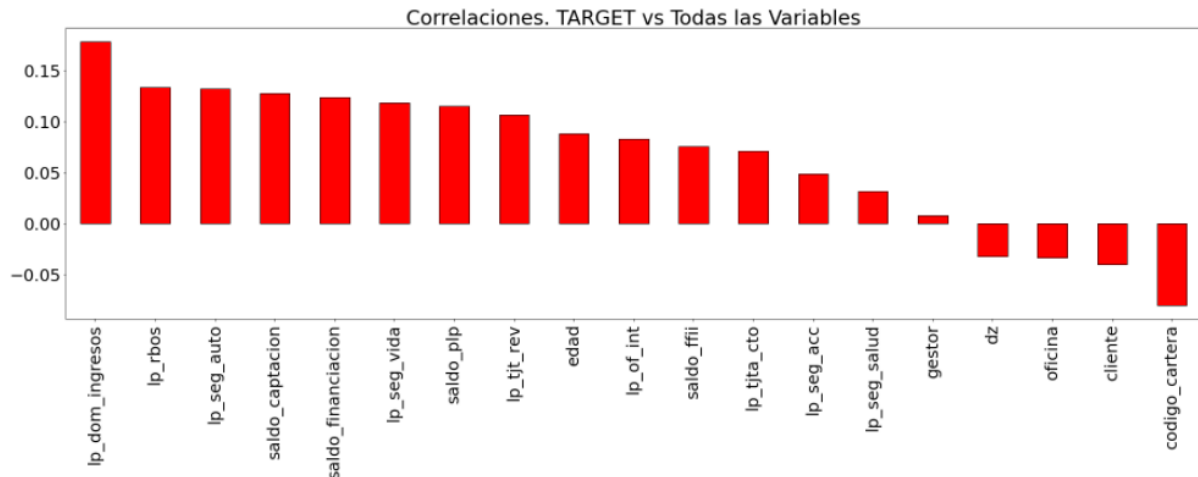
Seguro de Hogar

% Clientes con Seguro de Hogar 0.13 --TARGET



¿Cuál es la correlación que tiene el seguro de hogar con todas las otras variables de la data set.? Vemos como aquellos que tienen nomina domiciliada y recibos domiciliados son los clientes que más relación tienen con el seguro de hogar. Aun así, la mayor correlación es de 0.17.

lp_seg_hogar	1.000000	lp_of_int	0.082749
lp_dom_ingresos	0.177929	saldo_ffii	0.075671
lp_rbos	0.133464	lp_tjta_cto	0.070634
lp_seg_auto	0.132036	lp_seg_acc	0.048611
saldo_captacion	0.127449	lp_seg_salud	0.031651
saldo_financiacion	0.123231	gestor	0.007628
lp_seg_vida	0.118517	dz	-0.032331
saldo_plp	0.115034	oficina	-0.033456
lp_tjt_rev	0.106714	cliente	-0.039775
edad	0.087744	codigo_cartera	-0.080364



## # Bloque 6 Variables de Segmentación

Estas variables hacen referencia a distintos ámbitos en los que un cliente puede estar segmentado. El 89% de los clientes están carterizados y asignado a un asesor financiero, aunque también hay clientes asignados a otras figuras como el subdirector.

Una variable importante es el segmento de recorrido. Nos indica que el 70% de los clientes tiene recorrido comercial. (No hace más que confirmar lo visto hasta ahora.)

Camino digital y Digital 3 Meses. Otras variables importantes que reflejan la poca utilización del servicio internet . El hecho de no utilizar este servicio esta asociado a la edad de los clientes. Además de ser un servicio que no tienen muchos clientes el segmento de edad donde mas se utiliza es de 25 a 50. Clara evidencia de la realidad tecnológica y de la falta de transformación y adaptación digital de los clientes de mediana y avanzada edad.

```
=====
ESTA_CARTEORIZADO :::
SI      0.904416
NO      0.095584
Name: esta_carterizado, dtype: float64
-----
CLIENTE_BBP :::
NO      1.0
Name: cliente_bbp, dtype: float64
-----
TIPO_GESTOR :::
ASESOR FINANCIERO      0.894690
SIN GESTOR              0.095584
SUBDIRECCIÓN DE OFICINA 0.009726
Name: tipo_gestor, dtype: float64
-----
CARTERA_PATRON :::
ASESORAMIENTO FINANCIERO 0.697767
TUTELA                    0.206649
SIN CARTERA               0.095584
Name: cartera_patron, dtype: float64
-----
MARCA_BP :::
NO      0.720633
SI      0.279367
Name: marca_bp, dtype: float64
-----
```

```
-----
MARCA_CCTE :::
AF      0.612485
SIN MARCA 0.270126
CCTE     0.117388
Name: marca_ccte, dtype: float64
-----
SEG_VALOR :::
ALTO     0.552963
MEDIO    0.298890
BAJO     0.148148
Name: seg_valor, dtype: float64
-----
SEG_RECORRIDO :::
ALTO RECORRIDO 0.363315
MEDIO RECORRIDO 0.332148
BAJO RECORRIDO 0.283333
NO CALCULADO 0.021204
Name: seg_recorrido, dtype: float64
-----
DIGITAL_3_MESES :::
SI      0.510096
NO      0.489904
Name: digital_3_meses, dtype: float64
-----
CAMINO_DIGITAL :::
SIN USO      0.430175
CONSULTIVO   0.264466
TRANSACCIONAL 0.215511
POCO USO     0.058242
COMPRADOR    0.031606
Name: camino_digital, dtype: float64
-----
```

Variables categóricas “Esta Carterizado” y “BBP” serán eliminadas del modelo ya que el data set esta justamente construido sobre la base de un conjunto de clientes que tienen una cartera definida. Por otro lado, ningún cliente es BBP.

El resto de variables serán mantenidas a falta de un estudio mayor de importancia de las variables.

#### 4.8 Importancia de las Variables.

Estudiamos la importancia de las variables utilizando un modelo de Random Forest y una matriz de correlaciones.

Con este estudio pretendemos revelar que importancia tiene cada una de las variables dentro de nuestro data set. Los datos revelados son los siguientes:

En cuanto a la matriz de correlaciones hemos utilizado la tabla de correlación de Spearman, cuyo resultado arroja correlaciones más altas que los demás métodos.

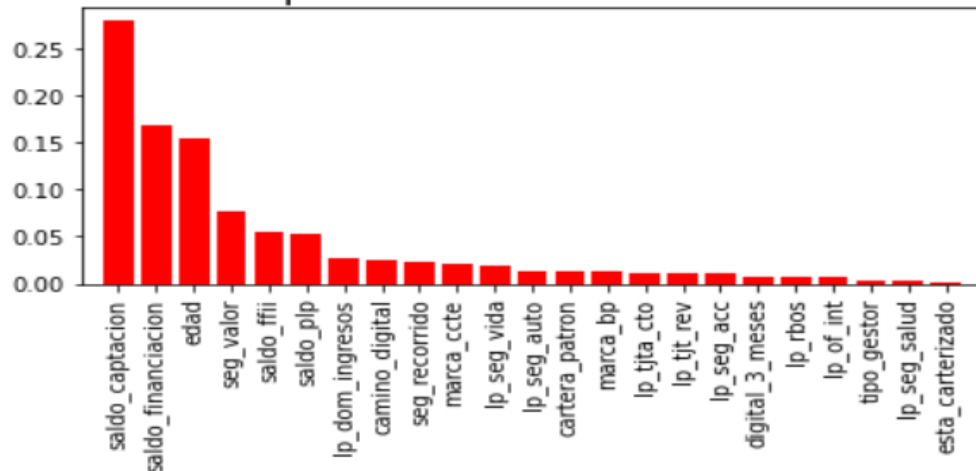
La correlación entre las variables es generalmente baja. Incluso hay correlaciones negativas, pero también bajas.

Basándonos en el resultado obtenido a través del modelo de Random Forest utilizado para revelar la importancia de cada una de las variables, vamos a eliminar del modelo por su escasa importancia las siguientes:

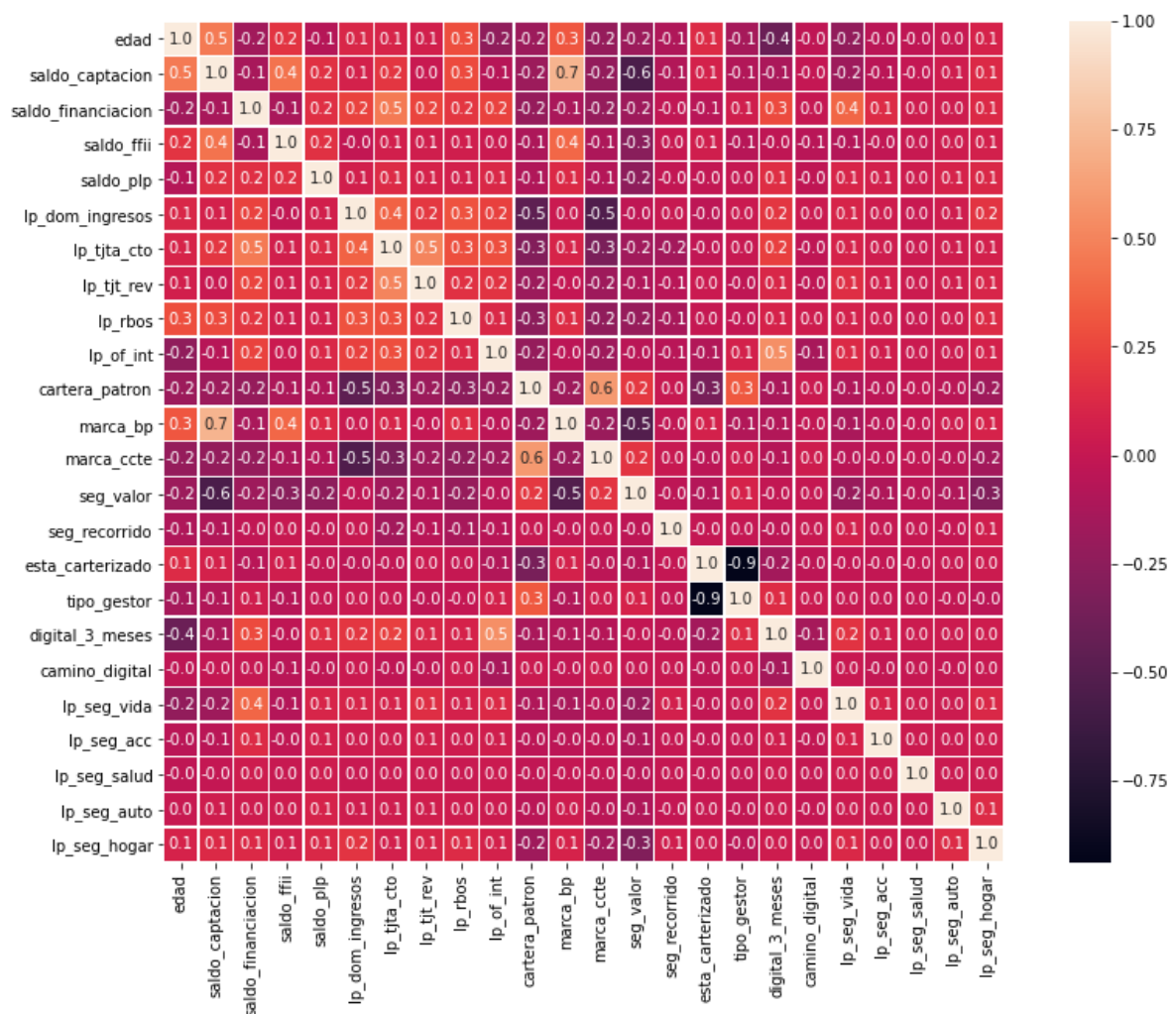
- 18) digital\_3\_meses 0,007651
- 19) lp\_rbos 0,007175
- 20) lp\_de\_int 0,006659
- 21) Tipo de gestor 0.003427
- 22) lp\_seg salud 0,003167
- 23) Esta\_Carterizado 0,001778. En este caso se confirma nuestra decisión anterior.
- 13) cartera\_patron 0,012169 Correlación con variable 10) marca\_ccte 0,021080

1) saldo_captacion	0.279079	13) cartera_patron	0.012169
2) saldo_financiacion	0.169123	14) marca_bp	0.012156
3) edad	0.154577	15) lp_tjta_cto	0.011191
4) seg_valor	0.076552	16) lp_tjt_rev	0.010838
5) saldo_ffii	0.054786	17) lp_seg_acc	0.010216
6) saldo_plp	0.053133	18) digital_3_meses	0.007651
7) lp_dom_ingresos	0.026537	19) lp_rbos	0.007175
8) camino_digital	0.024198	20) lp_of_int	0.006659
9) seg_recorrido	0.022719	21) tipo_gestor	0.003427
10) marca_ccte	0.021080	22) lp_seg_salud	0.003167
11) lp_seg_vida	0.018652	23) esta_carterizado	0.001778
12) lp_seg_auto	0.013138		

Importancia de las Variables



Matriz de Correlaciones





**Conclusiones Finales EDA** .Variables a mantener y datos a modificar.

**Bloque 1** - Variables unidades de negocio. Nuestro objetivo es determinar con los productos que tienen los clientes si son susceptibles de tener o no tener un seguro de hogar. Las Unidades de Negocio no han de influir en el modelo. Las elimino por lo tanto del modelo.

**Bloque 2** - Variables Edad. Elimino del modelo a los clientes de 0 a 25 años y de más de 90 años.

**Bloque 3** - Variables de Ahorro - Financiación. Fondos de inversión y Planes de pensión son convertidas a booleano.

**Bloque 4** - Variables de Servicios. Eliminamos Seguro Salud, Seguro Accidentes y tenencia de Internet.

**Bloque 5** - Variables de Seguros. Todos los seguros serán utilizados dentro del modelo.

**Bloque 6** - Variables de Segmentación. Cartera Patrón, Clientes carterizados y BBP son eliminados del modelo.

**Bloque 7** - Variables digitales . Se mantiene camino digital.

**30 variables originales. 17 se mantendrán en modelo incluida target.**

- 1) saldo captación 0.279079
- 2) saldo financiación 0.169123
- 3) edad 0.154577
- 4) seg\_valor 0.076552
- 5) saldo\_ffii 0.054786
- 6) saldo\_plp 0.053133
- 7) lp\_dom\_ingresos 0.026537
- 8) camino\_digital 0.024198
- 9) seg\_recorrido 0.022719
- 10) marca\_ccte 0.021080
- 11) lp\_seg\_vida 0.018652
- 12) lp\_seg\_auto 0.013138
- 14) marca\_bp 0.012156
- 15) lp\_tjta\_cto 0.011191
- 16) lp\_tjt\_rev 0.010838
- 17) lp\_seg\_acc 0.01021



#### 4.9 Preprocesado.

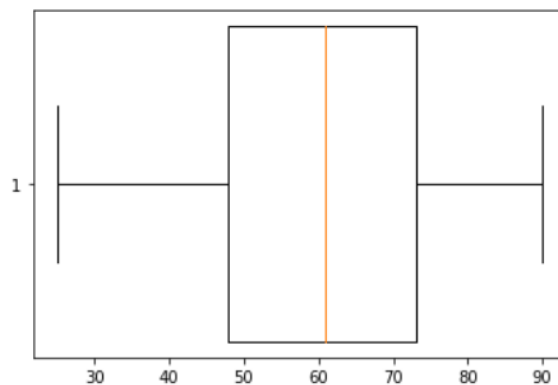
Recordamos que la preparación de los datos y el preprocesado de la base de datos comenzó inicialmente en el punto 4.6 Unión de archivos y Preparación de los Datos, justo después de haber construido y unido todos los datos en un solo archivo. En esa ocasión vimos y analizamos los Nans y les dimos valores acordes a las categorías de cada variable.

A lo largo del estudio EDA realizado, se han llegado a una serie de conclusiones las cuales vamos a poner en práctica en este capítulo. Realizaremos por lo tanto aquellas modificaciones necesarias para dejar limpio y preparado nuestra base de datos para la construcción del modelo.

**1º** Eliminamos 2 variables categóricas de segmentación. Está carterizado y cliente BBP.

**2º** Estudio de Outliers . La única variable que consideramos estudiar a tal efecto, es la variable Edad. Al eliminar a los clientes entre los rangos de edad 0-25 y 90 según hemos concluido en el Eda confirmamos mediante grafica boxplot que finalmente la variable edad no contiene outliers.

Box Plot Variable Edad



**3º** Fondos de Inversión y Planes de Pensión son convertidas en variables booleanas. (0-1) Tiene o no tiene.

**4º** Variable Captación: Damos valor cero a todos aquellos valores negativos quedando conformado de la siguiente forma:

```
df['saldo_captacion'].describe()
count    4.178120e+05
mean      5.194882e+04
std       8.914284e+04
min       0.000000e+00
25%      2.189145e+03
50%      1.787874e+04
75%      6.811098e+04
max       5.290216e+06
Name: saldo_captacion, dtype: float64
```

5º Label Encoder - Dentro de nuestras variables de segmentación que definen una condición específica de cada uno de los clientes, encontramos finalmente cinco variables categóricas que queremos utilizar y que necesitamos convertir en número para un mejor aprovechamiento.

```
variables_segmentacion = ['seg_valor', 'camino_digital', 'seg_recorrido', 'marca_ccte', 'marca_bp']
```

Sustituimos así cada uno de los posibles valores de definición por un número. Conseguimos mediante este método poder utilizar en nuestro modelo y en un mismo espacio todas las variables numéricas y categóricas.

Antes de Label Encoder

```
df['seg_valor'].unique()
array(['ALTO', 'MEDIO', 'BAJO'], dtype=object)
```

Después de Label Encoder

```
# Confirmamos que la transformación se ha realizado correctamente.
df['seg_valor'].value_counts()
0    249593
2    134911
1     66870
Name: seg_valor, dtype: int64
```

6º Normalización - Poco a poco vamos evolucionando nuestra base de datos. De las 17 variables definitivas que hemos dejado para el estudio del modelo, encontramos dos de ellas “Saldo Captación” y “Saldo financiación” con valores reales establecidos en Euros. Valores especificados en miles, cientos e incluso millones de euros. También tenemos la variable edad que debemos trabajar. Necesitamos convertir todas las variables a una misma escala común que no haga distorsionar o dar mayor importancia a unas variables numéricas de otras. Utilizamos el método de Normalización MinMaxEscaler y lo que conseguimos es traducir todas las variables a un rango entre 0 y 1

Antes de Normalizar

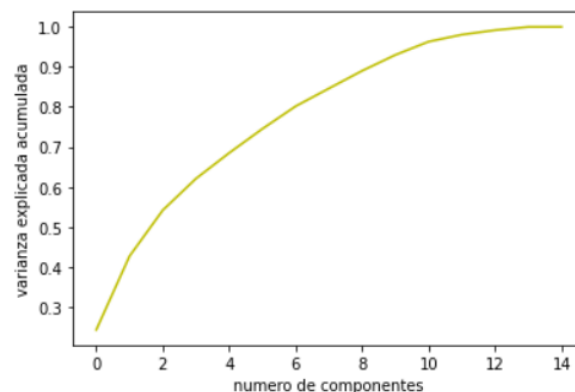
	saldo_captacion	saldo_financiacion	edad	seg_valor	saldo_ffii	saldo_plp	lp_dom_ingresos	camino_digital	seg_recorrido	marca_ccte	lp_seg_vida
216560	696.78	6784.51	56.0	1	0	0	0	2	0	2	0
306754	2537.44	5483.04	65.0	0	0	0	1	3	0	0	1
229289	205132.98	0.00	38.0	0	1	0	1	4	1	0	0

Después de Normalizar

	saldo_captacion	saldo_financiacion	edad	seg_valor	saldo_ffii	saldo_plp	lp_dom_ingresos	camino_digital	seg_recorrido	marca_ccte	lp_seg_vida
0	0.251050	0.001689	0.766667	0.0	1.0	1.0	1.0	0.00	0.333333	0.0	0.0
1	0.215915	0.001689	0.900000	0.0	1.0	0.0	0.0	0.00	0.000000	0.0	0.0
2	0.215915	0.001689	0.833333	0.0	1.0	0.0	0.0	0.75	0.333333	1.0	0.0

7º PCA Posible simplificación de variables reduciendo dimensionalidad a la base de datos.

Después de haber reducido el Data Frame eliminando unidades de negocio y otras variables según estudio EDA, nos hemos quedado únicamente 17 variables. No consideramos necesario utilizar reducción de dimensionalidad PCA y lo confirmamos con la siguiente grafica en la que vemos como después del estudio de PCA a partir de 13 – 14 variables explicarían el 99 % de los 17 originales. Por lo tanto, utilizar PCA no implicaría una mejora sustancial del resultado.



### Conclusiones Previas a la Construcción del Modelo.

Finalizamos con este capítulo la preparación, preprocesado de los datos para iniciar la construcción de nuestro modelo de clasificación. A lo largo de este proceso hemos generado hasta 3 base de datos distintas para utilizarlas en nuestro modelo.

- Base de datos eliminando variables categóricas, cambiando a booleano variables de FFII y PP, eliminando clientes de ciertos rangos de edad y adaptando las variables categóricas con Rabel Encoder.
- Base de datos anterior a la que se le une la normalización a través de minmaxscaler para que todas las variables estuviesen en una misma escala.
- Base de datos anterior implementando PCA reducción de dimensionalidad.

Hemos modificado valores Nans, hemos eliminado outliers en aquella variable susceptible de tenerla, se han adaptado variables de captación y financiación eliminando valores negativos, se han adaptado variables de fondos de inversión y planes de pensión dándoles valores booleanos 0 – 1 y se han aplicado técnicas de reducción de dimensionalidad y de transformación de variables categóricas a numeral.

Con toda esta cirugía aplicada a la base de datos construida inicialmente, desarrollaremos el modelo de clasificación cuya finalidad recordamos es “Generar un modelo predictivo de clasificación que ayude a toda la fuerza comercial de las sucursales a orientar la comercialización, a optimizar los tiempos, metodologías y sistemas utilizados. Todo ello en busca de un mayor éxito de ventas , margen para la entidad financiera y satisfacción de los clientes”

Todo el proceso de unión está ubicado en el notebook “preprocessing” al cual se puede acceder dentro de la carpeta Preprocessing del repositorio.

#### **4.10 Construcción del Modelo.**

Recordamos cual era nuestro Objetivo y Finalidad:

**Objetivo:** Desarrollar un modelo predictivo de compra de productos financieros concretando en los Seguros del Hogar.

**Finalidad:** Generar un modelo predictivo que ayude a toda la fuerza comercial de las sucursales a orientar la comercialización, a optimizar los tiempos, metodologías y sistemas utilizados. Todo ello en busca de un mayor éxito de ventas y satisfacción de los clientes.

#### **Elección del Modelo:**

El modelo final de clasificación elegido ha sido RandomForest

Buscamos una sistemática comercial que simplifique el estudio de los clientes antes de llamarles. Buscamos una optimización del tiempo para llegar a cuantos mas clientes mejor. Nos interesa finalmente filtrar aquellos clientes a los que tenemos que llamar. Este filtro se puede obtener desde dos perspectivas. Y lo buscamos dentro de una matriz de confusión. Buscando verdaderos positivos para cargarlos en el objetivo de contactos diarios y que los comerciales puedan llamarles o identificando verdaderos negativos para desecharlos y finalmente coger los verdaderos positivos y llamarles. En ambos casos llegamos a la misma conclusión y obtenemos lo que realmente queremos. Llamar a los que nos interesan.

Se han desarrollado 5 modelos distintos de clasificación con el fin de buscar aquel que mejor métricas y mejores comportamientos tenga según nuestras necesidades concretas. Las principales métricas que utilizaremos como evaluadores de nuestros modelos son Recall y Roc Auc.

### Modelos Machine Learning Entrenados.

- Modelo Clasificación Regresión Logística
- Modelo Clasificación K-Nearest Neighbor
- Modelo Clasificación TREE
- Modelo Clasificación Xgboost
- Modelo Clasificación Randon Forest - MODELO FINAL ELEGIDO

Mostramos de inicio y a efectos comparativos una Tabla de métricas obtenidas de cada uno de los modelos entrenados.

Modelo	accuracy	precision	recall	f1_score	roc_auc
Regresión Logística	0.6793	0.2818	0.8873	0.4277	0.8282
K Nearest Kneighbour	0.8767	0.5930	0.2785	0.3790	0.8283
Tree	0.7502	0.3327	0.8446	0.4774	0.8694
Random Forest	<b>0.7773</b>	<b>0.3664</b>	<b>0.8543</b>	<b>0.5128</b>	<b>0.8863</b>
XGBoost	0.8911	0.7173	0.3413	0.4625	0.8923

El desarrollo y código de todos estos modelos se encuentran dentro de la carpeta “ Models” de este repositorio.

Como hemos comentado al inicio, en esta memoria vamos a centrarnos en el **Modelo Final Elegido RANDOMFOREST**. Mostraremos como iniciamos el modelo, nuestra primera aproximación, como se ha evolucionado y desarrollado, dificultades encontrados y soluciones implementadas, evolución de métricas obtenidas y finalmente la conclusión y resultado final.

La utilización del Randon Forest obedece no solo a unas mejores métricas obtenidas sino al aprovechamiento de la baja correlación entre las variables, a una simplificación del estudio al no ser estrictamente necesarias técnicas de normalización o de reducción de dimensionalidad. Se trata de dividir rangos y no de compararlos. Random forest es la suma de muchos Arboles de decisión individuales lo que lo hace mucho más fuerte. Adicionalmente suele tener mejores rendimientos en algoritmos de clasificación como es nuestro caso. Por último, su fácil interpretación y explicación basada en divisiones.

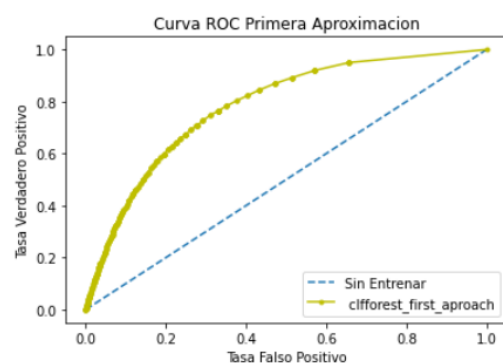
- Primera Aproximación

Nuestro camino comienza con la base de datos limpia con un preprocesado único de modificación de Nans. Se trata de la primera base de datos obtenida después de haberla unido en la primera fase de construcción. Queríamos saber con los datos en crudo que comportamiento tenía el modelo y que métricas obteníamos sin hacer nada intermedio.

Los Resultados fueron los siguientes. Claramente mejorables.

```
== Primera Aproximacion RandomForest ==  
-----  
accuracy_score = 0.8564386596510661  
-----  
precision      = 0.40895707422904076  
-----  
recall         = 0.22392872295290622  
-----  
f1_score       = 0.2893957670797236  
-----  
roc_auc_score  = 0.7831319874986574
```

Sin Entrenar: ROC AUC=0.500  
Tree: ROC AUC=0.783



Comenzamos el estudio real del modelo. Recordamos que durante el preprocesado de la base de datos, se generaron hasta 3 bases de datos distintas.

1. Base de datos eliminando variables categóricas, cambiando a booleano variables de FFII y PP, eliminando clientes de ciertos rangos de edad y adaptando las variables categóricas con Label Encoder.
2. Base de datos anterior a la que se le une la normalización a través de minmaxscaler para que todas las variables estuviesen en una misma escala.
3. Base de datos anterior implementando PCA reducción de dimensionalidad.

El modelo ha sido entrenado con las dos primeras bases de datos. La tercera base con PCA implementada no la vimos necesaria dado que lo único que hacíamos era reducir dos variables.

El entrenamiento y resultado de las dos primeras bases de datos ha sido casi idéntico. Observamos que el hecho de haber normalizado los datos no ha sido un mejor condicionante para obtener mejores resultados. La explicación de esto es que en un modelo de RandomForest, la normalización no tiene por qué ser necesaria ya que no se comparan magnitudes. Se trata de dividir rangos y no de compararlos. Por otro lado, durante el

preprocesado hemos visto como la importancia de las variables no difería mucho. Estos motivos han hecho que en el estudio final no se haya utilizado la base de datos normalizada.

La primera decisión que tomamos en esta primera evolución del modelo es utilizar un estimador que intente poner remedio al gran desbalanceo de nuestro target. Utilizamos el estimador de “Class Weigth = Balanced”. Las métricas obtenidas solo con este ajuste mejoran algo nuestra primera aproximación, pero en ningún caso lo suficiente.

==== Random Forest RESULTADOS ====

-----  
accuracy\_score = 0.8805452173808982  
-----

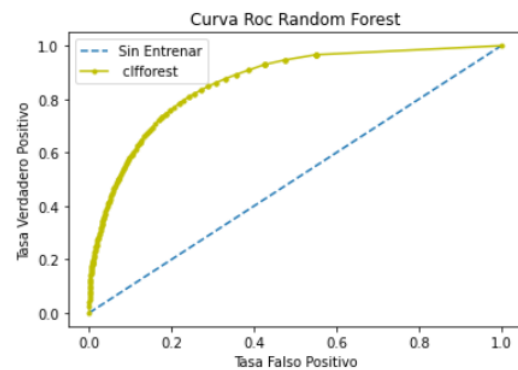
precision = 0.6033184606804238  
-----

recall = 0.3774424284717376  
-----

f1\_score = 0.4643700364885168  
-----

roc\_auc\_score = 0.8627974699671257  
-----

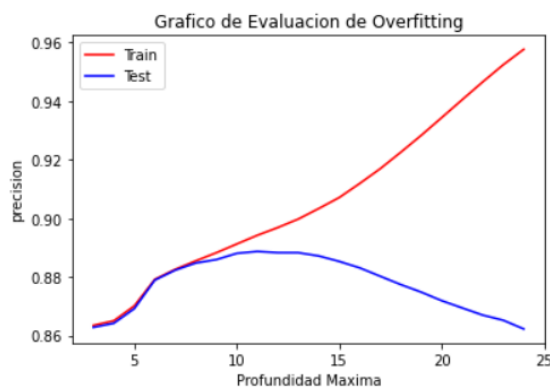
Sin Entrenar : ROC AUC=0.500  
Random Forest: ROC AUC=0.863



**Evaluación de Overfitting:** Lo que si evidenciamos en esta primera evolución del modelo es la existencia de Overfitting.

Los resultados obtenidos en train y test muestran de inicio que el modelo no está generalizando bien. Se obtienen métricas de 0.99 en train y 0.37 en Test. Hay una grave situación de sobreajuste confirmada mediante Cross Validation, que tenemos que resolver.

A través de la siguiente grafica se muestra como el modelo de RandomForest, a partir de 8 profundidades empieza a no generalizar bien, evidenciándose el Overfitting



=====RECALL TRAIN=====  
recall = 0.9995836620212113  
=====RECALL TEST=====  
recall = 0.37918702023726447

Podemos decir que este intento de mejora del modelo no es satisfactorio. Hemos confirmado que no se ha generalizado bien, que existe overfitting, que el desbalanceo del target no lo soluciona del todo y que las métricas requeridas no son acordes a nuestra necesidad. Conclusión. Hay que seguir evolucionado para mejorarlo.

Para ello **utilizaremos Gridsearch** que “permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicándole un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los segundos mediante validación cruzada”

#Parámetros sugeridos:

```
# Sugerimos la búsqueda de los mejores parametros

forest_grid_params = {
    'n_estimators' : [100,125],
    'max_features' : ["auto", "log2"],
    'criterion' : ['gini', 'entropy'],
    'max_depth' : [2,4,6,8,10],
    'min_samples_split' : [2,4],
    'min_samples_leaf' : [15,20]
}

forestgs = GridSearchCV(
    clfforest,
    forest_grid_params,
    verbose = 1,
    cv = 3,
    n_jobs = -1
)
```

#Parámetros recomendados mediante GridSearch:

```
# Mejores Estimadores
forestgs.best_estimator_

RandomForestClassifier(class_weight='balanced', max_depth=10,
                        min_samples_leaf=15)

# Mejores Parametros
forestgs.best_params_

{'criterion': 'gini',
 'max_depth': 10,
 'max_features': 'auto',
 'min_samples_leaf': 15,
 'min_samples_split': 2,
 'n_estimators': 100}
```



- **Adaptación del Modelo con mejores parámetros recomendados:**

Implementamos los mejores parámetros recomendados.

La primera mejora obtenida es que hemos conseguido solucionar el overfitting igualando las métricas de Train y Test a 0.85 Mediante Cross Validation se confirma. Primer problema solucionado.

```
=====RECALL TRAIN=====
recall = 0.8649750197212727
=====RECALL TEST=====
recall = 0.8543265875785067
```

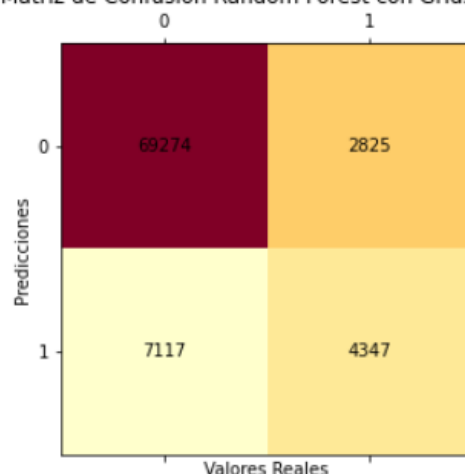
## Cross validation

0.8604203152364274,

## Estudio de la Matriz de Confusión.

En la diagonal de aciertos obtenemos 69274 verdaderos positivos y 4347 verdaderos negativos. En nuestro modelo prima la detección de los clientes que sí quieren contratar el seguro. Necesitamos obtener una buena tasa de verdaderos positivos o verdaderos negativos. Si detectamos a los clientes que probablemente vayan a contratar o no un seguro de hogar, podremos trabajar de forma más directa. Todo ello con el objetivo de optimizar el tiempo y la eficiencia comercial. La realidad comercial es que da lo mismo trabajar con verdaderos positivos o negativos. Ambos sirven de filtro para poder llamar o no llamar. El resultado de Recall obtenido es óptimo, con un 85% y un 88% en AUC.

Matriz de Confusion Random Forest con Gridsearch



- Resultados Finales y evolución de métricas

==== Random Forest RESULTADOS FINALES ====

-----  
accuracy\_score = 0.7773775474791474  
-----

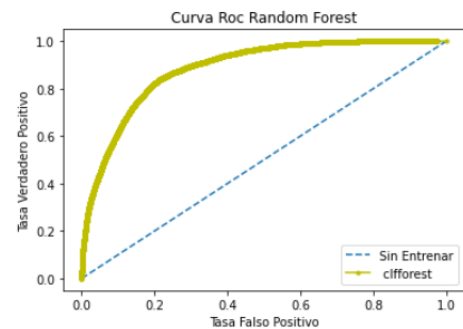
precision = 0.36644591611479027  
-----

recall = 0.8543265875785067  
-----

f1\_score = 0.5128957084129768  
-----

roc\_auc\_score = 0.8863073702441031

Sin Entrenar : ROC AUC=0.500  
Random Forest: ROC AUC=0.886



Random Forest	1ª Aproximación	Evolución	Gridsearch
accuracy	0.8564	0.8810	0.7773
precision	0.4089	0.6061	0.3664
recall	0.2239	0.3791	0.8543
f1_score	0.2893	0.4665	0.5128
roc_auc	0.7831	0.8644	0.8863

## 5 Conclusiones Generales.

- Modelo utilizado

La estrategia que he seguido para buscar un modelo y unos resultados con los que estuviese satisfecho y que finalmente se adaptasen a los objetivos que buscaba, ha sido ir paso a paso uno a uno probando, mezclando, entrenado, entendiendo y leyendo mucho sobre cada uno de los cinco modelos de clasificación que he utilizado. Con cada modelo he aprendido algo distinto, me he encontrado con dificultades y puntos muertos, he probado, mezclado experimentado y llegado a distintas conclusiones. Lo que puedo asegurar es que la búsqueda de la efectividad es compleja, difícil y muy complicada. Dos de los cinco modelos han dado buenas métricas y creo que han respondido bien a mi objetivo. Mi decisión final ha estado entre un algoritmo de regresión y un algoritmo de Random forest. Finalmente, empujado por un AUC superior que implica una mejor explicación del rendimiento del modelo, me he decidido por el Modelo de Clasificación Random Forest.

Sin embargo, la utilización del Random Forest no solo obedece a una mejores métricas obtenidas sino también al aprovechamiento de la baja correlación entre las variables y a una simplificación del estudio al no ser estrictamente necesarias técnicas de normalización o de reducción de dimensionalidad.

Random forest es la suma de muchos Arboles de decisión individuales lo que le hace mucho más fuerte y contundente en el resultado. Adicionalmente suele tener mejores rendimientos en algoritmos de clasificación como es nuestro caso. Por último, su fácil interpretación y explicación basada en divisiones.

- Dificultades encontradas.

A lo largo de la construcción del modelo he tenido problemas de convergencia en los modelos, he ido probando caminos, la mayoría de ellos sin llegar a ninguna parte, he entrenado los modelos con más y menos variables, por intuición, solo por testar esos escenarios, llegando la mayoría de las veces a puntos muertos y métricas que no me gustaban.

Me he encontrado con dificultades relacionadas con la base de datos con la que trabajar. Gracias al estudio pormenorizado del Data Frame en el EDA, intuyendo que pudieran ser necesarias el tenerlas diferenciadas, durante la fase de preprocessing generé hasta tres bases de datos distintas; Normalizada, con PCA y con Label Encoder. Estas tres bases de datos las he utilizado en cada uno de los modelos llegando a la conclusión que para nuestro modelo final elegido Random Forest, utilizar simplemente la base de datos original con Label Encoder aplicado era suficiente y daba iguales métricas o muy parecidas que con la implementación de las otras dos bases.

Pero quizás, los principales problemas detectados en el estudio del modelo fueron el gran desbalanceo del data set 87%-13% y overfitting, motivado por la dimensión de la base de datos. Ambos problemas se han podido resolver mediante la utilización de hiperparametros sugeridos por técnicas de Gridsearch y por la utilización de estimadores de desbalanceo dentro del propio algoritmo.

La combinación de todas estas herramientas y decisiones respecto a la base de datos finalmente han arrojado unas buenas métricas con las que quedo satisfecho.

- Métricas

Para la elección de las métricas que iban a ser utilizadas como explicativas del modelo, inicialmente estudié y entendí la teoría de la matriz de confusión. Finalmente, las métricas elegidas para evaluar los modelos de clasificación fueron Recall y AUC - Roc Curve.

La métrica de Recall nos va a informar sobre la cantidad de verdaderos positivos que el modelo es capaz de identificar. Qué porcentaje de los clientes están realmente interesados y somos capaces de identificar.

Por último, la Curva ROC-AUC que nos informa y verifica el rendimiento del modelo.

- Matriz de Confusión

En la diagonal de aciertos de nuestro modelo finalmente tenemos 69274 Verdaderos positivos y 4347 verdaderos negativos. En nuestro modelo prima la detección de los clientes que sí son susceptibles de contratar el seguro. Necesitamos obtener una buena tasa de verdaderos positivos pero tampoco nos importaba obtener verdaderos negativos. Si detectamos a los clientes que probablemente vayan a contratar un seguro de hogar o que no lo vayan a hacer, podremos trabajar de forma más directa. Por lo tanto, podemos obtener los mismos resultados por ambas vías. Todo ello con el objetivo de optimizar el tiempo y la eficiencia comercial. Los resultados finales obtenidos Recall 85% y un 88% en AUC son óptimos.

- Limpieza y Preprocesado

Técnicas de limpieza y preprocesado se hacen muy necesarias. En primer lugar la detección de campos vacíos y la decisión de qué hacer con ellos. Eliminarlos o rellenarlos y con qué datos hacerlo. Se pueden implementar distintas formas de preprocesado como hemos hecho. Técnicas de reducción de dimensionalidad, técnicas de normalización y aun así no ser necesarias. Lo que he confirmado es que la utilización final de estas técnicas depende no solo de la base de datos, su dimensión o tipología sino también del modelo final de Machine Learning utilizado.

- Base de datos

El modelo podría mejorarse sustancialmente si se dispusiesen de otras muchas variables económicas, sociales, financieras, comerciales, etc. Como he comentado en el estado del arte, existen muchísimos modelos ya implementados dentro de una entidad financiera. Modelos de riesgo, de comercialización, de morosidad, de recursos humanos... En concreto este modelo que he querido replicar, tengo conocimiento de la existencia de más de 1000 variables distintas para llegar a la decisión de si habilitarlo en una parrilla de llamadas o no. Actualmente se están llegando a tasas de conversión de 1 sobre 4, esto es un 25%. Porcentaje que, si bien siempre quiere ser aumentado, supone realmente un gran éxito.

- Comerciales

La importancia de detección de clientes que realmente sean susceptibles de comprar supone no solo poder dirigir la llamada y optimizar el tiempo sino también afinar todo lo posible en el precio ofrecido para ganar cuota de mercado.

Es importante que los comerciales de ventas de una entidad financiera sepan y puedan entender como se ha desarrollado, de donde han salido las variables utilizadas y de porqué la decisión de llamar o no a un cliente. Esta transmisión de conocimiento puede ser muy útil para toda la fuerza de ventas de las oficinas bancarias. Dado mi reciente incorporación al mundo del Data Science, creo que una de mis labores puede ser la de servir de nexo de unión entre el mundo senior del Data Science y la realidad comercial de asesoramiento y venta directa a los clientes. Cada una de estas realidades, desconoce mutuamente las labores, procesos y dificultades con los que se encuentra en el día a día. Explicar a la red comercial bajo estos parámetros, los motivos y ventajas de utilizar los modelos seguramente sea sinónimo final de éxito.

## ## CODIGO MODELO ##

## MODELO CLASIFICACION RANDOM FOREST

Presentamos e importamos todas las librerías que vamos a necesitar a lo largo del modelo.

```
In [1]: import pandas as pd
import numpy as np

#Libreria Metricas
from sklearn.metrics import f1_score, recall_score, precision_score, accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import auc
from sklearn.model_selection import KFold
from sklearn.metrics import log_loss

#Librerias Visualizacion
import matplotlib.pyplot as plt
import pylab as pl
import seaborn as sns
from pylab import rcParams
from matplotlib import pyplot

#Librerias Modelos
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

Leemos y presentamos nuestra Base de Datos. Mostramos las 5 primeras filas, el tipo y nombre de las Columnas / variables que la componen. Las variables que se refieren a unidades de negocio no van a ser utilizadas en el modelo por lo que las transformamos a Objeto para una mejor comprensión y segmentación de las propias variables.

```
In [2]: # CAMBIAR SEGUN RUTA LOCAL DONDE ESTE UBICADO EL REPOSITORIO
ruta = '/Users/manue/TFM'
```

```
In [3]: df = pd.read_csv(ruta + '/Financial-Product-Sales-Forecast-Model/Cleanning & Merging/df_cleanned.csv')
```

```
In [4]: # Transformamos las variables de la unidad de negocio en objeto ya que no queremos utilizarlas.
df[["cliente", "gestor",
    'codigo_cartera', 'dz', 'oficina']] = df[["cliente", "gestor",
    'codigo_cartera', 'dz', 'oficina']].astype(object);
```

```
In [5]: df.head(5)
```

```
Out[5]:
```

	cliente	saldo_captacion	esta_carterizado	cliente_bbp	tipo_gestor	gestor	cartera_patron	codigo_cartera	digital_3_meses	camino_digital	...	edad	seg_valor	seg_recorrido	dz	oficina	lp_seg_vida	lp_seg_acc	lp_seg_salud	lp_seg_hogar	lp_seg_auto
0	1	1328106.49	SI	NO	ASESOR FINANCIERO	18287	ASESORAMIENTO FINANCIERO	14881	SI	COMPRADOR	...	69.0	ALTO	BAJO RECORRIDO	2	210	0	0	0	1	0
1	2	1142234.34	SI	NO	ASESOR FINANCIERO	18287	ASESORAMIENTO FINANCIERO	14881	SI	COMPRADOR	...	81.0	ALTO	ALTO RECORRIDO	2	210	0	0	0	0	0
2	3	1142234.34	SI	NO	ASESOR FINANCIERO	18287	TUTELA	26332	NO	SIN USO	...	75.0	ALTO	BAJO RECORRIDO	2	210	0	0	0	0	0
3	4	1340503.88	SI	NO	ASESOR FINANCIERO	41475	ASESORAMIENTO FINANCIERO	14204	NO	SIN USO	...	94.0	ALTO	BAJO RECORRIDO	2	210	0	0	0	0	0
4	5	1758517.70	SI	NO	ASESOR FINANCIERO	39000	ASESORAMIENTO FINANCIERO	14219	SI	COMPRADOR	...	64.0	ALTO	BAJO RECORRIDO	2	210	0	0	0	0	0

5 rows x 30 columns

## Primera Aproximacion al modelo

```
In [10]: # Predicciones
clfforest_first_approach.predict(X)
```

```
Out[10]: array([1, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [11]: # Dividimos el Data Frame en set de entrenamiento y Test. 80% Entrenamiento y 20% TEST
test_size = 0.20
seed = 7
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = test_size, random_state = seed)
```

```
In [12]: # Entrenamos
clfforest_first_approach.fit(X_train, y_train)
```

```
Out[12]: RandomForestClassifier()
```

```
In [13]: # Predicciones
clfforest_first_approach.predict(X_test)
```

```
Out[13]: array([0, 0, 1, ..., 0, 0, 0], dtype=int64)
```

```
In [14]: # Probabilidades de tener seguro de hogar
clfforest_first_approach.predict_proba(X_test)[:, 1]
```

```
Out[14]: array([0.06, 0.25, 0.65, ..., 0.03, 0.09, 0.34])
```

```
In [15]: # Resultados obtenidos Primera Aproximación
```

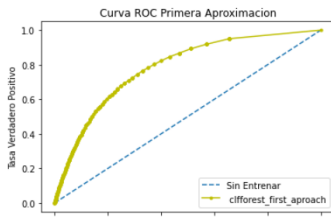
```
accuracy_score_first_approach = accuracy_score(y_test, clfforest_first_approach.predict(X_test))
precision_first_approach = precision_score(y_test, clfforest_first_approach.predict(X_test))
recall_first_approach = recall_score(y_test, clfforest_first_approach.predict(X_test))
f1_score_first_approach = f1_score(y_test, clfforest_first_approach.predict(X_test))
roc_auc_score_first_approach = roc_auc_score(y_test, clfforest_first_approach.predict_proba(X_test)[:, 1])

print('== Primera Aproximacion RandomForest ==')
print('-----')
print('accuracy_score =', accuracy_score_first_approach)
print('-----')
print('precision =', precision_first_approach)
print('-----')
print('recall =', recall_first_approach)
print('-----')
print('f1_score =', f1_score_first_approach)
print('-----')
print('roc_auc_score =', roc_auc_score_first_approach)

== Primera Aproximacion RandomForest ==
-----
accuracy_score = 0.856748823040709
-----
precision = 0.4119993862286537
-----
recall = 0.2278319898175647
-----
f1_score = 0.2934105562236354
-----
roc_auc_score = 0.7834543242566402
```

```
In [16]: # Ploteamos La Curva ROC
# Generamos un Clasificador sin entrenar
no_train = [0 for _ in range(len(X_test))]
# Calculamos AUC
ns_auc = roc_auc_score(y_test, no_train)
lr_auc = roc_auc_score(y_test, clfforest_first_approach.predict_proba(X_test)[:, 1])
# Print
print('Sin Entrenar: ROC AUC=%3f' % (ns_auc))
print('Tree: ROC AUC=%3f' % (lr_auc))
# Calculamos La Curva ROC
ns_fpr, ns_tpr, _ = roc_curve(y_test, no_train)
lr_fpr, lr_tpr, _ = roc_curve(y_test, clfforest_first_approach.predict_proba(X_test)[:, 1])
# Ploteamos
pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='Sin Entrenar')
pyplot.plot(lr_fpr, lr_tpr, markers='.', label='clfforest_first_approach', color='y')
# Etiquetas
pyplot.title('Curva ROC Primera Aproximacion')
pyplot.xlabel('Tasa Falso Positivo')
pyplot.ylabel('Tasa Verdadero Positivo')
pyplot.legend()
plt.savefig(ruta + '/Financial-Product-Sales-Forecast-Model/Images/RANDOMFOREST Model - Roc Curve First Approach.png', dpi=75, bbox_inches='tight')
plt.show()
pyplot.show()
```

Sin Entrenar: ROC AUC=0.500  
Tree: ROC AUC=0.783



### Construccion del Modelo

```
In [17]: # Read Data Frame
df_normalized = pd.read_csv(ruta + '/Financial-Product-Sales-Forecast-Model/Preprocessing/df_encoded.csv')
```

```
In [18]: df_normalized.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 417812 entries, 0 to 417811
Data columns (total 18 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   saldo_captacion        417812 non-null float64
 1   saldo_financiacion     417812 non-null float64
 2   edad                   417812 non-null float64
 3   seg_valor              417812 non-null int64
 4   saldo_ffii             417812 non-null int64
 5   saldo_plp              417812 non-null int64
 6   lp_dom_ingresos        417812 non-null int64
 7   lp_rbos                417812 non-null int64
 8   camino_digital         417812 non-null int64
 9   seg_recorrido          417812 non-null int64
10   marca_ctcte            417812 non-null int64
11   lp_seg_vida            417812 non-null int64
12   lp_seg_auto            417812 non-null int64
13   marca_bp              417812 non-null int64
14   lp_tjta_cto            417812 non-null int64
15   lp_tjt_rev             417812 non-null int64
16   lp_seg_acc             417812 non-null int64
17   lp_seg_hogar           417812 non-null int64
dtypes: float64(3), int64(15)
memory usage: 57.4 MB
```

```
In [19]: df_normalized.columns
```

```
Out[19]: Index(['saldo_captacion', 'saldo_financiacion', 'edad', 'seg_valor',
               'saldo_ffii', 'saldo_plp', 'lp_dom_ingresos', 'lp_rbos',
               'camino_digital', 'seg_recorrido', 'marca_ctcte', 'lp_seg_vida',
               'lp_seg_auto', 'marca_bp', 'lp_tjta_cto', 'lp_tjt_rev', 'lp_seg_acc',
               'lp_seg_hogar'],
              dtype='object')
```

```
In [20]: df.sample(5)
```

```
Out[20]:
```

	saldo_captacion	saldo_financiacion	saldo_ffii	saldo_plp	lp_dom_ingresos	lp_tjta_cto	lp_tjt_rev	lp_rbos	lp_of_int	edad	lp_seg_vida	lp_seg_acc	lp_seg_salud	lp_seg_hogar	lp_seg_auto
228079	1452.43	178716.85	0.0	0.00	1	1	0	1	1	52.0	0	0	0	1	0
213891	4164.25	12230.66	0.0	3707.61	1	1	1	1	1	53.0	1	0	0	0	0
384910	400.00	0.00	0.0	0.00	1	0	0	0	1	56.0	0	0	0	0	0

```
In [21]: # Inputs y Target
X = df_normalized.drop(['lp_seg_hogar'], axis = 1)
y = df_normalized['lp_seg_hogar']
print('Datos X =', X.size, X.shape)
print('Datos y =', y.size, y.shape)

Datos X = 7102804 (417812, 17)
Datos y = 417812 (417812,)
```

```
In [22]: # Dividimos el DF en set de entrenamiento y Test (80% train - 20% test)
test_size = 0.2
seed = 47
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = test_size, random_state = seed)
```

```
In [23]: # Shapes Train
X_train.shape, y_train.shape
```

```
Out[23]: ((334249, 17), (334249,))
```

```
In [24]: # Shapes Test
X_test.shape, y_test.shape
```

```
Out[24]: ((83563, 17), (83563,))
```

```
In [25]: # Instanciamos el Clasificador. Incorporamos un primer estimador para balancear el data set.
clfforest = RandomForestClassifier(class_weight = 'balanced')
```

```
In [26]: # Entrenamos
clfforest.fit(X_train, y_train)
```

```
Out[26]: RandomForestClassifier(class_weight='balanced')
```

```
In [27]: # Predicciones
clfforest.predict(X_test)
```

```
Out[27]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [28]: # Probabilidades de tener seguro de hogar
clfforest.predict_proba(X_test)[:, 1]
```

```
Out[28]: array([[0.11, 0.07, ..., 0.0154257,
                0.02, 0.07, ..., 0.0154257,
                ], ...])
```

```
In [29]: # Todas Las probabilidades
         clfforest.predict_proba(X_test)
```

```
Out[29]: array([[0.89, 0.11],
               [0.93, 0.07],
               [1., 0.],
               ...,
               [1., 0.],
               [0.9845743, 0.0154257],
               [0.98, 0.02]])
```

**Primeros resultados.** Los resultados obtenidos en el entrenamiento y test utilizando la métrica Recall como medida muestran que el modelo no está generalizando bien. Hay una situación grave de sobreajuste que debo resolver. Lo confirmaremos mediante la Validación Cruzada.

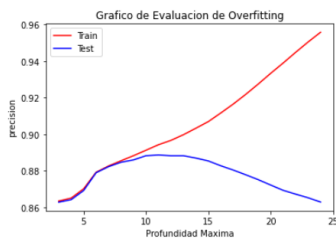
```
In [30]: print('=====RECALL TRAIN=====')
         print('recall =', recall_score(y_train,clfforest.predict(X_train)))
         print('=====RECALL TEST=====')
         print('recall =', recall_score(y_test,clfforest.predict(X_test)))
```

```
=====RECALL TRAIN=====
recall = 0.9995836620212113
=====RECALL TEST=====
recall = 0.38014654570830425
```

```
In [31]: # Grafico de Evaluacion de Overfitting
         # Se confirma con esta visualizacion que a partir de 8 profundidades el modelo empieza a generar overfitting
         train_prec = []
         eval_prec = []
         max_deep_list = list(range(3, 25))

         for deep in max_deep_list:
             arbol3 = DecisionTreeClassifier(criterion='gini', max_depth=deep)
             arbol3.fit(X_train, y_train)
             train_prec.append(arbol3.score(X_train, y_train))
             eval_prec.append(arbol3.score(X_test, y_test))

         # Plotamos:
         plt.plot(max_deep_list, train_prec, color='r', label='Train')
         plt.plot(max_deep_list, eval_prec, color='b', label='Test')
         plt.title('Grafico de Evaluacion de Overfitting')
         plt.legend()
         plt.ylabel('precision')
         plt.xlabel('Profundidad Maxima')
         plt.savefig(ruta + "/Finacial-Product-Sales-Forecast-Model/Images/RANDOMFOREST Model - Evaluacion de Overfitting.png", dpi=75, bbox_inches='tight')
```



## Técnicas de validación cruzada. Métricas: Validación cruzada

Utilizamos la validación cruzada iterando 5 veces a lo largo del conjunto de entrenamiento.

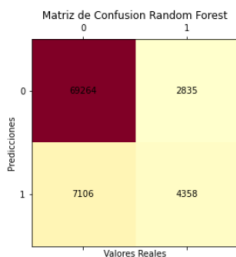
- Apartamos 1/5 muestras.
- Entrenamos el modelo con las 4/5 muestras restantes.
- Mediremos con diferentes métricas el resultado obtenido en las muestras apartadas.
- Esto significa que hacemos 5 entrenamientos independientes.
- El resultado será la media de las 5 métricas anteriores.

```
In [32]: from sklearn.model_selection import cross_val_score
         cross_val_score_precision = cross_val_score(clfforest,X,y,cv=5,scoring="precision").mean()
         cross_val_score_recall = cross_val_score(clfforest,X,y,cv=5,scoring="recall").mean()
         cross_val_score_f1 = cross_val_score(clfforest,X,y,cv=5,scoring="f1").mean()
         cross_val_score_precision, cross_val_score_recall, cross_val_score_f1
```

```
Out[32]: (0.6026682787464982, 0.3780910683012259, 0.4637312214035411)
```

```
In [33]: # Plot Matriz de Confusion
         # Plot Matriz de Confusion
         matriz_confusion_clfforest = confusion_matrix(y_test,clfforest.predict(X_test))
         fig, ax = plt.subplots(figsize=(4, 7))
         ax.matshow(matriz_confusion_clfforest, cmap=plt.cm.YlOrRd)
         for i in range(matriz_confusion_clfforest.shape[0]):
             for j in range(matriz_confusion_clfforest.shape[1]):
                 ax.text(x=j, y=i, s=matriz_confusion_clfforest[i, j], va='center', ha='center')

         plt.title('Matriz de Confusion Random Forest')
         plt.xlabel('Valores Reales')
         plt.ylabel('Predicciones')
         plt.tight_layout()
         plt.savefig(ruta + "/Finacial-Product-Sales-Forecast-Model/Images/RANDOMFOREST Model - Confusion_Matrix_Random Forest.png", dpi=75, bbox_inches='tight')
         plt.show()
```





```
In [34]: # Resultados

accuracy_score_clfforest = accuracy_score(y_test,clfforest.predict(X_test))
precision_clfforest = precision_score(y_test,clfforest.predict(X_test))
recall_clfforest = recall_score(y_test,clfforest.predict(X_test))
f1_score_clfforest = f1_score(y_test,clfforest.predict(X_test))
roc_auc_score_clfforest = roc_auc_score(y_test,clfforest.predict_proba(X_test)[:, 1])

print("==== Random Forest RESULTADOS ====")
print('-----')
print('accuracy_score =', accuracy_score_clfforest)
print('-----')
print('precision =', precision_clfforest)
print('-----')
print('recall =', recall_clfforest)
print('-----')
print('f1_score =', f1_score_clfforest)
print('-----')
print('roc_auc_score =', roc_auc_score_clfforest)

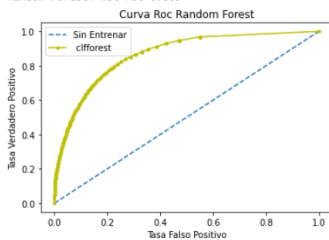
==== Random Forest RESULTADOS ====
accuracy_score = 0.88103586515631
-----
precision = 0.6058668149589879
-----
recall = 0.38014654570830425
-----
f1_score = 0.4671704990804151
-----
roc_auc_score = 0.8647433622250448
```

## Ploteamos Curva ROC

```
In [35]: # Ploteamos La Curva ROC
# Generamos un Clasificador sin entrenar
no_train = [0 for _ in range(len(X_test))]
# Calculamos AUC
ns_auc = roc_auc_score(y_test, no_train)
lr_auc = roc_auc_score(y_test, clfforest.predict_proba(X_test)[:, 1])
# Print
print('Sin Entrenar : ROC AUC=%3f' % (ns_auc))
print('Random Forest: ROC AUC=%3f' % (lr_auc))
# Calculamos La Curva Roc
ns_fpr, ns_tpr, _ = roc_curve(y_test, no_train)
lr_fpr, lr_tpr, _ = roc_curve(y_test, clfforest.predict_proba(X_test)[:, 1])
# Ploteamos
pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='Sin Entrenar')
pyplot.plot(lr_fpr, lr_tpr, marker='.', label='clfforest', color = "y")
# Etiquetas
pyplot.title('Curva Roc Random Forest')
pyplot.xlabel('Tasa Falso Positivo')
pyplot.ylabel('Tasa Verdadero Positivo')
pyplot.legend()
plt.savefig(ruta + '/Financial-Product-Sales-Forecast-Model/Images/RANDOMFOREST- Roc Curve.png', dpi=75, bbox_inches='tight')
plt.show()
pyplot.show()
```

Sin Entrenar : ROC AUC=0.500

Random Forest: ROC AUC=0.865



## GridSearchCV

Permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicándole un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los segundos mediante validación cruzada.

```
In [36]: from sklearn.model_selection import GridSearchCV
```

```
In [37]: # Sugérimos la búsqueda de los mejores parametros
```

```
forest_grid_params = {
    'n_estimators': [100,125],
    'max_features': ["auto", "log2"],
    'criterion': ['gini', 'entropy'],
    'max_depth': [2,4,6,8,10],
    'min_samples_split': [2,4],
    'min_samples_leaf': [15,20]
}

forestgs = GridSearchCV(
    clfforest,
    forest_grid_params,
    verbose = 1,
    cv = 3,
    n_jobs = -1
)
```

```
In [38]: forestgs.fit(X_train,y_train)
```

Fitting 3 folds for each of 160 candidates, totalling 480 fits

[Parallel(n\_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.

[Parallel(n\_jobs=-1)]: Done 26 tasks | elapsed: 56.3s

[Parallel(n\_jobs=-1)]: Done 176 tasks | elapsed: 9.0min

[Parallel(n\_jobs=-1)]: Done 426 tasks | elapsed: 23.6min

[Parallel(n\_jobs=-1)]: Done 480 out of 480 | elapsed: 27.8min finished

```
Out[38]: GridSearchCV(cv=3, estimator=RandomForestClassifier(class_weight='balanced'),
    n_jobs=-1,
    param_grid={'criterion': ['gini', 'entropy'],
    'max_depth': [2, 4, 6, 8, 10],
    'max_features': ['auto', 'log2'],
    'min_samples_leaf': [15, 20],
    'min_samples_split': [2, 4],
    'n_estimators': [100, 125]},
    verbose=1)
```

```
In [39]: # Mejores Estimadores
         forestgs.best_estimator_

Out[39]: RandomForestClassifier(class_weight='balanced', max_depth=10,
                               min_samples_leaf=15)
```

```
In [40]: # Mejores Parametros
         forestgs.best_params_

Out[40]: {'criterion': 'gini',
          'max_depth': 10,
          'max_features': 'auto',
          'min_samples_leaf': 15,
          'min_samples_split': 2,
          'n_estimators': 100}
```

## Construcción del Modelo con los Mejores Parámetros atendiendo a GridSearch

```
In [41]: clfforest_gs = RandomForestClassifier(class_weight = 'balanced', criterion = 'gini',
          max_depth = 10, max_features = 'auto', min_samples_split = 2,
          n_estimators = 100, min_samples_leaf = 15 )

In [42]: clfforest_gs.fit(X_train, y_train)

Out[42]: RandomForestClassifier(class_weight='balanced', max_depth=10,
                               min_samples_leaf=15)

In [43]: clfforest_gs.predict(X_test)

Out[43]: array([1, 1, 0, ..., 0, 0, 0], dtype=int64)

In [44]: clfforest_gs.predict_proba(X_test)[: , 1]

Out[44]: array([0.57580347, 0.56123645, 0.05530982, ..., 0.07895772, 0.03462971,
                0.18769305])
```

**Evaluación del overfitting** Los resultados obtenidos del entrenamiento y test usando la métrica recall como medida muestran que el overfitting ha sido resuelto usando mejores parámetros. Confirmado por la Validación Cruzada.

```
In [45]: print('=====RECALL TRAIN=====')
         print('recall =', recall_score(y_train,clfforest_gs.predict(X_train)))
         print('=====RECALL TEST=====')
         print('recall =', recall_score(y_test,clfforest_gs.predict(X_test)))

         =====RECALL TRAIN=====
         recall = 0.8671224471908142
         =====RECALL TEST=====
         recall = 0.8573796231681786
```

## Cross validation

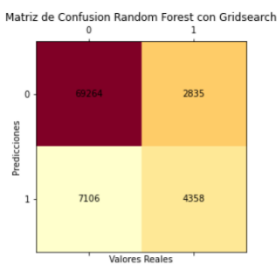
```
In [46]: from sklearn.model_selection import cross_val_score
         cross_val_score_precision = cross_val_score(clfforest_gs,X,y,cv=5,scoring="precision").mean()
         cross_val_score_recall = cross_val_score(clfforest_gs,X,y,cv=5,scoring="recall").mean()
         cross_val_score_f1 = cross_val_score(clfforest_gs,X,y,cv=5,scoring="f1").mean()
         cross_val_score_precision , cross_val_score_recall, cross_val_score_f1

Out[46]: (0.36047829872767195, 0.8604203152364274, 0.5082430463419795)
```

## Matriz de Confusion

```
In [47]: matriz_confusion_clfforest_gs = confusion_matrix(y_test,clfforest_gs.predict(X_test))
         fig, ax = plt.subplots(figsize=(4, 7))
         ax.imshow(matriz_confusion_clfforest_gs, cmap=plt.cm.YlOrRd)
         for i in range(matriz_confusion_clfforest_gs.shape[0]):
             for j in range(matriz_confusion_clfforest_gs.shape[1]):
                 ax.text(x=j, y=i, s=matriz_confusion_clfforest_gs[i, j], va='center', ha='center')

         plt.title('Matriz de Confusion Random Forest con Gridsearch')
         plt.xlabel('Valores Reales')
         plt.ylabel('Predicciones')
         plt.tight_layout()
         plt.savefig(ruta + '/Financial-Product-Sales-Forecast-Model/Images/RANDOMFOREST Model - Confusion_Matrix_Gridsearch.png', dpi=75,bbox_inches='tight')
         plt.show()
```



```
In [48]: # Resultados

         accuracy_score_clfforest_gs = accuracy_score(y_test,clfforest_gs.predict(X_test))
         precision_clfforest_gs = precision_score(y_test,clfforest_gs.predict(X_test))
         recall_clfforest_gs = recall_score(y_test,clfforest_gs.predict(X_test))
         f1_score_clfforest_gs = f1_score(y_test,clfforest_gs.predict(X_test))
         roc_auc_score_clfforest_gs = roc_auc_score(y_test,clfforest_gs.predict_proba(X_test)[: , 1])

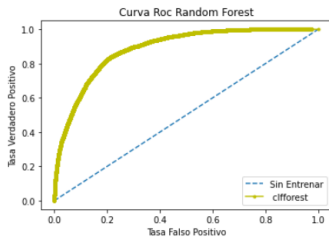
         print('==== Random Forest RESULTADOS FINALES ====')
         print('-----')
         print('accuracy_score =', accuracy_score_clfforest_gs )
         print('-----')
         print('precision =',precision_clfforest_gs)
         print('-----')
         print('recall =', recall_clfforest_gs)
         print('-----')
         print('f1_score =', f1_score_clfforest_gs)
         print('-----')
         print('roc_auc_score =', roc_auc_score_clfforest_gs)

         ===== Random Forest RESULTADOS FINALES =====
         accuracy_score = 0.7749362756243792
         precision = 0.3640235546831599
         -----
         recall = 0.8573796231681786
         -----
         f1_score = 0.5110620044196023
         -----
         roc_auc_score = 0.8868846784264333
```

## Curva Roc

```
In [49]: # Ploteamos La Curva ROC
# Generamos un Clasificador sin entrenar
no_train = [0 for _ in range(len(X_test))]
# Calculamos AUC
ns_auc = roc_auc_score(y_test, no_train)
lr_auc = roc_auc_score(y_test, clfforest_gs.predict_proba(X_test)[: , 1])
# Print
print('Sin Entrenar : ROC AUC=%3f' % (ns_auc))
print('Random Forest: ROC AUC=%3f' % (lr_auc))
# Calculamos La Curva Roc
ns_fpr, ns_tpr, _ = roc_curve(y_test, no_train)
lr_fpr, lr_tpr, _ = roc_curve(y_test, clfforest_gs.predict_proba(X_test)[: , 1])
# Ploteamos
pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='Sin Entrenar')
pyplot.plot(lr_fpr, lr_tpr, marker='.', label='clfforest', color = "y")
# Etiquetas
pyplot.title('Curva Roc Random Forest')
pyplot.xlabel('Tasa Falso Positivo')
pyplot.ylabel('Tasa Verdadero Positivo')
pyplot.legend()
plt.savefig(ruta + '/Financial-Product-Sales-Forecast-Model/Images/RANDOMFOREST- Roc Curve GridSearch.png', dpi=75, bbox_inches='tight')
plt.show()
pyplot.show()
```

Sin Entrenar : ROC AUC=0.500  
Random Forest: ROC AUC=0.887



## Evolucion de Resultados

```
In [50]: print('Resultados 1ª Aproximacion: RECALL %f, AUC %f' %(recall_first_aproach, roc_auc_score_first_aproach))
print('Resultados model: RECALL %f, AUC %f' %(recall_clfforest, roc_auc_score_clfforest))
print('Resultados model con gridsearch: RECALL %f, AUC %f' %(recall_clfforest_gs, roc_auc_score_clfforest_gs))

Resultados 1ª Aproximacion: RECALL 0.227832, AUC 0.783454
Resultados model: RECALL 0.380147, AUC 0.864743
Resultados model con gridsearch: RECALL 0.857380, AUC 0.886885
```

```
In [199]: import pickle
pickle_out = open(ruta + '/Financial-Product-Sales-Forecast-Model/Frontend/clfforest_gs.pkl', mode = "wb")
pickle.dump(clfforest_gs, pickle_out)
pickle_out.close()
```

## Testing

```
In [52]: X_new = pd.DataFrame({'edad':[45], 'saldo_captacion':[1000], 'saldo_financiacion':[100],
'seg_valor':[1], 'seg_recorrido':[3], 'marca_ccte':[1], 'camino_digital':[0], 'marca_bp':[1],
'saldo_ffii':[1], 'saldo_plp':[1], 'lp_dom_ingresos':[1], 'lp_rbos':[0], 'lp_seg_vida':[1],
'lp_seg_auto':[1], 'lp_seg_acc':[1], 'lp_tjta_cto':[1], 'lp_tjt_rev':[0],})
```

```
In [53]: prediccion = clfforest_gs.predict(X_new)
```

```
In [54]: prediccion
```

```
Out[54]: array([0], dtype=int64)
```

```
In [55]: probabilidades = clfforest_gs.predict_proba(X_new)
```

```
In [56]: if prediccion == 1:
print('Llama')
print('Probabilidad de que si contrate: ', probabilidades[: , 1])
else:
print('No llames')
print('Probabilidad de que no contrate: ', probabilidades[: , 0])
```

No llames  
Probabilidad de que no contrate: [0.57186477]