# Predicción de pérdida de clientes bancarios

Bank Customer Churn Prediction

# Data Science

**Comisión 46295**
**Profesor: Jorge Ruiz**
**Tutor: Gabriel Gutiérrez Mas**

# CONTENIDO

## Introducción

El presente trabajo intentará desarrollar un modelo que pueda predecir cuál es la probabilidad de abandono de los clientes de un banco. Con la ayuda del Machine Learnig, y con los datos que se presentarán para su análisis, se creará un modelo que buscará patrones que permitan determinar con la mayor certeza la tasa de abandono. De esta manera se podrá ofrecer a los usuarios o interesados una herramienta más que permita tomar decisiones ante determinadas situaciones para prevenir el abandono.

## Definición de Churn

El "Churn de clientes bancarios" se refiere a la tasa de rotación o pérdida de clientes en una entidad bancaria. Es un indicador muy importante y que se puede también aplicar a distintas empresas de servicios. Mide la cantidad de clientes que dejan de utilizar los servicios de un banco en un período de tiempo determinado.
Es relevante para las instituciones financieras, ya que las afecta directamente en su rentabilidad y éxito a largo plazo. Afecta la permanencia de las empresas en determinados mercados. (GPT, s.f.)

## Contexto y problemas comerciales

A diferencia de Argentina, el contexto actual de los bancos en España, Alemania y Francia puede variar, pero en general, los tres países tienen sistemas financieros muy sólidos.
Por hacer una comparación, los bancos en Argentina experimentan situaciones como la alta inflación, el riesgo crediticio o la competencia por captar depósitos de sus clientes.
Es importante destacar que al momento de revisar estos modelos se tendrían que consultar distintas fuentes financieras relacionadas a la industria para poder evaluar cambios como nuevas regulaciones bancarias. (se puede ampliar)

# Objetivos e Hipótesis

## Objetivo

El objetivo principal es desarrollar un modelo predictivo, que permita usando determinadas variables, sugerir medidas para evitar el abandono de los clientes de un banco.  (a revisar)
El objetivo secundario es que sirva de referencia para otras empresas de servicios similares. (a revisar)

## Hipótesis principal

“Los que abandonan el banco son los clientes que tienen menor edad y no tienen productos activos como tarjetas de crédito”.
Es decir  son clientes inactivos. Se suma a ello, que son mujeres y tienen los salarios más bajos.

Se irán generando preguntas durante el desarrollo que puedan ayudar a resolver el ¿por qué los clientes abandonan el banco de acuerdo a la información que brinda el data set?

# Desarrollo

## Obtención de datos

La base seleccionada es Churn_Modelling_archivo de Kaggle.
El data frame tiene 1000 filas con 14 atributos, de los cuales se determinarán cuáles son los más relevantes.

## Variables Originales

Las variables consideradas por el momento son 'CreditScore', 'Gender', 'Age', 'Tenure','NumOfProducts', 'HasCrCard','IsActiveMember', 'EstimatedSalary', 'Exited'.

## Variables Modificadas

Se modificaron aquellas variables que era de texto a número para facilitar el análisis quedando las siguientes:

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited', 'Gender_modificado',
       'Geography_modificado'],
      dtype='object')
```

## Análisis de datos

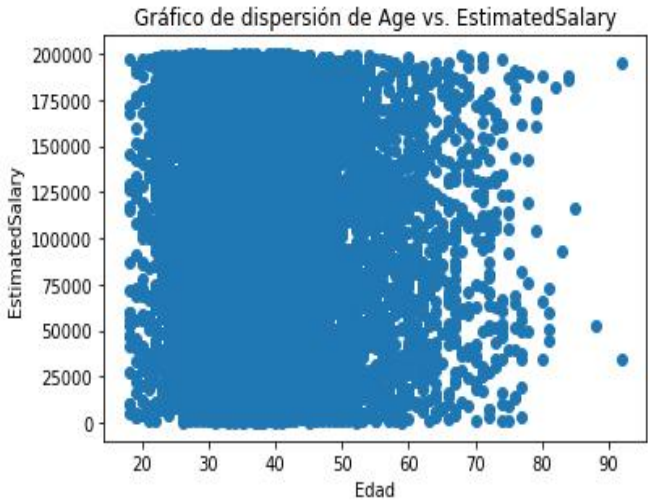
A partir de esta etapa se comienza el análisis del Df(dataframe) con las variables que se plantearon inicialmente.
Se irá manipulando el df y generando visualizaciones explicando los avances para la problemática planteada.

## Desarrollando el modelo

El objetivo se dijo es lograr predecir que clientes pueden abandonar un banco.

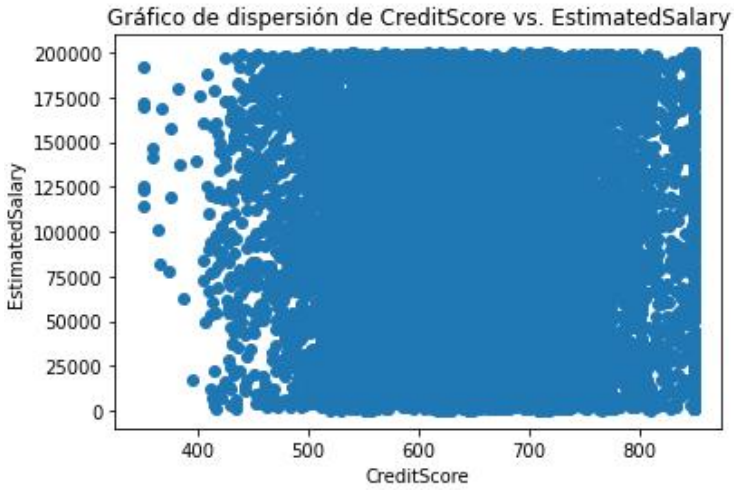
### *Edad vs Salario*

Se comienza el análisis tratando de relacionar Edad vs Salario con un gráfico con dispersión de cual no puedo sacar muchas conclusiones, salvo que se concentran los salarios estimados entre la edad de 20 hasta los 60 como ocurre en varios países.




## *Score vs Salario*

Continúa el análisis tratando de relacionar Score Vs Salario. Como se esperaba se pude afirmar que, a mayor score, mayor salario o viceversa. De todas maneras, se necesita seguir ampliando la información.




## *Edad*

Si bien es más apropiado cuánto mayor es la cantidad de datos a analizar para obtener conclusiones, por ahora se va a comenzar de menor a mayor para facilitar la interpretación de los datos. Me interesaba conocer las edades de los clientes. Como se vio anteriormente en los gráficos, las edades se concentran entre los 30 y 50 años.

| | Edad | Cantidad |
|---|---|---|
| 0 | 37 | 478 |
| 1 | 38 | 477 |
| 2 | 35 | 474 |
| 3 | 36 | 456 |
| 4 | 34 | 447 |
| 5 | 33 | 442 |
| 6 | 40 | 432 |
| 7 | 39 | 423 |
| 8 | 32 | 418 |
| 9 | 31 | 404 |
| 10 | 41 | 366 |
| 11 | 29 | 348 |
| 12 | 30 | 327 |
| 13 | 42 | 321 |
| 14 | 43 | 297 |
| 15 | 28 | 273 |
| 16 | 44 | 257 |
| 17 | 45 | 229 |
| 18 | 46 | 226 |
| 19 | 27 | 209 |
| 20 | 26 | 200 |
| 21 | 47 | 175 |
| 22 | 48 | 168 |
| 23 | 25 | 154 |
| 24 | 49 | 147 |
| 25 | 50 | 134 |
| 26 | 24 | 132 |
| 27 | 51 | 119 |

También vemos aisladas algunas edades entre 82 y 92 años.

```
Out[5]: 37    478
        38    477
        35    474
        36    456
        34    447
              ...
        92      2
        88      1
        82      1
        85      1
        83      1
        Name: Age, Length: 70, dtype: int64
```

Algunas estadísticas descriptivas sobre el DF para facilitar la lectura de los datos y nos concentramos en la edad, ampliando las conclusiones que se dejaron anteriormente.

```
In [6]: #Algunas algunas estadísticas descriptivas sobre el DataFrame para facilitar la lectura de los datos
        df_Churn_Modelling_archivo.describe()
```

Out[6]:

| | RowNumber | CustomerId | CreditScore | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 10000.00000 | 1.000000e+04 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | 10000.000000 | 10000.000000 |
| mean | 5000.50000 | 1.569094e+07 | 650.528800 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 |
| std | 2886.89568 | 7.193619e+04 | 96.653299 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 |
| min | 1.00000 | 1.556570e+07 | 350.000000 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 |
| 25% | 2500.75000 | 1.562853e+07 | 584.000000 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 51002.110000 |
| 50% | 5000.50000 | 1.569074e+07 | 652.000000 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.00000 | 1.000000 | 100193.915000 |
| 75% | 7500.25000 | 1.575323e+07 | 718.000000 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.00000 | 1.000000 | 149388.247500 |
| max | 10000.00000 | 1.581569e+07 | 850.000000 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.00000 | 1.000000 | 199992.480000 |

## *Rangos de edad*

Decido filtrar por rangos a partir de los resultados del gráfico de dispersión sobre Edad vs Score. En este caso entre 20 a 59 que concentra la mayor cantidad:

| | Edad | Cantidad |
|---|---|---|
| 0 | 37 | 478 |
| 1 | 38 | 477 |

| | | |
|---|---|---|
| 2 | 35 | 474 |
| 3 | 36 | 456 |
| 4 | 34 | 447 |
| 5 | 33 | 442 |
| 6 | 40 | 432 |
| 7 | 39 | 423 |
| 8 | 32 | 418 |
| 9 | 31 | 404 |
| 10 | 41 | 366 |
| 11 | 29 | 348 |
| 12 | 30 | 327 |
| 13 | 42 | 321 |
| 14 | 43 | 297 |
| 15 | 28 | 273 |
| 16 | 44 | 257 |
| 17 | 45 | 229 |
| 18 | 46 | 226 |
| 19 | 27 | 209 |
| 20 | 26 | 200 |
| 21 | 47 | 175 |
| 22 | 48 | 168 |
| 23 | 25 | 154 |
| 24 | 49 | 147 |
| 25 | 50 | 134 |
| 26 | 24 | 132 |
| 27 | 51 | 119 |
| 28 | 52 | 102 |
| 29 | 23 | 99 |
| 30 | 54 | 84 |
| 31 | 22 | 84 |
| 32 | 55 | 82 |
| 33 | 57 | 75 |
| 34 | 53 | 74 |
| 35 | 56 | 70 |
| 36 | 58 | 67 |
| 37 | 59 | 62 |
| 38 | 21 | 53 |
| 39 | 20 | 40 |

Filtro por rango de edad para que solo traiga los de 30 ya que seguía siendo mucha información.

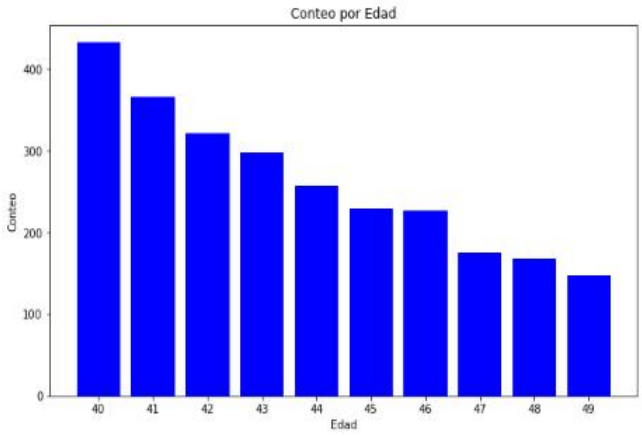
| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age |
|---|---|---|---|---|---|---|---|
| 3 | 4 | 15701354 | Boni | 699 | France | Female | 39 |
| 10 | 11 | 15767821 | Bearce | 528 | France | Male | 31 |
| 12 | 13 | 15632264 | Kay | 476 | France | Female | 34 |
| 14 | 15 | 15600882 | Scott | 635 | Spain | Female | 35 |
| 21 | 22 | 15597945 | Dellucci | 636 | Spain | Female | 32 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 9990 | 9991 | 15798964 | Nkemakonam | 714 | Germany | Male | 33 |
| 9992 | 9993 | 15657105 | Chukwualuka | 726 | Spain | Male | 36 |
| 9995 | 9996 | 15606229 | Obijiaku | 771 | France | Male | 39 |
| 9996 | 9997 | 15569892 | Johnstone | 516 | France | Male | 35 |
| 9997 | 9998 | 15584532 | Liu | 709 | France | Female | 36 |

| | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | \ |
|---|---|---|---|---|---|---|
| 3 | 1 | 0.00 | 2 | 0 | 0 | |
| 10 | 6 | 102016.72 | 2 | 0 | 0 | |
| 12 | 10 | 0.00 | 2 | 1 | 0 | |
| 14 | 7 | 0.00 | 2 | 1 | 1 | |
| 21 | 8 | 0.00 | 2 | 1 | 0 | |
| ... | ... | ... | ... | ... | ... | |
| 9990 | 3 | 35016.60 | 1 | 1 | 0 | |
| 9992 | 2 | 0.00 | 1 | 1 | 0 | |
| 9995 | 5 | 0.00 | 2 | 1 | 0 | |
| 9996 | 10 | 57369.61 | 1 | 1 | 1 | |
| 9997 | 7 | 0.00 | 1 | 0 | 1 | |

| | EstimatedSalary | Exited |
|---|---|---|
| 3 | 93826.63 | 0 |
| 10 | 80181.12 | 0 |
| 12 | 26260.98 | 0 |
| 14 | 65951.65 | 0 |
| 21 | 138555.46 | 0 |
| ... | ... | ... |
| 9990 | 53667.08 | 0 |
| 9992 | 195192.40 | 0 |
| 9995 | 96270.64 | 0 |
| 9996 | 101699.77 | 0 |
| 9997 | 42085.58 | 1 |

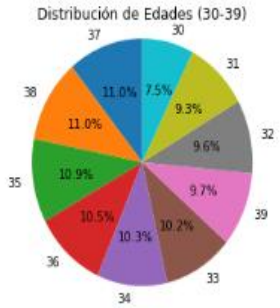
[4346 rows x 14 columns]

Luego se filtra por los de 40 años y se grafica como para ir teniendo un panorama:

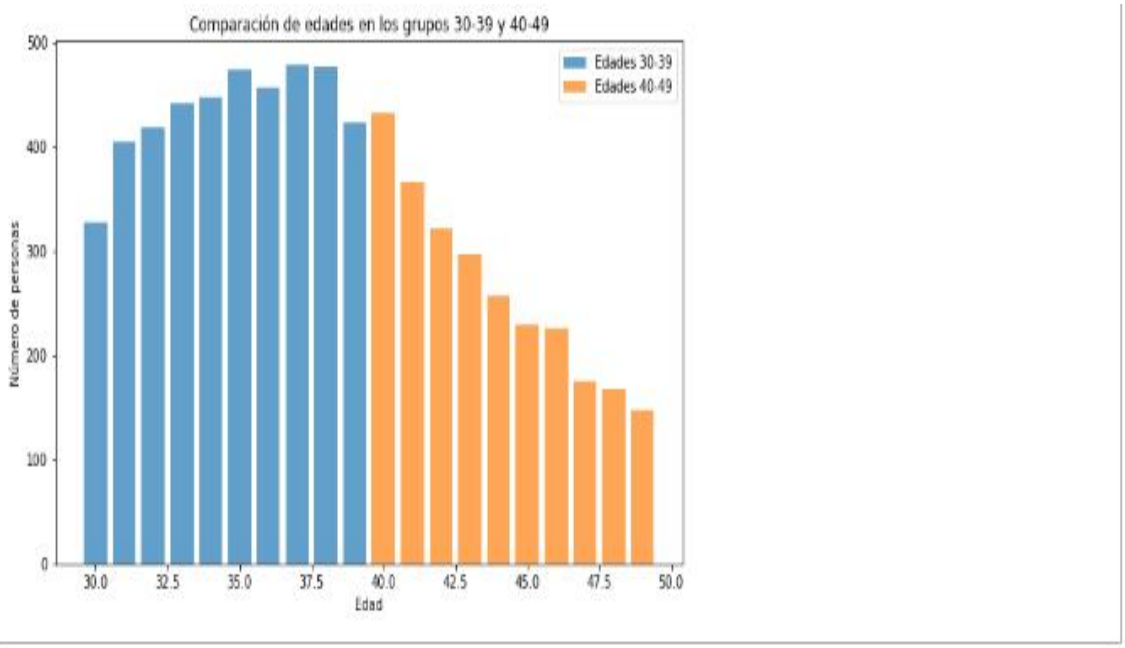



Ahora vuelvo a los de 30 años buscando más adelante ir haciendo una comparación:

```
In [8]: # Filtro por los de 30s para contar la cantidad que se repite de cada edad pero siempre con los de 30
Age_df_30 = df_Churn_Modelling_archivo[(df_Churn_Modelling_archivo['Age'] >= 30) & (df_Churn_Modelling_archivo['Age'] <= 39)]

# Contar las ocurrencias de cada categoría
age_counts = Age_df_30['Age'].value_counts()

# Crear el gráfico de torta con Matplotlib
plt.pie(age_counts, labels=age_counts.index, autopct='%1.1f%%', startangle=90)
plt.axis('equal')  # Hace que el gráfico sea un círculo
plt.title('Distribución de Edades (30-39)')
plt.show()
```



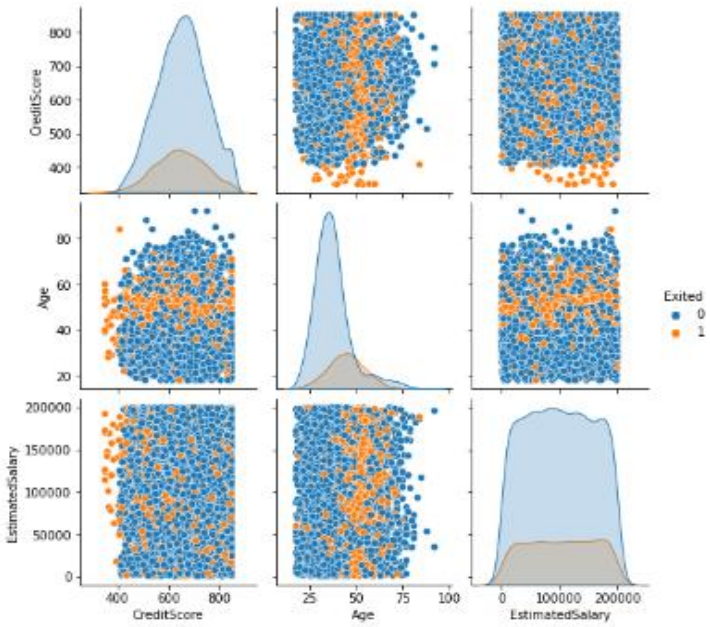

Ahora puedo comparar los de 30(azul) años y 40 años(naranja):




La conclusión más importante que obtengo de los análisis de las edades es que a medida que aumenta la edad bajan las cantidades de personas o clientes del banco.

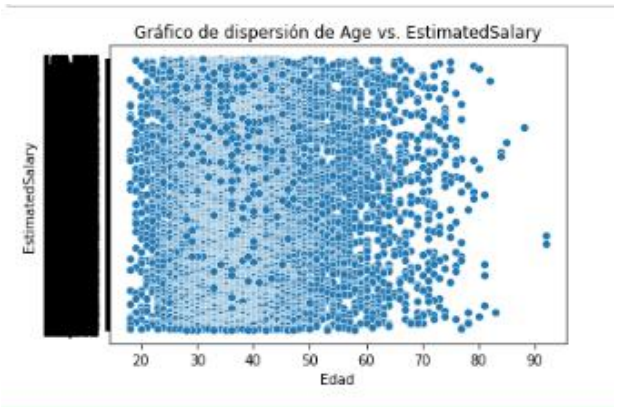
Ahora uso seaborn para mostrar lo que consideré más importante buscando relaciones, esto es Credit Score, Edad y Estiated Salary.

```
In [15]: #Uso seaborn para mostrar lo que consiré más importante  buscando relaciones, esto es Credit Score, Edad etc
         importantes = df_Churn_Modelling_archivo[['CreditScore','Age','EstimatedSalary','Exited']]

         sns.pairplot(importantes , hue = 'Exited')

Out[15]: <seaborn.axisgrid.PairGrid at 0x24ea89bb190>
```

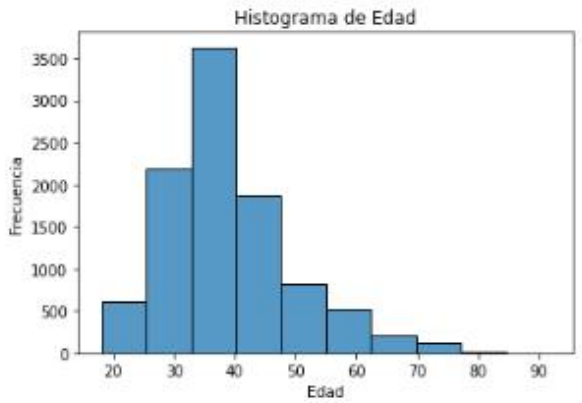



Como se puede observar, las posibilidades que brinda esta visualización es bastante amplia. Me permite entre otras cosas, inferir rápidamente cuáles son los clientes que abandonan el banco (Exited = 1), por Creditscore, por Edad(Age) y por EstimatedSalary. Por ejemplo, observo que los que tienen menor CreditScore son los que abandonan el banco, o los que tienen tienen menor EstimatedSalary,etc.
Como se vio anteriormente, la edad no sería un factor determinante de abandono del banco. Por lo cual, voy en busca de la/las variables.

**Sigo comparando la relación Edad vs Salary probando con sns(seaborn)**



Me interesa seguir probando sns ahora edad pero con un Histrograma buscando que me traiga la misma concentración de edades que en los gráficos anteriores




## *Género*

Necesito ver si el género influye, por lo cual lo verifico:

```
In [10]: #Acá vamos a ver ahora que cantidades tenemos de hombres y mujeres
         df_Churn_Modelling_archivo.describe(include=['O'])
```

Out[10]:

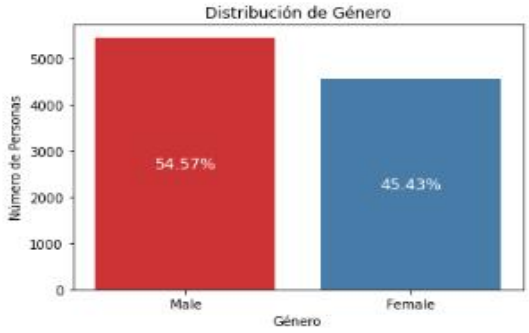
| | Surname | Geography | Gender |
|---|---|---|---|
| count | 10000 | 10000 | 10000 |
| unique | 2932 | 3 | 2 |
| top | Smith | France | Male |
| freq | 32 | 5014 | 5457 |

```
In [ ]: #Se observa que es mayor la cantidad de hombres, pero no sabemos en que cantidad
```

Se verifica que en el top están los hombres, pero no sabemos si es grande la diferencia por sobre las mujeres, así que se sigue desarrollando.

Se grafica por o cuál Mujeres vs Hombres:

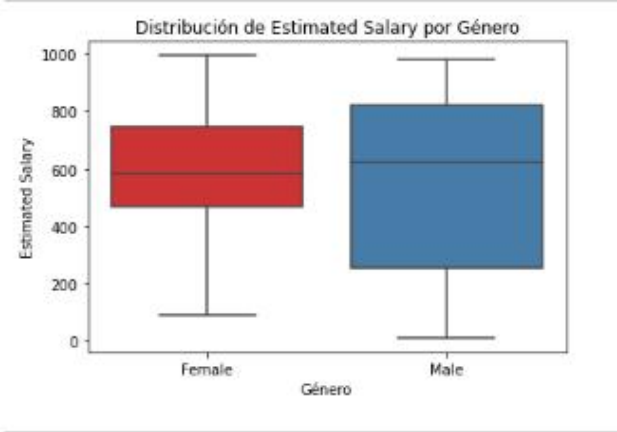
```
In [35]: # Muestra el porcentaje de hombres y mujeres
gender_counts = df_Churn_Modelling_archivo['Gender'].value_counts()

# Gráfico de barras
sns.barplot(x=gender_counts.index, y=gender_counts.values, palette="Set1")

# Los porcentajes en el centro de cada barra
total_personas = len(df_Churn_Modelling_archivo['Gender'])
for i, value in enumerate(gender_counts.values):
    percentage = (value / total_personas) * 100
    plt.text(i, value/2, f'{percentage:.2f}%', ha='center', va='center', color='white', fontsize=12)

# <gráfico
plt.xlabel('Género')
plt.ylabel('Número de Personas')
plt.title('Distribución de Género')

# Grafico
plt.show()
```



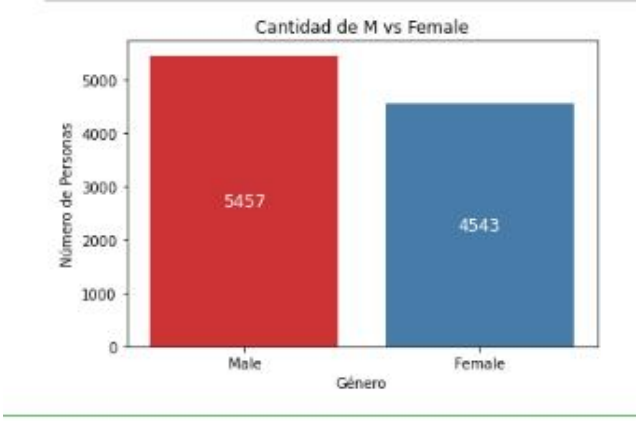

Vemos que la distribución en porcentajes de  Male vs Female es muy parecida, casi un 50% de c/u. Por lo cual, puedo inferir que el sexo no sería un determinante por ahora.

## *EstimatedSalary distribuidos por Gender*

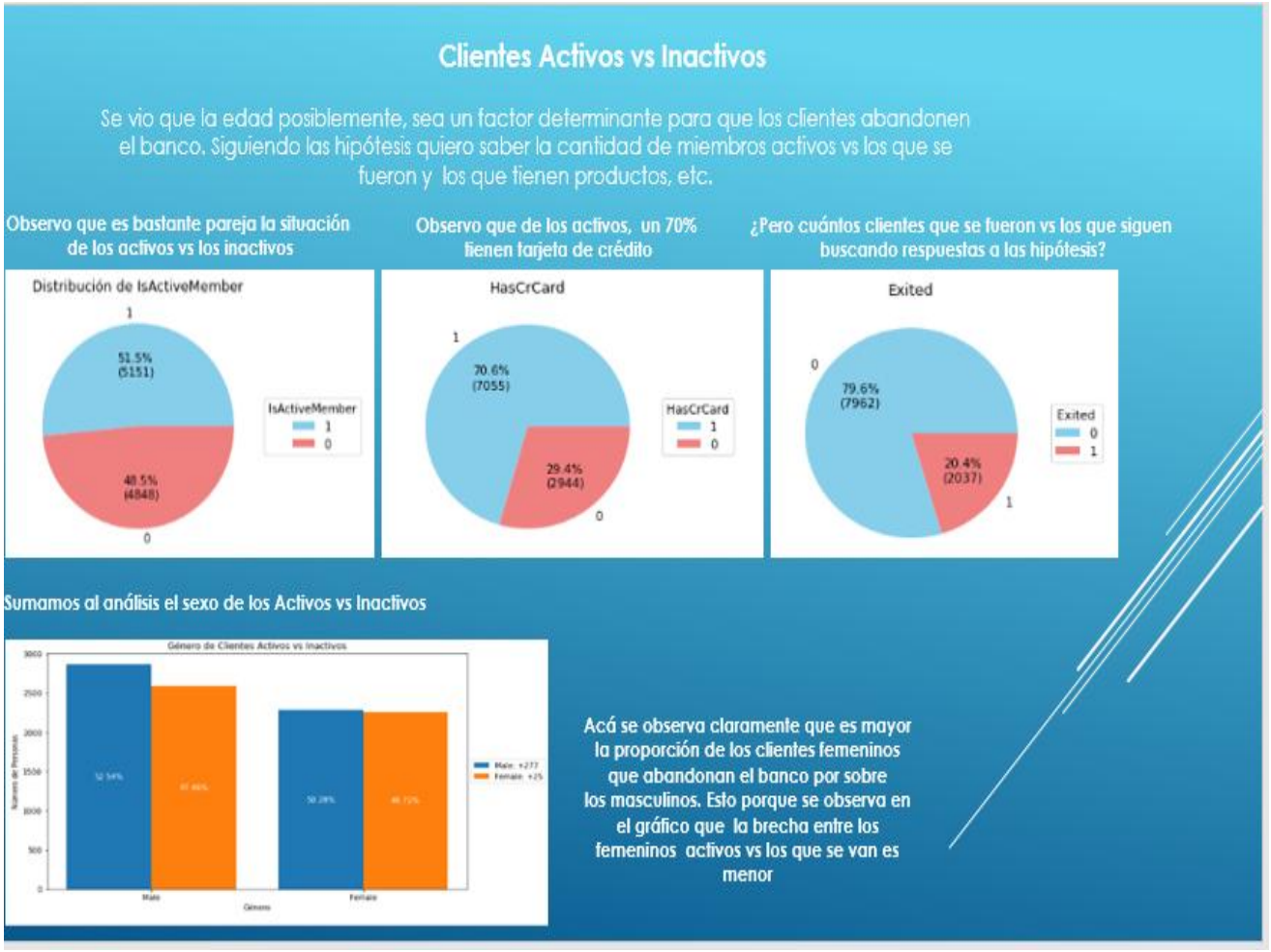
Así se ve la distribución de los salarios vs el género, pero no sé las cantidades de Hombres Vs Mujeres



Ahora veremos a cuánto equivalen las cantidades:




## *Género de los clientes activos*



### *Clientes Activos vs Inactivos*

Se vio que la edad posiblemente, no es un factor determinante para que los clientes abandonen el banco.

Por ello me interesa saber la actividad de los clientes. Por acá pienso que se puede obtener alguna respuesta, para saber quiénes abandonan o no el banco:

a- Cuento clientes que están activos:

df_Churn_Modelling_archivo['IsActiveMember'].value_counts()

```
1    7055(activos)
0    2945(inactivos)
Name: HasCrCard, dtype: int64
```

b- Cuento los clientes que tienen tarjetas de Crédito, por lo cual siguen activos:

df_Churn_Modelling_archivo['HasCrCard'].value_counts()

```
1    7055
0    2945
Name: HasCrCard, dtype: int64
```

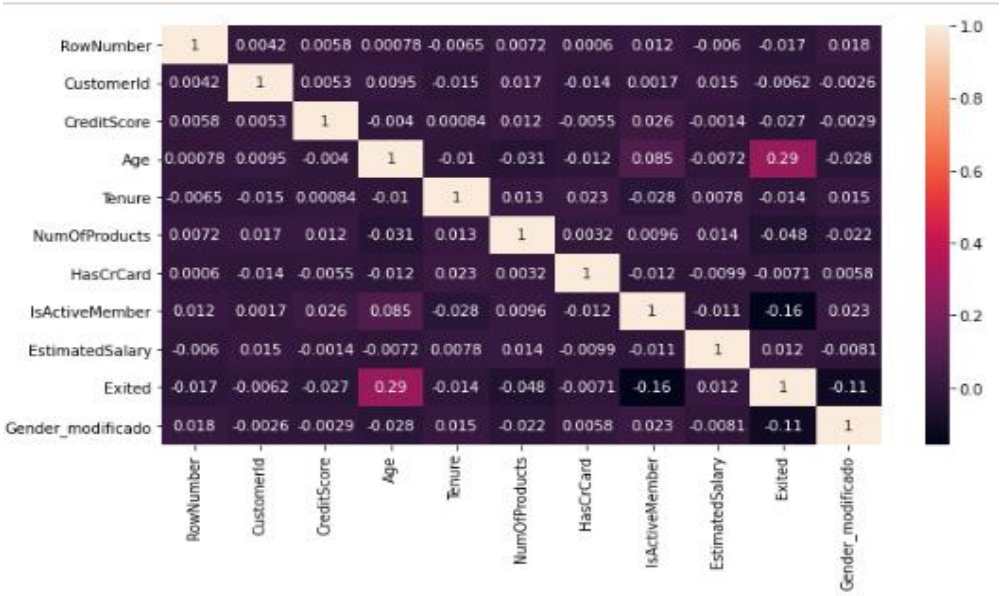
c- Cuento clientes que se fueron vs los que siguen, buscando respuestas a las hipótesis:

df_Churn_Modelling_archivo['Exited'].value_counts()

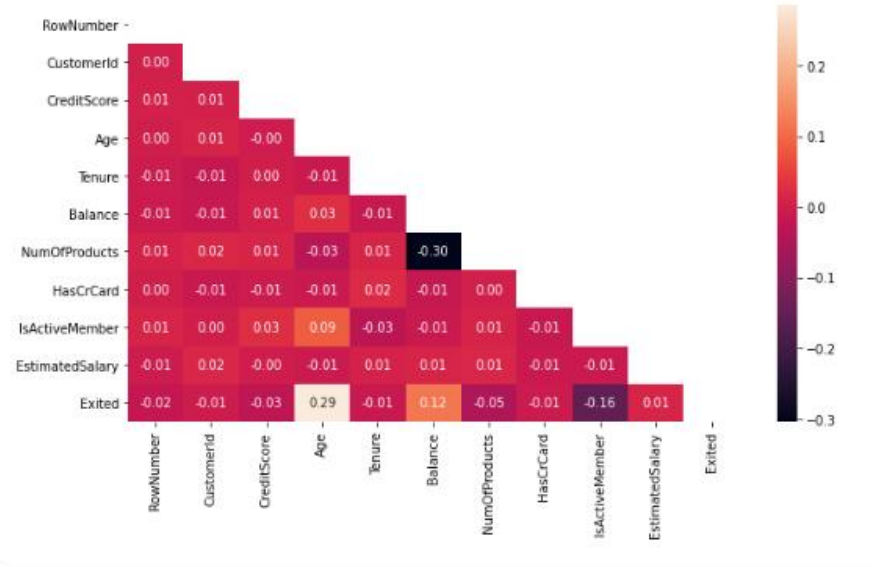
```
0    7963(activos)
1    2037(inactivos)
Name: Exited, dtype: int64
```

### *Análisis de correlaciones*

Considero que, para poder seguir un camino aceptable para conseguir el modelo, se tendría que hacer un análisis de correlaciones.



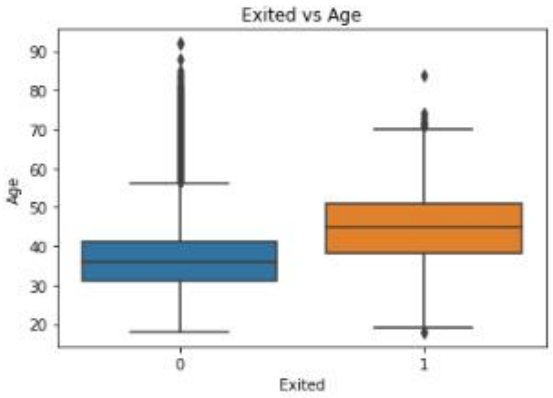

Para seguir aclarando lo anterior, se descartan las variables que no son relevantes, resultando lo siguiente:




Se observa que la relación más importante se daría entre en Exited en función de Age.

## Modelos

Previo a armar los modelos quiero veras las cantidades de Exited y la relación con la Age.
Observo los  Exited distribuidos por Age, veo que sigue siendo 50 y 50.




Ahora voy a contar los clientes que se fueron, entiendo que serían de la columna Exited = 1, y los que siguen = 0

```
In [51]: #Ahora voy a contar los clientes que se fueron, entiendo que serían de la columna Exited = 1, y los que siguen = 0

         #Cuento clientes que se fueron vs los que siguen , posiblemente por acá pueda  obtener más respuestas a las hipótesis , para sabe
         df_Churn_Modelling_archivo['Exited'].value_counts()

Out[51]: 0    7963
         1    2037
         Name: Exited, dtype: int64
```

## Modelos 1- Exited - Age

Busco predecir la probabilidad de salir (Exited) en función de la edad con un modelo con la variable dependiente Exited y como variable independiente a Age.  Surge del gráfico de correlación en donde se observó esa relación. Se presenta lo siguiente:

```
8]: #Hago un modelo para predecir el EstimatedSalary y el CreditScore. EstimatedSalary variable dependiente (Y) y CreditScore indepe
    #se puede ver que el Pv es mayor a 0,05 por lo cual vamos a descartar la variable CreditScore.
    import statsmodels.api as sm
    import statsmodels.formula.api as smf

    model = 'Exited ~ Age'

    # La función ols() se utiliza para especificar el modelo de regresión lineal.
    lm = smf.ols(formula=model, data=df_Churn_Modelling_archivo).fit()
    print(lm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Exited   R-squared:                       0.081
Model:                            OLS   Adj. R-squared:                  0.081
Method:                 Least Squares   F-statistic:                     886.1
Date:                Tue, 30 Jan 2024   Prob (F-statistic):          1.24e-186
Time:                        20:39:38   Log-Likelihood:                -4670.4
No. Observations:               10000   AIC:                             9345.
Df Residuals:                    9998   BIC:                             9359.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.2228      0.015    -15.014      0.000      -0.252      -0.194
Age            0.0116      0.000     29.767      0.000       0.010       0.012
==============================================================================
Omnibus:                     1663.435   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2569.513
Skew:                           1.262   Prob(JB):                         0.00
Kurtosis:                       3.183   Cond. No.                         155.
==============================================================================
```

Que se concluye:

1. R-cuadrado (R-squared): El R-cuadrado es 0.081, lo que significa que alrededor del 8.1% de la variabilidad en la variable dependiente (Exited) puede explicarse por la variable independiente (Age).
2. P>|t|: Ambos valores p son muy cercanos a cero (0.000), lo que sugiere que ambas variables son estadísticamente significativas.
3. F-statistic: El valor es 886.1, y la probabilidad asociada también es 1.24e-186. Este F-statistic y su probabilidad evalúan la significancia global del modelo.

La variable "Age" tiene un coeficiente positivo, lo que sugiere que hay una relación positiva entre la edad y la probabilidad de salir como se viene afirmando.

Pero necesito mejorar el R- R-squared por lo cual voy a intentar agregando al modelo otro coeficiente como "IsActiveMember".Para ello se instaló  Statsmodels.

## Modelos 11- Exited – Age + IsActiveMember

Ahora sabiendo que es probable que la edad pueda que ser un factor determinante, me interesa saber la actividad de los clientes:
#Cuento clientes activos vs inactivos, posiblemente por acá pueda obtener alguna respuesta, para saber quiénes abandonan o no el banco
df_Churn_Modelling_archivo['IsActiveMember'].value_counts()

```
Out[25]: 1    5151
         0    4849
         Name: IsActiveMember, dtype: int64
```

Entonces ahora Busco predecir la probabilidad de salir (Exited) en función de la Edad con un modelo con la variable dependiente Exited y como variable independiente a Age y agrego IsactiveMember:

```
In [61]: import statsmodels.formula.api as smf

         # Asumo que df_Churn_Modelling_archivo es tu DataFrame

         # Modelo con Exited como variable dependiente y Age y IsActiveMember como variables independientes
         model2 = 'Exited ~ Age + IsActiveMember'

         # Utilizamos la función ols() para especificar el modelo de regresión lineal
         lm2 = smf.ols(formula=model2, data=df_Churn_Modelling_archivo).fit()

         # Imprimimos el resumen del modelo
         print(lm2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Exited   R-squared:                       0.114
Model:                            OLS   Adj. R-squared:                  0.114
Method:                 Least Squares   F-statistic:                     644.6
Date:                Tue, 30 Jan 2024   Prob (F-statistic):          4.70e-264
Time:                        20:09:54   Log-Likelihood:                -4488.4
No. Observations:               10000   AIC:                             8983.
Df Residuals:                    9997   BIC:                             9005.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1705      0.015    -11.505      0.000      -0.200      -0.141
Age            0.0116      0.000     31.846      0.000       0.011       0.012
IsActiveMember    -0.1465      0.008    -19.248      0.000      -0.161      -0.132
==============================================================================
Omnibus:                     1606.552   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2523.129
Skew:                           1.226   Prob(JB):                         0.00
Kurtosis:                       3.201   Cond. No.                         159.
==============================================================================
```

1. R-cuadrado (R-squared): El R-cuadrado es 0.114. Esto significa que alrededor del 11.4% de la variabilidad en la variable dependiente "Exited" puede explicarse por las variables independientes "Age" e "IsActiveMember". Aunque no es muy alto, sugiere que el modelo explica un cierto porcentaje de la variabilidad.
2. Valores p (P>|t|): Todos los valores p asociados con los coeficientes son muy cercanos a cero (0.000), lo que sugiere que todas las variables son estadísticamente significativas.
3. F-statistic: El valor del estadístico F es 644.6, y su probabilidad asociada es muy cercana a cero (4.70e-264). Esto indica que al menos una de las variables independientes tiene un efecto significativo en la variable dependiente.

En resumen, el modelo sugiere que tanto la edad ("Age") como la membresía activa ("IsActiveMember") están asociadas con la probabilidad de salir ("Exited"). Sin embargo, el R-cuadrado indica que solo el 11.4% de la variabilidad de la variable dependiente se explica con estas dos variables en el modelo, pero subió un 3% aproximadamente con respecto al Modelo I.

## Modelos III- Exited – Age + IsActiveMember + EstimatedSalary

Quiero verificar si agregando el Salario Estimado mejora el modelo. Pero como resultado no se ve que mejore la robustez del modelo si lo comparo con el modelo 1 teniendo en cuenta el R-squared

```
In [62]: #Modelo con Exited como variable dependiente y Age y IsActiveMember como variables independientes
         model3 = 'Age ~  Exited  + IsActiveMember + EstimatedSalary '

         # Utilizamos la función ols() para especificar el modelo de regresión lineal
         lm3 = smf.ols(formula=model3, data=df_Churn_Modelling_archivo).fit()

         # Imprimimos el resumen del modelo
         print(lm3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    Age   R-squared:                       0.099
Model:                            OLS   Adj. R-squared:                  0.099
Method:                 Least Squares   F-statistic:                     365.4
Date:                Tue, 30 Jan 2024   Prob (F-statistic):          3.44e-225
Time:                        20:10:19   Log-Likelihood:                -37171.
No. Observations:               10000   AIC:                         7.435e+04
Df Residuals:                    9996   BIC:                         7.438e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     36.0293      0.235    153.390      0.000      35.569      36.490
Exited         7.9740      0.250     31.855      0.000       7.483       8.465
IsActiveMember     2.7946      0.202     13.853      0.000       2.399       3.190
EstimatedSalary -1.711e-06   1.73e-06     -0.988      0.323   -5.11e-06    1.68e-06
==============================================================================
Omnibus:                     1691.644   Durbin-Watson:                   2.032
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3425.051
Skew:                           1.021   Prob(JB):                         0.00
Kurtosis:                       5.012   Cond. No.                     3.40e+05
==============================================================================
```

## Modelos IV- 'Exited ~ Age + IsActiveMember + EstimatedSalary + Tenure +  HasCrCard'

Con este último modelo quise ver si le agregaba variables y lo hacía más complejo iba a mejorar la capacidad de predecir. Por ejemplo teniendo en cuenta que a mayor salario, mayor tenencias de productos por lo cual podía haber menos probablidades de que deje el banco, pero no tuve esos resultados.  Si se tiene en cuenta el F-statistic es más alto en el Modelo II, indicando un mejor ajuste general del modelo, y es más simple al incluir menos variables.

```
In [63]: # Modelo con Exited como variable dependiente y Age e IsActiveMember como variables independientes
         model4 = 'Exited ~ Age + IsActiveMember + EstimatedSalary + Tenure +  HasCrCard'

         # Utilizamos la función ols() para especificar el modelo de regresión lineal
         lm4 = smf.ols(formula=model4, data=df_Churn_Modelling_archivo).fit()

         # Imprimimos el resumen del modelo
         print(lm4.summary())
```

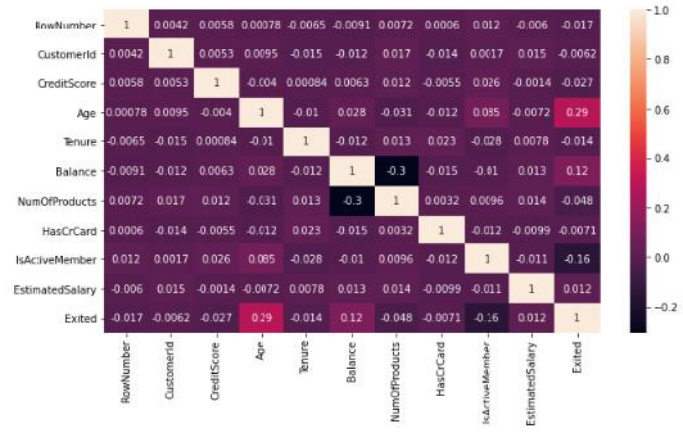
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Exited   R-squared:                       0.115
Model:                            OLS   Adj. R-squared:                  0.114
Method:                 Least Squares   F-statistic:                     258.9
Date:                Tue, 30 Jan 2024   Prob (F-statistic):          4.86e-261
Time:                        20:19:31   Log-Likelihood:                -4485.9
No. Observations:               10000   AIC:                             8984.
Df Residuals:                    9994   BIC:                             9027.
Df Model:                           5
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         -0.1642      0.019     -8.828      0.000      -0.201      -0.128
Age                0.0116      0.000     31.835      0.000       0.011       0.012
IsActiveMember    -0.1468      0.008    -19.280      0.000      -0.162      -0.132
EstimatedSalary  8.582e-08   6.59e-08      1.302      0.193   -4.34e-08    2.15e-07
Tenure            -0.0022      0.001     -1.710      0.087      -0.005       0.000
HasCrCard         -0.0047      0.008     -0.562      0.574      -0.021       0.012
==============================================================================
Omnibus:                     1605.634   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2520.999
Skew:                           1.226   Prob(JB):                         0.00
Kurtosis:                       3.201   Cond. No.                     5.75e+03
==============================================================================
```

Observaciones, anteriormente no tuve en cuenta que en el mapa de correlaciones no estaba la columna BALANCE. Detecto eso y era porque estaba como objeto. La paso a variable númerica y ahora la veo en el mapa. A partir eso iré probando mejorar el modelo.  Por ahora quedo en lo siguiente:

## Modelos V- Exited ~ Age + IsActiveMember + Balance

Agrego Balance luego de haber pasado los datos del set a numéricos

```
In [93]: #Buscando correlación entre variables

         fig, ax = plt.subplots(figsize=(10,6))

         sns.heatmap(df_Churn_Modelling_archivo.corr(),
                     annot=True,
                     ax=ax)

         plt.tight_layout()
```



1. R-squared: Indica que aproximadamente el 12.6% de la variabilidad en la variable dependiente (Exited) es explicada por estas tres variables.
P-values: Todas las variables tienen p-values significativamente bajos, lo que sugiere que todas son estadísticamente significativas.
Si lo comparo con el que hasta ahora era el mejor que fue **el Modelo II**, este tiene un R-squared más alto, lo que indica que explica una mayor proporción de la variabilidad en la variable dependiente.
Sin embargo, la inclusión de más variables no siempre es mejor. Es posible que se esté incurriendo en un sobreajuste (overfitting) al incluir variables adicionales que no mejoran significativamente la capacidad predictiva del modelo.
Pero hay que seguir analizando y teniendo en cuenta el contexto.

```
In [95]: # Modelo con Exited como variable dependiente y Age e IsActiveMember como variables independientes
         model5 = 'Exited ~ Age + IsActiveMember + Balance'

         # Utilizamos la función ols() para especificar el modelo de regresión lineal
         lm5 = smf.ols(formula=model5, data=df_Churn_Modelling_archivo).fit()

         # Imprimimos el resumen del modelo
         print(lm5.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Exited   R-squared:                       0.126
Model:                            OLS   Adj. R-squared:                  0.126
Method:                 Least Squares   F-statistic:                     480.1
Date:                Tue, 30 Jan 2024   Prob (F-statistic):          1.79e-291
Time:                        21:12:33   Log-Likelihood:                -4421.9
No. Observations:               10000   AIC:                             8852.
Df Residuals:                    9996   BIC:                             8881.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.2198      0.015    -14.340      0.000      -0.250      -0.190
Age            0.0114      0.000     31.705      0.000       0.011       0.012
IsActiveMember -0.1454     0.008    -19.229      0.000      -0.160      -0.131
Balance      6.99e-07   6.04e-08     11.575      0.000    5.81e-07    8.17e-07
==============================================================================
Omnibus:                     1555.848   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2411.404
Skew:                           1.199   Prob(JB):                         0.00
Kurtosis:                       3.187   Cond. No.                     4.04e+05
==============================================================================
```

## Modelos-Sklearn

### *PRUEBA I*

Se crea un diccionario de mapeo para la transformación de Geography string en números 1 = Spain , 2= France y 3= Germany

```
In [36]: # Se crea un diccionario de mapeo para la transformación de  Geography string en números 1 = Spain , 2= France  y 3= Germany
         geography_mapping = {'France': 2, 'Spain': 1, 'Germany': 3}

         # Se aplica la transformación a la columna 'Geography' y se crea una nueva columna 'Geography_modificado'
         df_Churn_Modelling_archivo['Geography_modificado'] = df_Churn_Modelling_archivo['Geography'].map(geography_mapping)

         # Se Imprime el data frame
         print(df_Churn_Modelling_archivo.head())
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0          1    15634602  Hargrave          619    France  Female   42
1          2    15647311      Hill          608     Spain  Female   41
2          3    15619304      Onio          502    France  Female   42
3          4    15701354      Boni          699    France  Female   39
4          5    15737888  Mitchell          850     Spain  Female   43

   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited  Geography_modificado
0        101348.88       1                     2
1        112542.58       0                     1
2        113931.57       1                     2
3         93826.63       0                     2
4         79084.10       0                     1
```

La variable objetivo (variable dependiente) va a ser se llama 'Exited'

`El Error cuadrático medio del modelo: 0.15505171269008644`

Por lo cual veo un buen ajuste del modelo.

## *PRUEBA 2 – R2*

Ahora pruebo con R2 pero consigo un valor bajo.

```
In [43]: #Prueba 2 con R²


         from sklearn.metrics import mean_squared_error, r2_score

         mse = mean_squared_error(y_test, y_pred)
         r2 = r2_score(y_test, y_pred)

         print("Mean Squared Error:", mse)
         print("R-squared:", r2)


         #Analiza las métricas de evaluación para comprender qué tan bien el modelo se ajusta a
         #los datos de prueba. Por ejemplo, un MSE más bajo y un  más alto indican un mejor rendimiento del modelo.
         #El R2 está un poco bajo
```

```
Mean Squared Error: 0.1550517126900854
R-squared: 0.010970375712446634
```

## *PRUEBA 3 - SVC*

Ahora pruebo con **SVC**

```
Accuracy en el conjunto de prueba: 0.8046666666666666
```

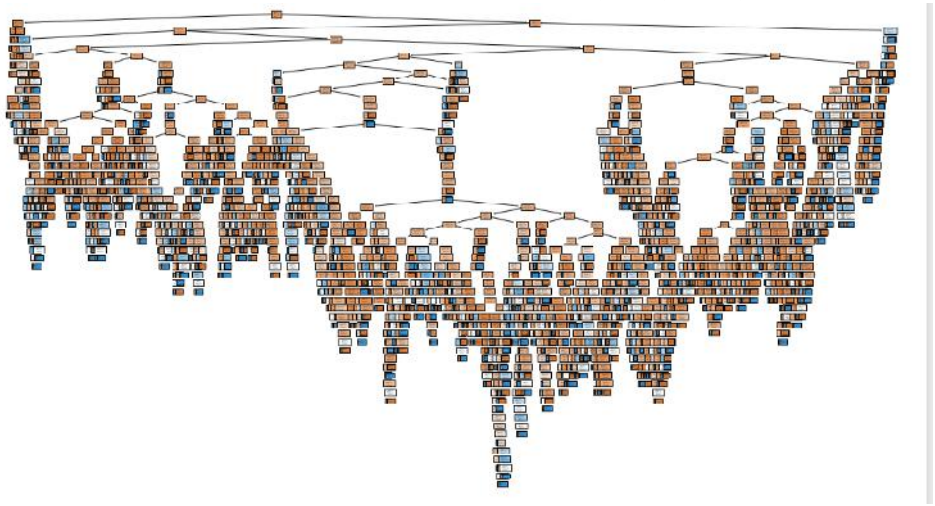
```
In [21]: #se genera el informe de  clasificación que resume el rendimiento del modelo

from sklearn.metrics import classification_report
print(classification_report(y_true=y_test, y_pred=y_pred))
```

```
              precision    recall  f1-score   support

           0       0.81      1.00      0.89      2416
           1       0.33      0.00      0.01       584

    accuracy                           0.80      3000
   macro avg       0.57      0.50      0.45      3000
weighted avg       0.71      0.80      0.72      3000
```

Accuracy en el conjunto de prueba: 0.8046666666666666

### *Prueba 4 - Random Forest*

Acá quise ver cómo era un árbol de decisiones y lo grafiqué



Precisión del árbol de decisión: 0.72 lo cual fue aceptable, pero se puede mejorar.

## Agregando Variables

1. Se crea un diccionario de mapeo para la transformación de  Gender string en números 0- 1
2. Se aplica la transformación a la columna 'Gender' y se crea una nueva columna 'Gender_modificado'
3. Se quitan las columnas, RowNumber, Surname y CustomerId

## Prueba 5 - Vuelvo agregar Balance al modelo

modelregresion = 'Age ~  Exited  + IsActiveMember + EstimatedSalary +Balance'

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    Age   R-squared:                       0.099
Model:                            OLS   Adj. R-squared:                  0.099
Method:                 Least Squares   F-statistic:                     274.2
Date:                Thu, 25 Apr 2024   Prob (F-statistic):          5.63e-224
Time:                        19:13:34   Log-Likelihood:                -37171.
No. Observations:               10000   AIC:                         7.435e+04
Df Residuals:                    9995   BIC:                         7.439e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      36.1082      0.261    138.302      0.000      35.596      36.620
Exited          7.9946      0.252     31.712      0.000       7.500       8.489
IsActiveMember  2.7958      0.202     13.859      0.000       2.400       3.191
EstimatedSalary -1.698e-06   1.73e-06     -0.980      0.327   -5.09e-06    1.7e-06
Balance        -1.113e-06   1.61e-06     -0.692      0.489   -4.26e-06    2.04e-06
==============================================================================
Omnibus:                     1690.787   Durbin-Watson:                   2.032
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3421.508
Skew:                           1.021   Prob(JB):                         0.00
Kurtosis:                       5.010   Cond. No.                     4.20e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.2e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

## Prueba 6 - Vuelvo agregar Balance quito EstimatedSalary

modelregresion2 = 'Exited ~ Age + IsActiveMember + Balance'

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Exited   R-squared:                       0.126
Model:                            OLS   Adj. R-squared:                  0.126
Method:                 Least Squares   F-statistic:                     480.1
Date:                Thu, 25 Apr 2024   Prob (F-statistic):          1.79e-291
Time:                        19:17:31   Log-Likelihood:                -4421.9
No. Observations:               10000   AIC:                             8852.
Df Residuals:                    9996   BIC:                             8881.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.2198      0.015    -14.340      0.000      -0.250      -0.190
Age            0.0114      0.000     31.705      0.000       0.011       0.012
IsActiveMember -0.1454     0.008    -19.229      0.000      -0.160      -0.131
Balance        6.99e-07   6.04e-08     11.575      0.000    5.81e-07    8.17e-07
==============================================================================
Omnibus:                     1555.848   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2411.404
Skew:                           1.199   Prob(JB):                         0.00
Kurtosis:                       3.187   Cond. No.                     4.04e+05
==============================================================================
```

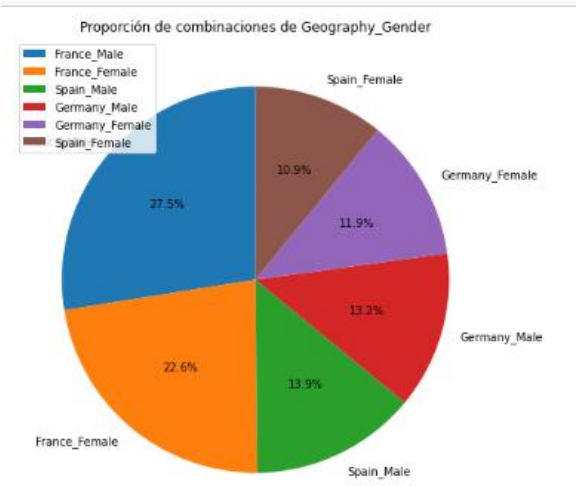
## Nueva revisión de variables

### *Comienzo a agregar las nuevas variables que pide el desafío*

1. Quería saber la cantidad de personas por  país

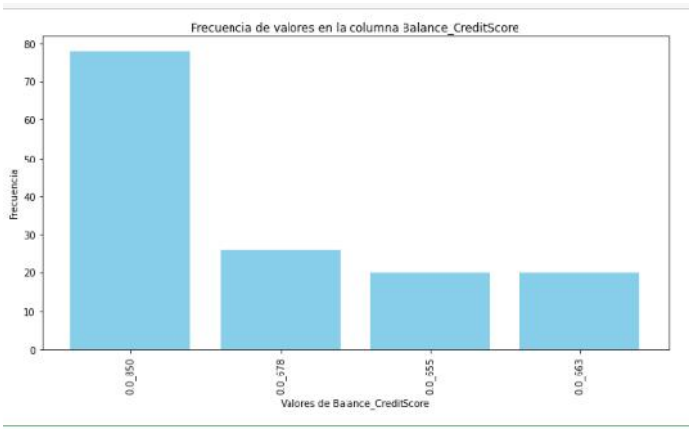
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

2. Creo una nueva variable combinando 'Geography' y 'Gender'
3. Cuento la frecuencia de cada combinación de 'Geography' y 'Gender'
4. Grafico las combinaciones para saber cuál es la combinación q más se da de personas




Concluyo que, como vimos anteriormente, la mayor cantidad de datos provenían de Francia, era lógico que las mayores combinaciones se iban a dar con personas de Francia.

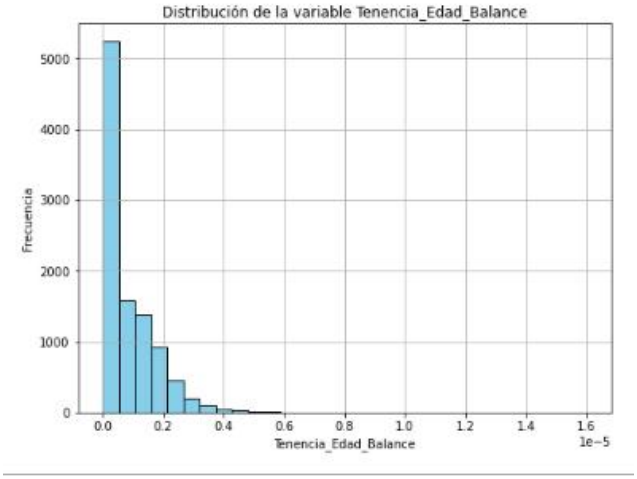
5. Sigo agregando nuevas variables/columnas para mejorar el modelo combinando Balance con Credit Score
6. Cuento la frecuencia de los valores en la nueva columna 'Balance_CreditScore' e imprimo los resultados.
7. Grafico los resultados con las frecuencias obtenidas



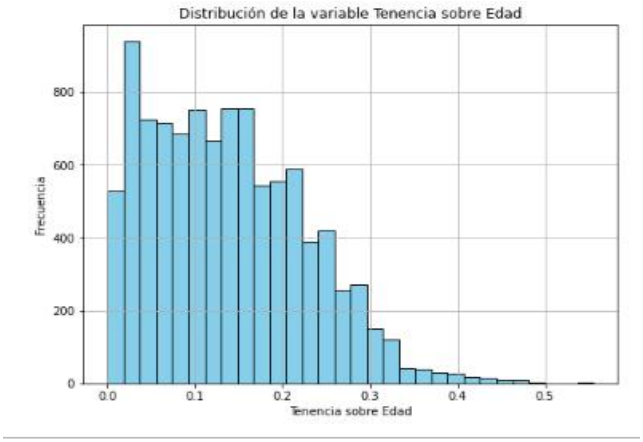
8. Cuento los valores únicos en la nueva columna 'Balance_CreditScore'

```
Número de valores únicos en Balance_CreditScore: 6817
```

9. Sigo agregando nuevas variables/columnas dividiendo la tenencia sobre el balance y grafico




10. Calculo una nueva variable 'Tenencia sobre Edad' y grafico




Se observa que tiene una forma sesgada hacia la izquierda (negativamente sesgada), esto podría indicar que hay más clientes jóvenes con una mayor relación 'Tenencia sobre Edad'

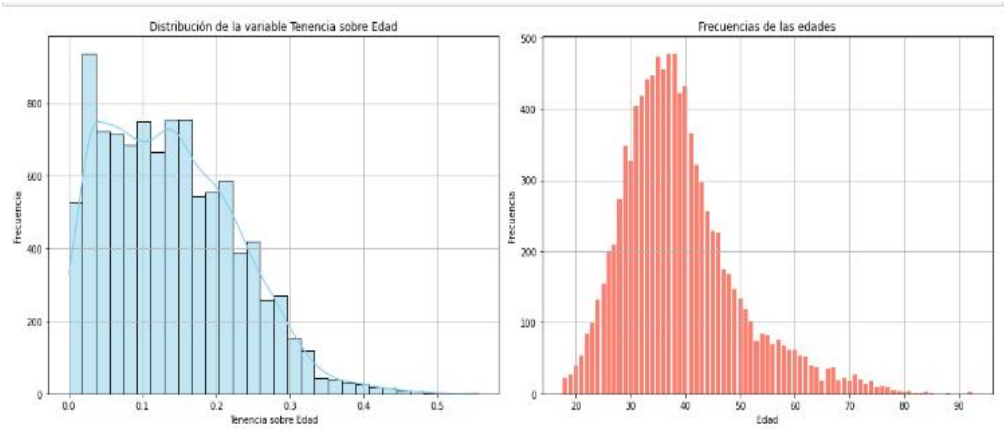
La variable 'Tenencia sobre Edad' que se calculó representa la relación entre la cantidad de tiempo que un cliente ha sido titular de una cuenta ('Tenure') y su edad.

Al graficar esta variable, se visualiza la distribución de esta relación en el conjunto de datos.
Eje x (Tenencia sobre Edad): Este eje muestra los valores de la variable 'Tenencia sobre Edad'.
Cada barra representa un rango específico de valores de esta relación.

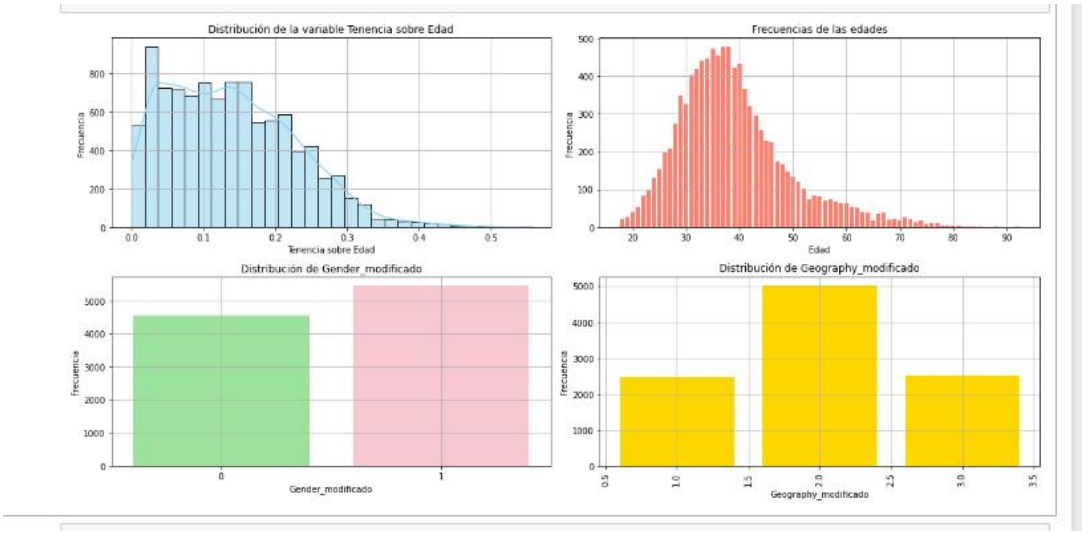
Eje y (Frecuencia): Este eje muestra la frecuencia o cantidad de observaciones que caen dentro de cada rango de valores de 'Tenencia sobre Edad'.
Cuanto más alta sea la barra en el histograma, mayor será la cantidad de observaciones que tienen una relación 'Tenencia sobre Edad' dentro de ese rango.

11. Grafico la nueva variable con KDE

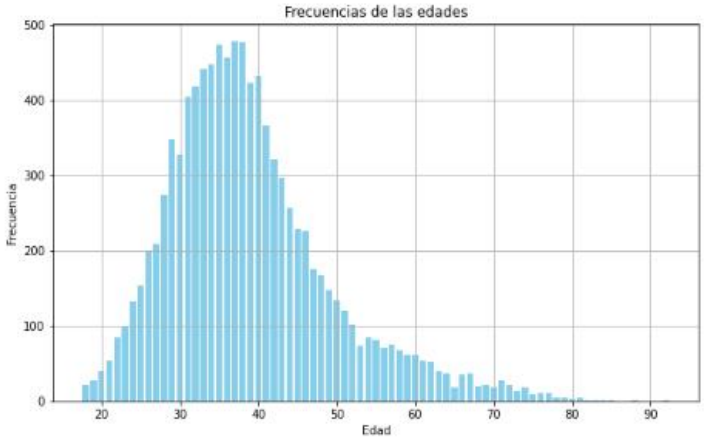
La variable Tenencia sobre Edad y las frecuencias de las edades



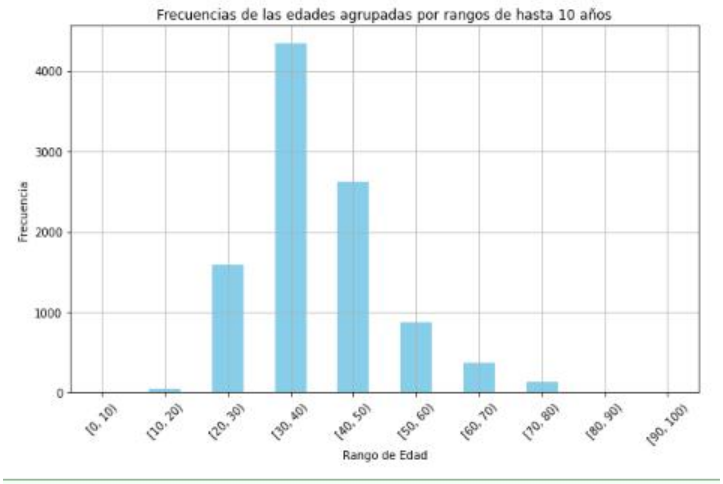

12. Grafico completo de las variables que me interesan como Tenencia, Edad, Género etc.



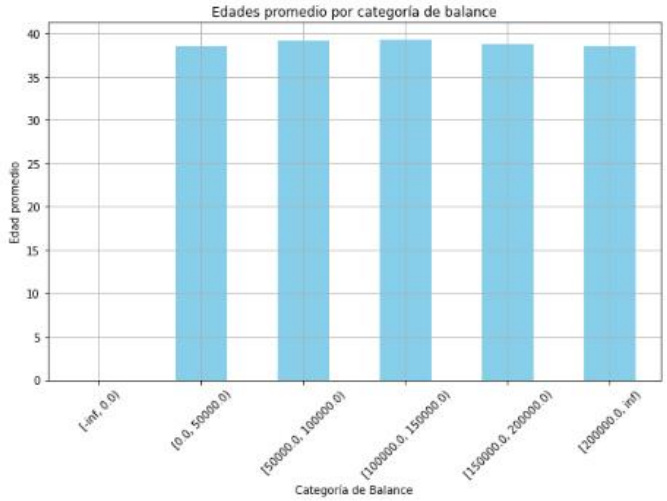

13. Cuento las edades con su frecuencia y grafico las frecuencias de las edades



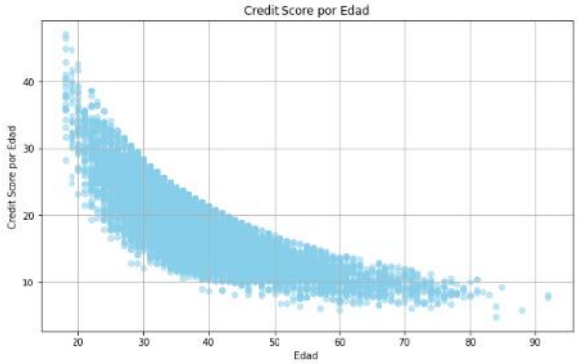

14. Contar las edades con su frecuencia, quiero verla de otra forma para asegurarme las afirmaciones defiendo rangos de edad cada 10 años de diferencia, creando categorías de edad, contándolas frecuencias y graficando




15.Se crea lo mismo para los balances y edades




16.Creo una nueva variable CreditScore por edad y grafico



17. Creo la nueva variable CreditScore por edad y agregarla al DataFrame, esto quería recorar gráficamente como era el credit score en relación a la edad

18. Elimino las columnas Gender y Geography preparando los datos para poder crear los modelos
19. Convierto la columna 'Balance_CreditScore' a tipo float para poder modelar
20. Se Inicializa el codificador de etiquetas
21. Codifico 'Geography_Gender'
22. Elimino 'Geography_Gender'

## Prueba 7 – Ramdom Forest con nuevas variables agregadas

```
El rendimiento del modelo Accuracy: 0.864
```

## Prueba 8 – LogisticRegression

```
El rendimiento del modelo Accuracy: 0.7985
```

## Prueba 9 - Pruebo con SVC

```
El rendimiento del modelo Accuracy: 0.8585
```

## Prueba 10 - Aplico PCA combinado con RandomForestClassifier para tener los mejores parámetros y entrenar

```
El rendimiento del modelo Accuracy with PCA: 0.861
```

## Prueba 11 – Combinar GRID SERACH con PCA para buscar los mejores parámetros y entrenar el modelo

```
El rendimiento del modelo Accuracy with Grid Search and PCA: 0.8675
```

## Prueba 12 - Con Random Search Cross-Validation

```
El rendimiento del modelo Accuracy : 0.8625
```

## Prueba 13 - Upgrade de scikit-learn y pruebo con HalvingRandomSearchCV

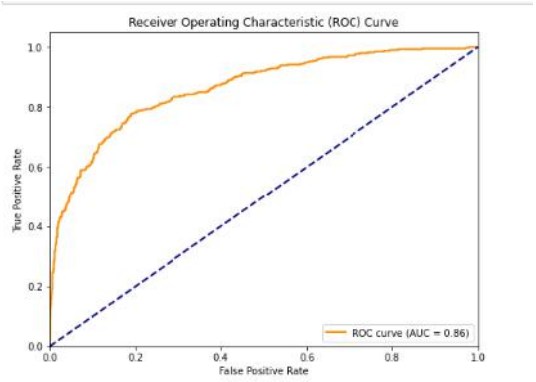
```
El rendimiento del modelo Accuracy: 0.863
```

## Prueba 14 - HyperOpt-Sklear

# CONCLUSIONES

```
El rendimiento del modelo Accuracy: 0.863
```

## Curva Roc




Se observa por medio de la curva, que el modelo tiene un buen rendimiento de clasificación

Considero que fue importante experimentar con diferentes modelos y técnicas para determinar cuál funcionó mejor para el conjunto de datos que se seleccionaron con todas las variaciones que se hicieron. Además, se realizó la validación cruzada y se ajustaron los hiperparámetros para obtener el mejor rendimiento posible del modelo.

Comenzando con Modelos de Correlación, para el mejor que obtuve, el R-squared indicó que aproximadamente el 12.6% de la variabilidad en la variable dependiente (Exited) es explicada por tres variables y los P-values decían que todas las variables tienen p-values significativamente bajos, lo que sugirió que todas son estadísticamente significativas.

Comparando modelos, el que tuvo un R-squared más alto, indicó que explicaba una mayor proporción de la variabilidad en la variable dependiente. Sin embargo, la inclusión de más variables no siempre es mejor. Es posible que se esté incurriendo en un sobreajuste (overfitting) al incluir variables adicionales que no mejoran significativamente la capacidad predictiva del modelo.

Se supo que el 20% de los clientes abandonarán al banco. Por lo cual, dado que el 20% es un número pequeño, se intentó que el modelo elegido sea lo suficiente preciso para poder inferir sobre ese porcentaje de abandono. Esto porque a cualquier banco le interesaría identificar y conservar este grupo en lugar de predecir los clientes que se retienen.

La mayoría de los datos provienen de personas de Francia, por lo cuál los resultados obtenidos se van a referir en su mayoría a clientes franceses.

Curiosamente, la mayoría de los clientes que abandonaron son aquellos con tarjetas de crédito. Dado que la mayoría de los clientes tienen tarjetas de crédito, esto podría ser sólo una coincidencia.

Luego de esas consideraciones, y por medio de la combinación de GRID SERACH con PCA, se puede sugerir con una precisión del 86% aproximadamente que sobre el 20 % de clientes que abandonan el banco se tendrá que :

- El banco tendrá que prestar especial atención a los clientes jóvenes, comenzando con las mujeres francesas. Estas porque tienen la mayor proporción de abandono, luego los hombres franceses.
- También comenzar la prevención sobre el abandono con aquellos que tienen la mayor cantidad de tenencias. Ellos son más propensos a ser tentados con nuevas inversiones que los clientes que tienen mayor edad. El banco puede necesitar implementar un programa para convertir a este grupo en clientes activos, ya que esto definitivamente tendrá un impacto positivo en la rotación de clientes. Esto porque se observó que las cantidad de clientes activos vs inactivos es muy parecida.
- Tener muy cuenta la variable 'Tenencia sobre Edad' que se calculó y que representa la relación entre la cantidad de tiempo que un cliente ha sido titular de una cuenta ('Tenure') y su edad. Al graficar esta variable, se visualizó la distribución de esta relación en el conjunto de datos.

- Por último, prestar atención a aquellos que tienen menor EstimatedSalary y CreditScore.