```
import pandas as pd
import numpy as np
df = pd.read_csv
('https://raw.githubusercontent.com/yessss28/Estadistica/refs/heads/
main/Churn_Modelling%20(1).csv')
df.dropna(inplace=True)
df
df.drop(columns = ["RowNumber", "CustomerId", "Surname"], inplace =
True)
df
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 9998,\n \"fields\":
[\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 96,\n
\"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\":
460,\n   \"samples\": [\n   716,\n   475,\n
588\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Geography\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 3,\n   \"samples\":
[\n   \"France\",\n   \"Spain\",\n   \"Germany\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Gender\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"Male\",\n   \"Female\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 10.487985680801943,\n
\"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\":
73,\n   \"samples\": [\n   29.0,\n   71.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Tenure\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 2,\n   \"min\": 0,\n   \"max\": 10,\n
\"num_unique_values\": 11,\n   \"samples\": [\n   3,\n
2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Balance\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 62393.18703475617,\n   \"min\": 0.0,\n
\"max\": 250898.09,\n   \"num_unique_values\": 6379,\n
\"samples\": [\n   156834.34,\n   161592.76\n   ],\
n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"NumOfProducts\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0,\n   \"min\": 1,\n   \"max\": 4,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   3,\n

4\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"HasCrCard\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.4558219327223667,\n   \"min\":
0.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   0.0,\n   1.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"IsActiveMember\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.4998058202347973,\n   \"min\": 0.0,\n   \"max\": 1.0,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0.0,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"EstimatedSalary\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 57510.939962097764,\n   \"min\":
11.58,\n   \"max\": 199992.48,\n   \"num_unique_values\":
9995,\n   \"samples\": [\n   51752.18,\n
121408.55\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Exited\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0,\n
1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df"}

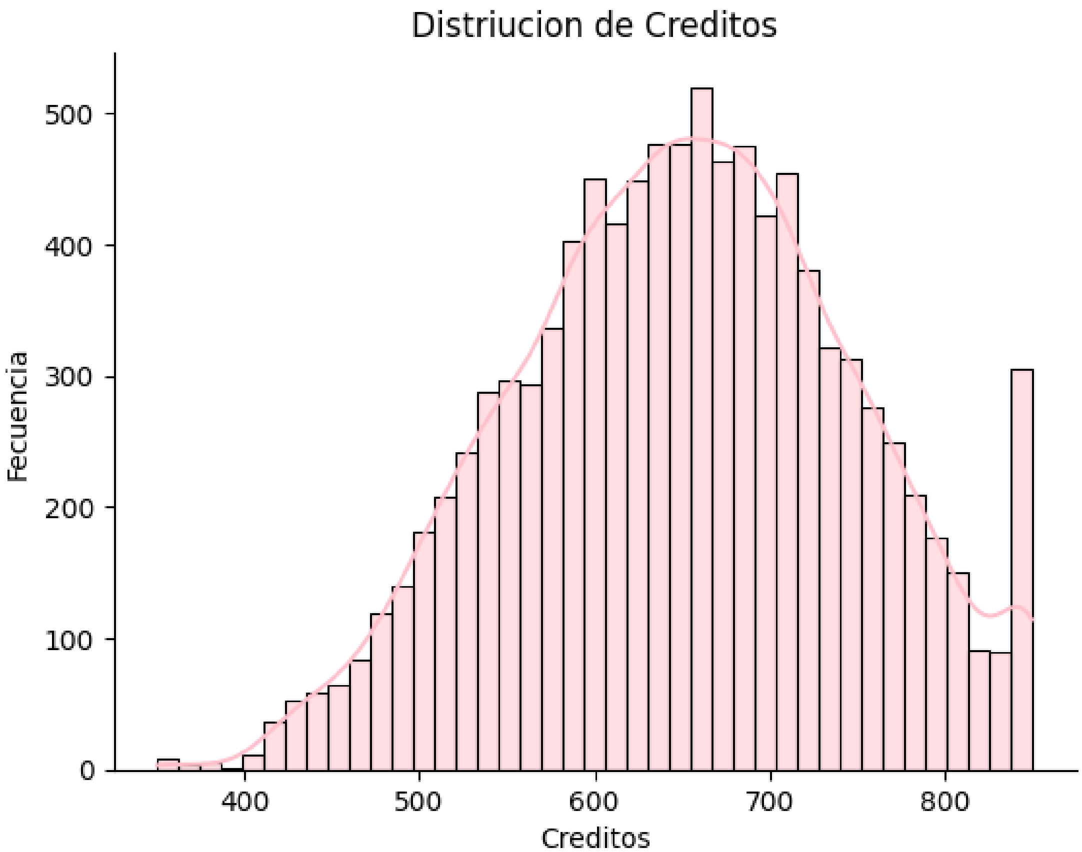
```
df["Geography"].unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(Geography)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova

#No se rechaza la hipotesis nula, no hay diferencia de grupos
```

{"summary":"{\n \"name\": \"tabla_anova\",\n \"rows\": 2,\n
\"fields\": [\n   {\n   \"column\": \"df\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
7066.118064397169,\n   \"min\": 2.0,\n   \"max\": 9995.0,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   9995.0,\n
2.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sum_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 23375249960878.414,\n   \"min\": 3782055029.3742604,\n
\"max\": 33061377573564.785,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   33061377573564.785,\n

```
3782055029.3742604\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"mean_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 1001803529.6723961,\n        \"min\":
1891027514.6871302,\n        \"max\": 3307791653.18307,\n
\"num_unique_values\": 2,\n        \"samples\": [\n
3307791653.18307,\n          1891027514.6871302\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"F\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": null,\n        \"min\":
0.5716888223196901,\n        \"max\": 0.5716888223196901,\n
\"num_unique_values\": 1,\n        \"samples\": [\n
0.5716888223196901\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PR(>F)\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": null,\n        \"min\": 0.5645896326325097,\n        \"max\":
0.5645896326325097,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          0.5645896326325097\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"tabla_anova"}

import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['CreditScore'], kde = True, color = 'pink')
plt.xlabel('Creditos')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Creditos')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

# no hay datos atipicos
```




```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(Geography)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova

#No se rechaza la hipotesis nula por que no hay diferencia de grupos
```


```
{"summary":"{\n  \"name\": \"tabla_anova\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"df\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
7066.118064397169,\n        \"min\": 2.0,\n        \"max\": 9995.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          9995.0,\n
2.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sum_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 23375249960878.414,\n        \"min\": 3782055029.3742604,\n
\"max\": 33061377573564.785,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          33061377573564.785,\n
3782055029.3742604\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"mean_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\
```

n   \"std\": 1001803529.6723961,\n   \"min\":
1891027514.6871302,\n   \"max\": 3307791653.18307,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
3307791653.18307,\n   1891027514.6871302\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"F\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": null,\n   \"min\":
0.5716888223196901,\n   \"max\": 0.5716888223196901,\n
\"num_unique_values\": 1,\n   \"samples\": [\n
0.5716888223196901\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"PR(>F)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": null,\n   \"min\": 0.5645896326325097,\n   \"max\":
0.5645896326325097,\n   \"num_unique_values\": 1,\n
\"samples\": [\n   0.5645896326325097\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"tabla_anova"}

```
df.drop(columns = ["Geography"], inplace = True)
df
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 9998,\n \"fields\":
[\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 96,\n
\"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\":
460,\n   \"samples\": [\n   716,\n   475,\n
588\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Gender\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"Male\",\n   \"Female\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 10.487985680801943,\n
\"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\":
73,\n   \"samples\": [\n   29.0,\n   71.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Tenure\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 2,\n   \"min\": 0,\n   \"max\": 10,\n
\"num_unique_values\": 11,\n   \"samples\": [\n   3,\n
2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Balance\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 62393.18703475617,\n   \"min\": 0.0,\n
\"max\": 250898.09,\n   \"num_unique_values\": 6379,\n
\"samples\": [\n   156834.34,\n   161592.76\n   ],\
n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"NumOfProducts\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":

0,\n   \"min\": 1,\n   \"max\": 4,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   3,\n
4\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"HasCrCard\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.4558219327223667,\n   \"min\":
0.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   0.0,\n   1.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"IsActiveMember\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.4998058202347973,\n   \"min\": 0.0,\n   \"max\": 1.0,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0.0,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"EstimatedSalary\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 57510.939962097764,\n   \"min\":
11.58,\n   \"max\": 199992.48,\n   \"num_unique_values\":
9995,\n   \"samples\": [\n   51752.18,\n
121408.55\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Exited\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0,\n
1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df"}

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(Gender)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova

#No rechazamos las hipotesis nula, no hay diferencia de grupos
```

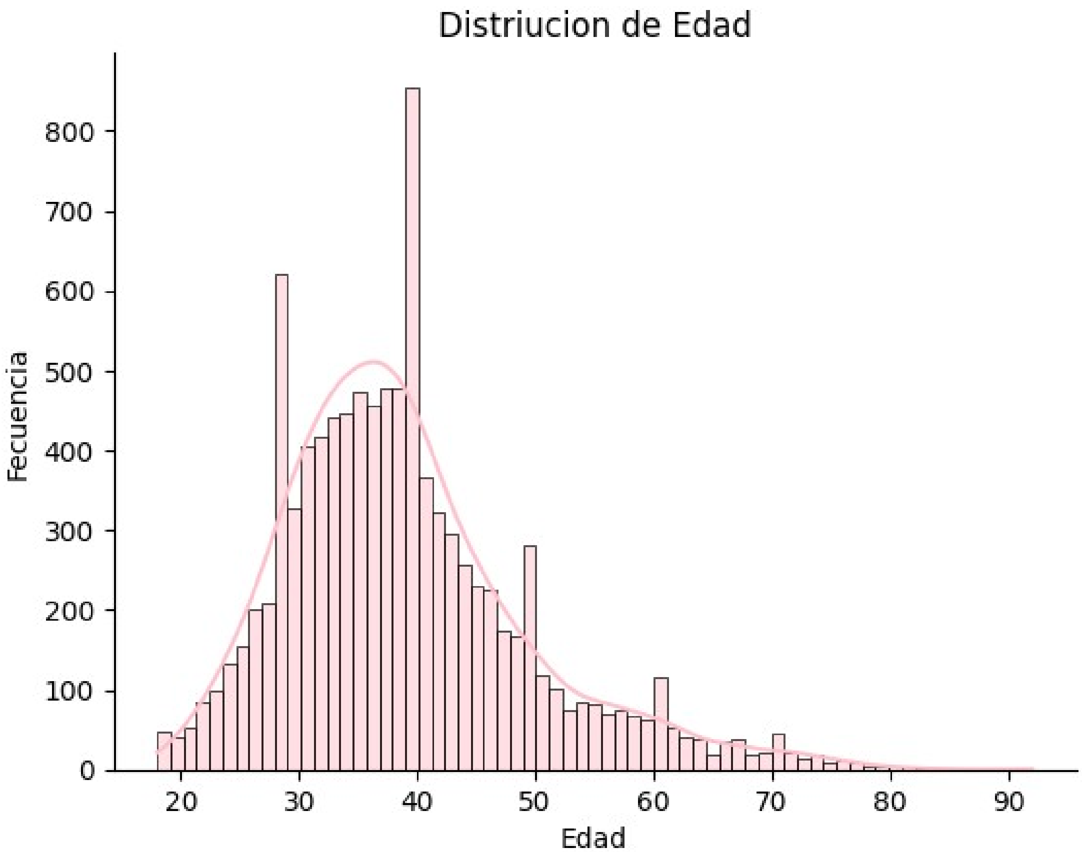
{"summary":"{\n \"name\": \"tabla_anova\",\n \"rows\": 2,\n
\"fields\": [\n   {\n   \"column\": \"df\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
7067.532277959543,\n   \"min\": 1.0,\n   \"max\": 9996.0,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   9996.0,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sum_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 23377739453147.39,\n   \"min\": 2021718164.2719057,\n
\"max\": 33063137910429.895,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   33063137910429.895,\n
2021718164.2719057\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":

```
\"mean_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 909281819.7497585,\n        \"min\":
2021718164.2719057,\n        \"max\": 3307636845.781302,\n
\"num_unique_values\": 2,\n        \"samples\": [\n
3307636845.781302,\n          2021718164.2719057\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"F\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": null,\n        \"min\":
0.6112273682192437,\n        \"max\": 0.6112273682192437,\n
\"num_unique_values\": 1,\n        \"samples\": [\n
0.6112273682192437\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PR(>F)\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": null,\n        \"min\": 0.4343445755159868,\n        \"max\":
0.4343445755159868,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          0.4343445755159868\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"tabla_anova"}
```

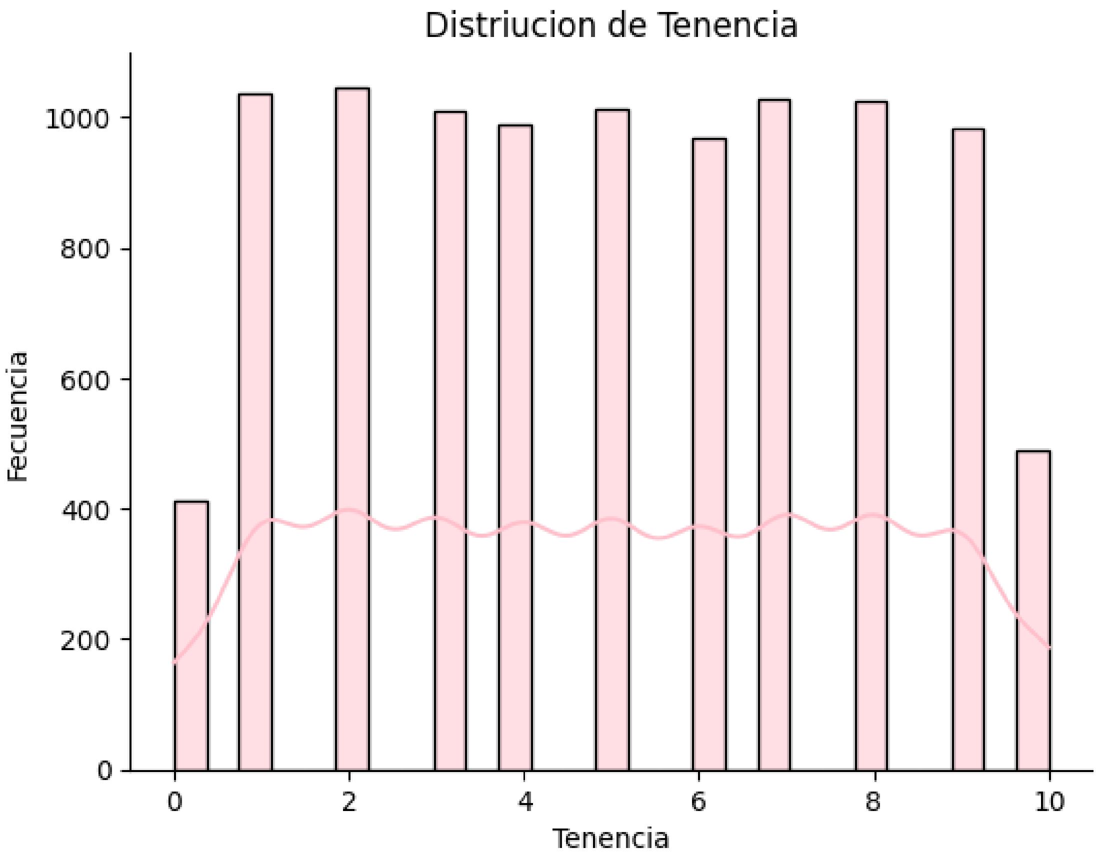

```
df.drop(columns = ["Gender"], inplace = True)
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 9998,\n  \"fields\":
[\n    {\n      \"column\": \"CreditScore\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 96,\n
\"min\": 350,\n        \"max\": 850,\n        \"num_unique_values\":
460,\n        \"samples\": [\n          716,\n          475,\n
588\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 10.487985680801943,\n        \"min\": 18.0,\n        \"max\":
92.0,\n        \"num_unique_values\": 73,\n        \"samples\": [\n
29.0,\n          71.0,\n          36.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Tenure\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 2,\n
\"min\": 0,\n        \"max\": 10,\n        \"num_unique_values\": 11,\
n        \"samples\": [\n          3,\n          2,\n          9\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Balance\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
62393.18703475617,\n        \"min\": 0.0,\n        \"max\":
250898.09,\n        \"num_unique_values\": 6379,\n        \"samples\":
[\n          156834.34,\n          161592.76,\n          166883.07\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"NumOfProducts\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          3,\n
4,\n          1\n        ],\n        \"semantic_type\": \"\",\n
```

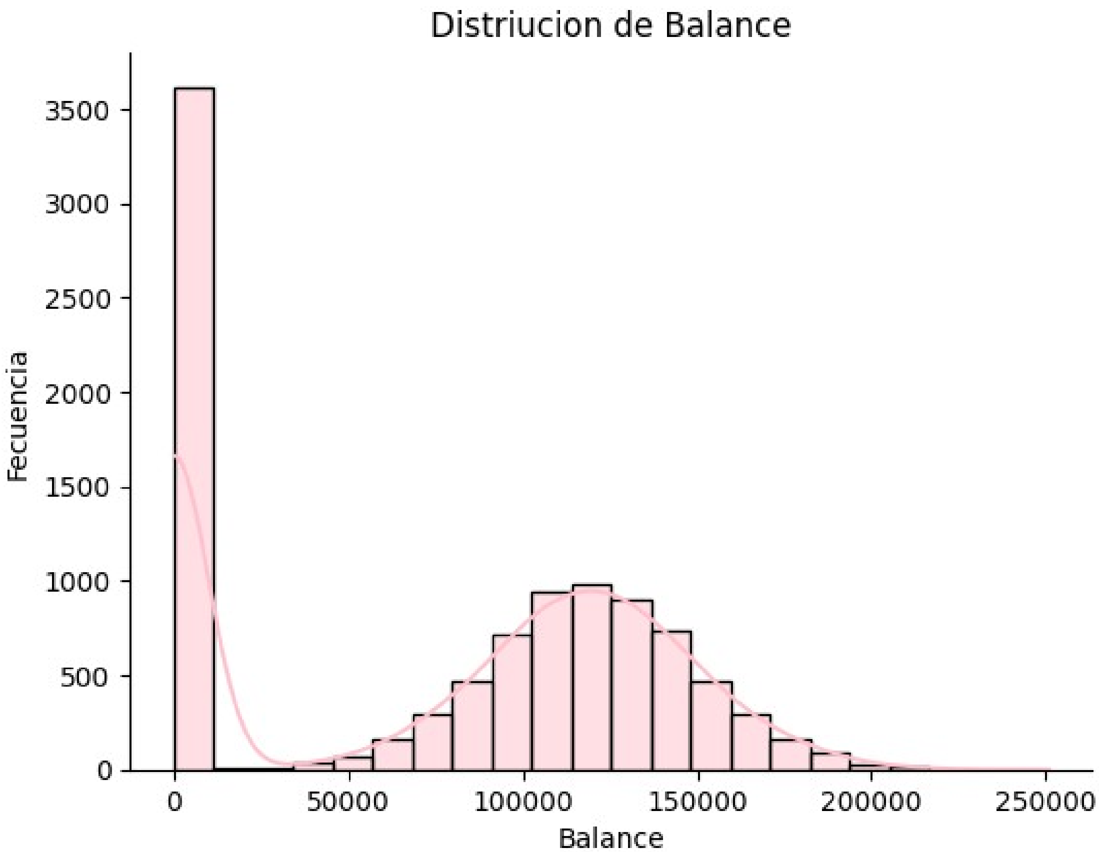
```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"HasCrCard\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.4558219327223667,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0.0,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"IsActiveMember\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.4998058202347973,\n        \"min\": 0.0,\n        \"max\": 1.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0.0,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"EstimatedSalary\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 57510.939962097764,\n        \"min\":
11.58,\n        \"max\": 199992.48,\n        \"num_unique_values\":
9995,\n        \"samples\": [\n          51752.18,\n
121408.55\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Exited\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}

import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['Age'], kde = True, color = 'pink')
plt.xlabel('Edad')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Edad')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

# no hay datos atipicos
```




```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['Tenure'], kde = True, color = 'pink')
plt.xlabel('Tenencia')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Tenencia')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

# no hay datos atipicos
```




```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['Balance'], kde = True, color = 'pink')
plt.xlabel('Balance')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Balance')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

# este si es atipico
```

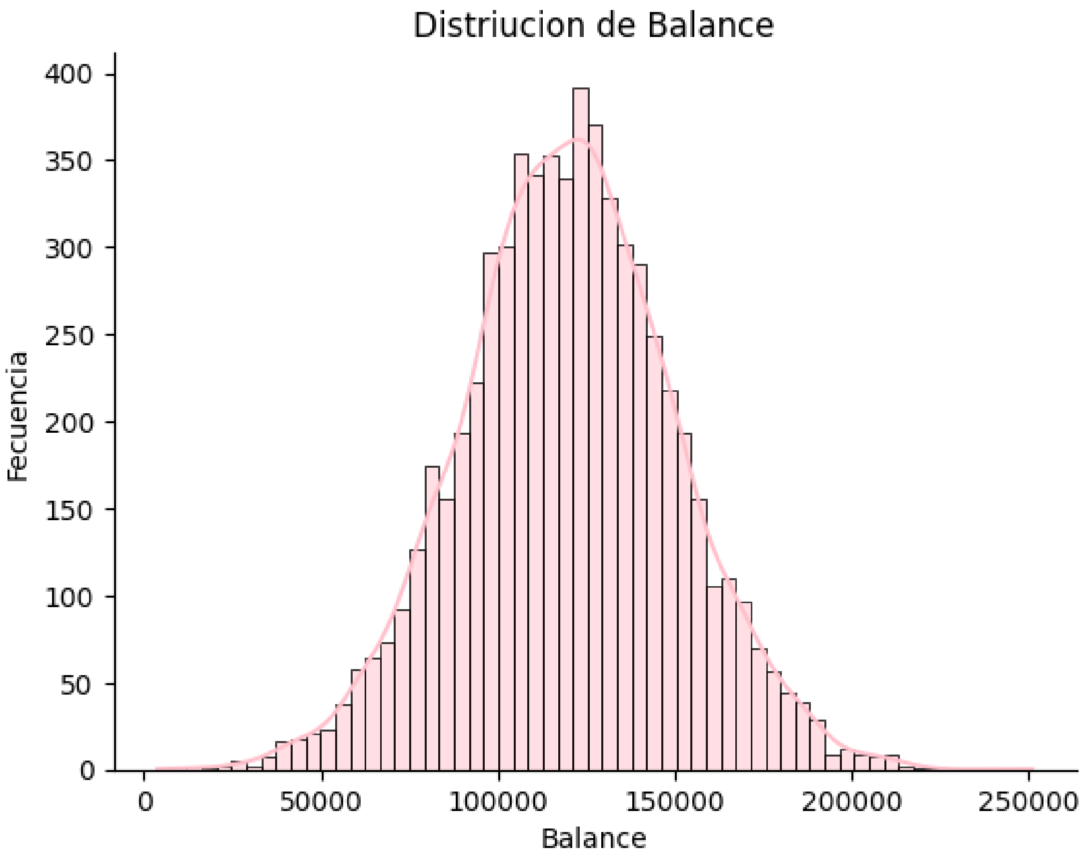



```
df = df.loc[df["Balance"]!=0.0,:]
df
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 6382,\n \"fields\": [\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 96,\n   \"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\": 450,\n   \"samples\": [\n   810,\n   467,\n   763\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10.476612570392128,\n   \"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\": 67,\n   \"samples\": [\n   47.0,\n   39.0,\n   31.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Tenure\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2,\n   \"min\": 0,\n   \"max\": 10,\n   \"num_unique_values\": 11,\n   \"samples\": [\n   2,\n   1,\n   5\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Balance\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 30100.964691131376,\n   \"min\": 3768.69,\n   \"max\":

250898.09,\n   \"num_unique_values\": 6378,\n   \"samples\": [\n   121863.61,\n   142946.18,\n   125167.74\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"NumOfProducts\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 1,\n   \"max\": 4,\n   \"num_unique_values\": 4,\n   \"samples\": [\n   3,\n   4,\n   1\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"HasCrCard\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.45859377845608573,\n   \"min\": 0.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   1.0,\n   0.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"IsActiveMember\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.4998658949563229,\n   \"min\": 0.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0.0,\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"EstimatedSalary\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 57387.54061299657,\n   \"min\": 11.58,\n   \"max\": 199970.74,\n   \"num_unique_values\": 6380,\n   \"samples\": [\n   82276.62,\n   138051.19\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Exited\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   1,\n   0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['Balance'], kde = True, color = 'pink')
plt.xlabel('Balance')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Balance')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```




```
df['NumOfProducts'].unique()
```

```
array([1, 3, 2, 4])
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(NumOfProducts)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova

#Se acepta la hipotesis nula
```


```
{"summary":"{\n  \"name\": \"tabla_anova\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"df\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 4507.80573006424,\
n        \"min\": 3.0,\n        \"max\": 6378.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          6378.0,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sum_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 14836100282901.814,\n        \"min\": 16661666675.839527,\n
\"max\": 20998075899482.9,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          20998075899482.9,\n
```

16661666675.839527\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"mean_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1599208511.8148003,\n   \"min\": 3292266525.475525,\n   \"max\": 5553888891.946509,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3292266525.475525,\n   5553888891.946509\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"F\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 1.6869499625776265,\n   \"max\": 1.6869499625776265,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   1.6869499625776265\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"PR(>F)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 0.1675103678907714,\n   \"max\": 0.1675103678907714,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   0.1675103678907714\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"tabla_anova"}

```
df["HasCrCard"].unique()
```

```
array([0., 1.])
```

```
df.drop(columns = ["HasCrCard"], inplace = True)
df
```

```
<ipython-input-18-05a1d017a22e>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop(columns = ["HasCrCard"], inplace = True)
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 6382,\n \"fields\": [\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 96,\n   \"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\": 450,\n   \"samples\": [\n   810,\n   467,\n   763\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10.476612570392128,\n   \"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\": 67,\n   \"samples\": [\n   47.0,\n   39.0,\n   31.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Tenure\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2,\n

```
\"min\": 0,\n        \"max\": 10,\n        \"num_unique_values\": 11,\
n        \"samples\": [\n          2,\n          1,\n          5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Balance\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
30100.964691131376,\n        \"min\": 3768.69,\n        \"max\":
250898.09,\n        \"num_unique_values\": 6378,\n        \"samples\":
[\n          121863.61,\n          142946.18,\n          125167.74\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"NumOfProducts\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          3,\n
4,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"IsActiveMember\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.49986589495632229,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0.0,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"EstimatedSalary\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
57387.54061299657,\n        \"min\": 11.58,\n        \"max\":
199970.74,\n        \"num_unique_values\": 6380,\n        \"samples\":
[\n          82276.62,\n          138051.19\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Exited\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
df["IsActiveMember"].unique()
```

```
array([1., 0.])
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(IsActiveMember)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova
```


```
{"summary":"{\n  \"name\": \"tabla_anova\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"df\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
7067.532277959543,\n        \"min\": 1.0,\n        \"max\": 9996.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          9996.0,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"sum_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 23374918049243.47,\n   \"min\": 4016751997.1980176,\n   \"max\": 33061142876596.96,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   33061142876596.96,\n   4016751997.1980176\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"mean_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 501561258.8547397,\n   \"min\": 3307437262.564722,\n   \"max\": 4016751997.1980176,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3307437262.564722,\n   4016751997.1980176\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"F\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 1.2144605258765404,\n   \"max\": 1.2144605258765404,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   1.2144605258765404\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"PR(>F)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 0.2704770549220117,\n   \"max\": 0.2704770549220117,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   0.2704770549220117\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"tabla_anova"}

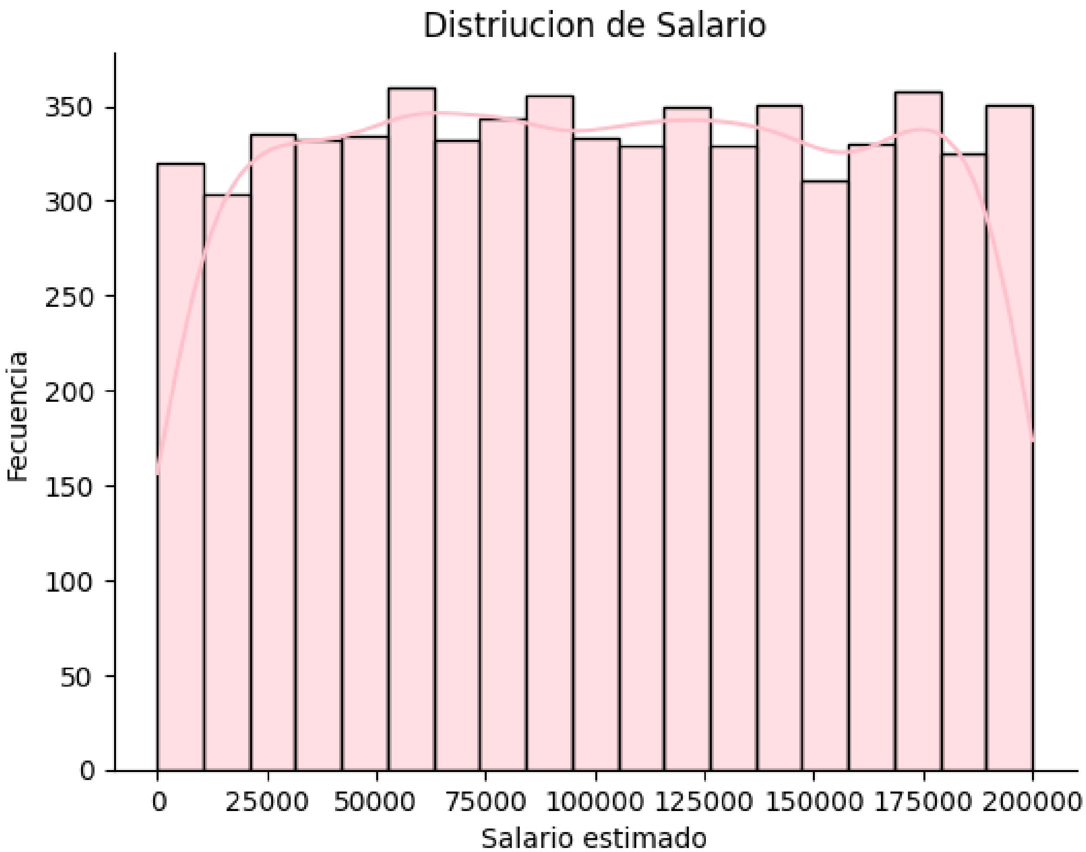
```
df.drop(columns = ["IsActiveMember"], inplace = True)
df
```

```
<ipython-input-20-13c8f583d392>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop(columns = ["IsActiveMember"], inplace = True)
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 6382,\n \"fields\": [\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 96,\n   \"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\": 450,\n   \"samples\": [\n   810,\n   467,\n   763\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10.476612570392128,\n   \"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\": 67,\n   \"samples\": [\n   47.0,\n   39.0,\n   31.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Tenure\",\n   \"properties\":

```
{\n        \"dtype\": \"number\",\n        \"std\": 2,\n
\"min\": 0,\n        \"max\": 10,\n        \"num_unique_values\": 11,\
n        \"samples\": [\n          2,\n          1,\n          5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Balance\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
30100.964691131376,\n        \"min\": 3768.69,\n        \"max\":
250898.09,\n        \"num_unique_values\": 6378,\n        \"samples\":
[\n          121863.61,\n          142946.18,\n          125167.74\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"NumOfProducts\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          3,\n
4,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"EstimatedSalary\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 57387.54061299657,\n        \"min\":
11.58,\n        \"max\": 199970.74,\n        \"num_unique_values\":
6380,\n        \"samples\": [\n          82276.62,\n
138051.19,\n          108008.65\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Exited\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```python
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['EstimatedSalary'], kde = True, color = 'pink')
plt.xlabel('Salario estimado')
plt.ylabel('Fecuencia')
plt.title('Distriucion de Salario')

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```




```
df["Exited"].unique()
```

```
array([0, 1])
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

modelo=smf.ols("EstimatedSalary ~ C(Exited)",data=df).fit()
tabla_anova = sm.stats.anova_lm(modelo)
tabla_anova

#Los datos no aportan nada
```

{"summary":"{\n \"name\": \"tabla_anova\",\n \"rows\": 2,\n \"fields\": [\n   {\n   \"column\": \"df\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 4510.634157188987,\n   \"min\": 1.0,\n   \"max\": 6380.0,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   6380.0,\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"sum_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 14858493566058.293,\n   \"min\": 827224302.8695034,\n   \"max\": 21013910341855.88,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   21013910341855.88,\n

827224302.8695034\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"mean_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1744073255.4668121,\n   \"min\": 827224302.8695034,\n   \"max\": 3293716354.522865,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3293716354.522865,\n   827224302.8695034\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"F\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 0.2511522589774847,\n   \"max\": 0.2511522589774847,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   0.2511522589774847\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"PR(>F)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": null,\n   \"min\": 0.6162821924514956,\n   \"max\": 0.6162821924514956,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   0.6162821924514956\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"tabla_anova"}

```
df.drop(columns = ["Exited"], inplace = True)
df
```

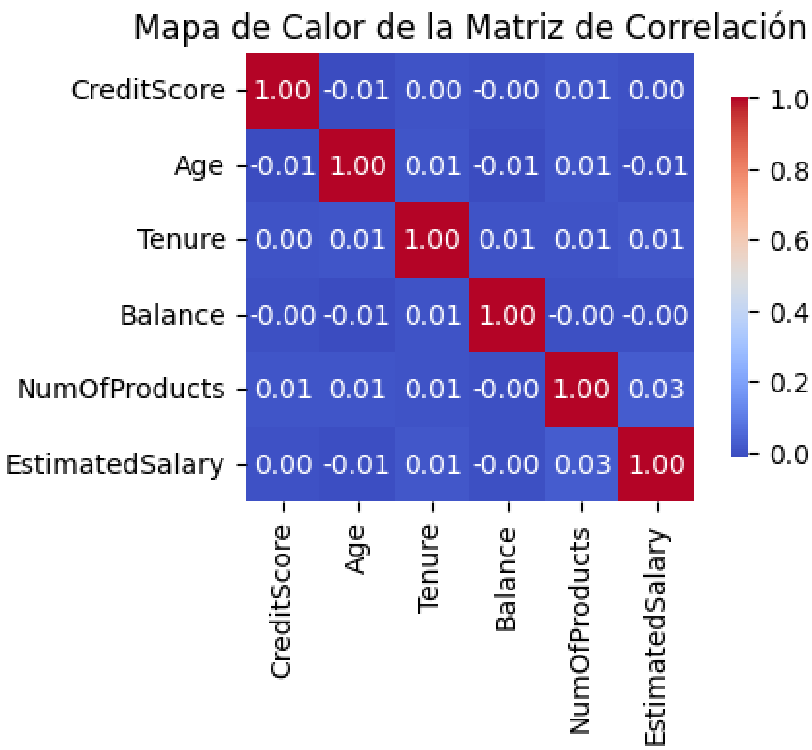
```
<ipython-input-24-bac1af30eeba>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop(columns = ["Exited"], inplace = True)
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 6382,\n \"fields\": [\n   {\n   \"column\": \"CreditScore\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 96,\n   \"min\": 350,\n   \"max\": 850,\n   \"num_unique_values\": 450,\n   \"samples\": [\n   810,\n   467,\n   763\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10.476612570392128,\n   \"min\": 18.0,\n   \"max\": 92.0,\n   \"num_unique_values\": 67,\n   \"samples\": [\n   47.0,\n   39.0,\n   31.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Tenure\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2,\n   \"min\": 0,\n   \"max\": 10,\n   \"num_unique_values\": 11,\n   \"samples\": [\n   2,\n   1,\n   5\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Balance\",\n

```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
30100.964691131376,\n        \"min\": 3768.69,\n        \"max\":
250898.09,\n        \"num_unique_values\": 6378,\n        \"samples\":
[\n          121863.61,\n          142946.18,\n          125167.74\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"NumOfProducts\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          3,\n
4,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"EstimatedSalary\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 57387.54061299657,\n        \"min\":
11.58,\n        \"max\": 199970.74,\n        \"num_unique_values\":
6380,\n        \"samples\": [\n          82276.62,\n
138051.19,\n          108008.65\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

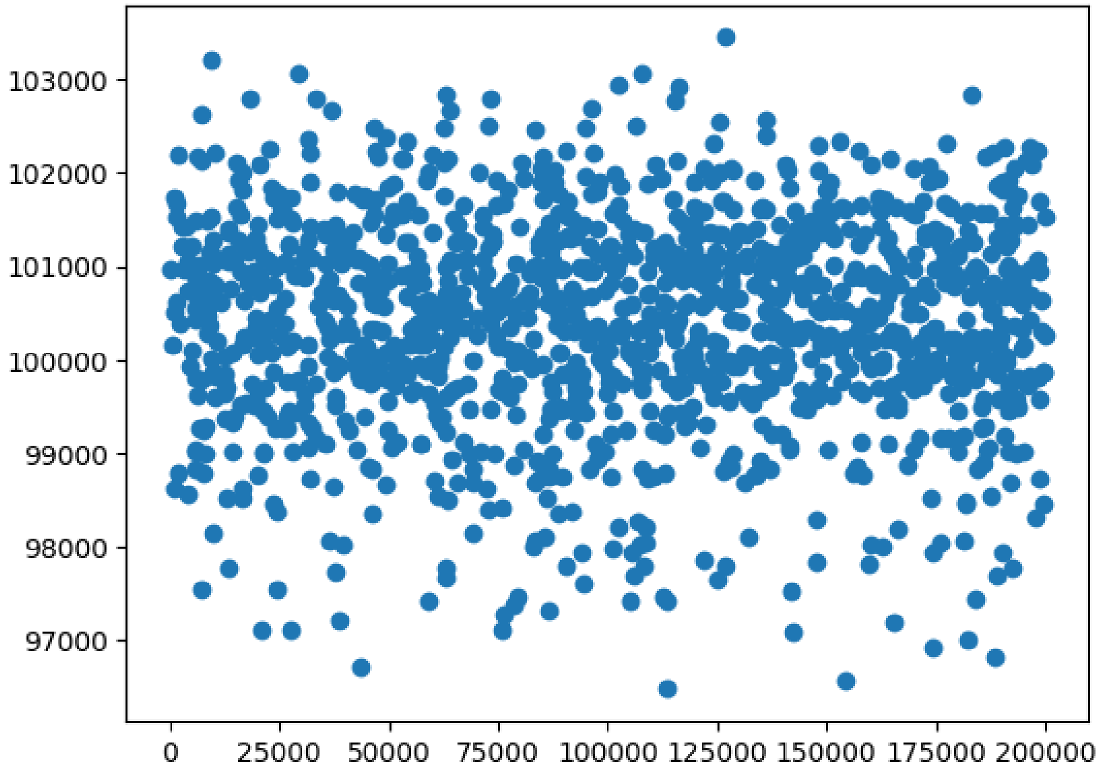

```python
import seaborn as sns
import matplotlib.pyplot as plt

matriz_de_correlacion = df.corr()
plt.figure(figsize=(5, 3))
sns.heatmap(matriz_de_correlacion, annot=True, cmap='coolwarm',
fmt=".2f", square=True, cbar_kws={"shrink": .8})
plt.title('Mapa de Calor de la Matriz de Correlación')
```

```
Text(0.5, 1.0, 'Mapa de Calor de la Matriz de Correlación')
```




```
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
import statsmodels.api as sm
import matplotlib.pyplot as plt


X = df[['CreditScore',     'Age',     'Tenure',  'Balance']]
Y = df["EstimatedSalary"]

# Datos de entrenamiento y datos de prueba

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 42)

X_train_constante = sm.add_constant(X_train)
X_test_constante = sm.add_constant(X_test)

modelo = sm.OLS(Y_train, X_train_constante).fit()


Yc = modelo.predict(X_test_constante)

plt.scatter(Y_test, Yc)

r2 = r2_score(Y_test, Yc)
print(f'Coeficiente de correlacion: {r2: 0.4f}\n')
```

```
Coeficiente de correlacion: -0.0012
```



Un coeficiente de correlacion de -0.0012 indica que no hay una relacion lineal aparentemente entre las variables, es decir, no hay correlacion.

```
b0, b1, b2, b3, b4 = modelo.params

Fun = lambda x1, x2, x3, x4: b0 + b1 * x1 + b2 * x2 + b3 * x3 + b4 *
x4

# El salario estimado se puede calcular usando variables como el
puntaje de crédito, edad, antigüedad laboral y saldo financiero , ya
que reflejan estabilidad, experiencia y capacidad de pago.
# Estos datos permiten predecir ingresos mediante modelos estadísticos
o de aprendizaje automático.

Fun(432, 60, 2, 120000)

97992.76900028627

from statsmodels.formula.api import ols

moedo_2 = ols(formula = 'EstimatedSalary ~ CreditScore + Age + Tenure
+ Balance', data = df).fit()
```

```python
tabla_anova = sm.stats.anova_lm(moedo_2, typ = 2)
tabla_anova
```

{"summary":"{\n \"name\": \"tabla_anova\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": \"sum_sq\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 9395322884568.426,\n   \"min\": 882655.0107485803,\n   \"max\": 21009822850002.05,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   1548041963.779398,\n   21009822850002.05,\n   3339432886.047395\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"df\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2851.4338849077317,\n   \"min\": 1.0,\n   \"max\": 6377.0,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   6377.0,\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"F\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.47607102336217594,\n   \"min\": 0.00026790758987970 94,\n   \"max\": 1.0136003366788102,\n   \"num_unique_values\": 4,\n   \"samples\": [\n   0.46986895955766034,\n   0.024694446999010765\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"PR(>F)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.3164837826968018,\n   \"min\": 0.3140800511817597,\n   \"max\": 0.9869414141075771,\n   \"num_unique_values\": 4,\n   \"samples\": [\n   0.49307237595804465,\n   0.8751358090292845\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"tabla_anova"}

CONCLUSIÓN El análisis de regresión lineal múltiple revela que las variables utilizadas en el modelo tienen poca capacidad para explicar la variabilidad del Salario Estimado (EstimatedSalary).

La suma de cuadrados residuales es considerablemente mayor que la de las variables predictoras, lo que indica que la mayor parte de la variabilidad en el salario no se debe a las variables incluidas en el modelo.

Cada variable tiene un grado de libertad, ya que se analizan individualmente, mientras que los residuos cuentan con 6377 grados de libertad.

El estadístico F es muy bajo (por ejemplo, 0.000268 para CreditScore y 0.469869 para Age), lo que sugiere que las variables no explican significativamente la variabilidad en el salario.

Los valores p son altos (todos superiores a 0.05, como 0.986941 para CreditScore y 0.493072 para Age), lo que significa que no hay suficiente evidencia estadística para afirmar que estas variables influyen en el salario estimado.

En resumen, el modelo indica que ni la edad ni el puntaje crediticio tienen un impacto significativo en el salario estimado. Esto sugiere que es necesario explorar otras variables o enfoques alternativos para mejorar la predicción del salario.