1. Importación de paquetes necesarios

```
# Paquetes básico para lectura y análisis de datos
import pandas as pd
import numpy as np

# Pauetes para graficar
import matplotlib.pyplot as plt
import seaborn as sns

# Paquetes para train, test
from sklearn.model_selection import train_test_split

# Paquetes de métricas de modelos
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from scipy import interp

# Paquetes de modelos
import lightgbm as lgb
import xgboost as xgb
!pip install catboost
from catboost import CatBoostClassifier, Pool
from sklearn.tree import DecisionTreeClassifier
```

```
Collecting catboost
  Downloading catboost-1.2-cp310-cp310-manylinux2014_x86_64.whl (98.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.6/98.6 MB 8.7 MB/s eta 0:00:00
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.22.4)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.5.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.10.1)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.13.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2022.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.40.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.1.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost) (8.2.2)
Installing collected packages: catboost
Successfully installed catboost-1.2
```

2. Conexión y lectura de datos a DRIVE

```
#Acceder a archivos en Google Drive

from google.colab import drive
drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

```
# Ignorar warnings
import warnings
warnings.filterwarnings("ignore")

pd.options.display.max_columns = 100 # para ver todos las columnas en este caso hasta 100 columnas
pd.options.display.max_rows = None # para ver todas las filas en este caso.
```

```
# Lectura de datos CSV en la carpeta donde se guarda la data
```

```
df = pd.read_csv('/content/drive/My Drive/MaestriaDS_FIECS_UNI/Trabajo1/DATA_VILLA_ASTON.csv')
df.head()
```

	ITEM	FECHA DE CONTRATO	CÓDIGO ASESOR VENTAS	PROYECTO	DNI CLIENTE	EDAD CLIENTE	SEXO CLIENTE	ZONA DE RESIDENCIA CLIENTE
0	1.0	8/09/2022	NaN	LA PLANICIE DE SANTA CATALINA	42487000	39.0	M	BARRIO LA ACHIRANA S/N
1	2.0	8/09/2022	NaN	LA PLANICIE DE SANTA CATALINA	77381090	23.0	M	CASERIO ALTO PERÚ
2	3.0	23/08/2022	NaN	LA PLANICIE DE SANTA CATALINA	47284299	31.0	M	CASERIO MARRIPON
3	4.0	23/08/2022	NaN	LA PLANICIE DE SANTA CATALINA	46238646	33.0	F	CALLE SALITRAL PROLONGACION EL CARMEN
4	5.0	23/08/2022	NaN	LA PLANICIE DE SANTA CATALINA	17627335	51.0	F	CALLE NICOLAS CARNERO 152

```
df.columns
```

```
Index(['ITEM', 'FECHA DE CONTRATO', 'CÓDIGO ASESOR VENTAS', 'PROYECTO',
       'DNI CLIENTE', 'EDAD CLIENTE', 'SEXO CLIENTE',
       'ZONA DE RESIDENCIA CLIENTE', 'DEPARTAMENTO', 'PROVINCIA', 'DISTRITO',
       'CLIENTE', 'EST.CIVIL', 'ZONA PROYECTO', 'VALOR TERRENO', 'M2 TERRENO',
       'CUOTA QUE DESISTIÓ', 'CUOTA_INICIAL', 'DESISTIMIENTO'],
      dtype='object')
```

```
# Conteo de "1" y porcentaje

print("Observaciones CON datos de sobre endeudamiento")
print(df['DESISTIMIENTO'].value_counts())
print("")

print("Porcentaje de Observaciones CON datos de sobre endeudamiento")
print(100 * df['DESISTIMIENTO'].value_counts(normalize = True))
```

```
Observaciones CON datos de sobre endeudamiento
0    939
1     46
Name: DESISTIMIENTO, dtype: int64

Porcentaje de Observaciones CON datos de sobre endeudamiento
0    95.329949
1     4.670051
Name: DESISTIMIENTO, dtype: float64
```

3. Limpieza y eliminación de columnas innecesarias

```
# Eliminar las columnas que no aportan valor al modelo

df = df.drop('ITEM', axis = 1)
df = df.drop('FECHA DE CONTRATO', axis = 1)
df = df.drop('CÓDIGO ASESOR VENTAS', axis = 1)
df = df.drop('DNI CLIENTE', axis = 1)
df = df.drop('DISTRITO', axis = 1)
df = df.drop('ZONA DE RESIDENCIA CLIENTE', axis = 1)
df = df.drop('CLIENTE', axis = 1)
df = df.drop('CUOTA QUE DESISTIÓ', axis = 1)
df = df.drop('PROYECTO', axis = 1)
```

```
# Contar los valores nulos por columna
conteo_nulos = df.isnull().sum()

# Calcular el porcentaje de nulos por columna
porcentaje_nulos = (conteo_nulos / len(df)) * 100

# Crear un nuevo dataframe con los resultados
df_nulos = pd.DataFrame({'Nulos': conteo_nulos, 'Porcentaje de Nulos': porcentaje_nulos})

# Ordenar el dataframe por la cantidad de nulos de manera descendente
df_nulos = df_nulos.sort_values('Nulos', ascending=False)

# Imprimir el dataframe de nulos
print(df_nulos)
```

```
               Nulos  Porcentaje de Nulos
CUOTA_INICIAL    962            97.664975
DEPARTAMENTO      46             4.670051
PROVINCIA         46             4.670051
EDAD CLIENTE       3             0.304569
VALOR TERRENO      1             0.101523
SEXO CLIENTE       0             0.000000
EST.CIVIL          0             0.000000
ZONA PROYECTO      0             0.000000
M2 TERRENO         0             0.000000
DESISTIMIENTO      0             0.000000
```

```
# Eliminar las variables con valores vacios en el desestimiento
df = df.drop('DEPARTAMENTO', axis = 1)
df = df.drop('PROVINCIA', axis = 1)


# Rellenar valores en blanco de EDAD con la media del resto de esades con el método fillna()
df['VALOR TERRENO'].fillna(df['VALOR TERRENO'].mean(), inplace = True)
df['EDAD CLIENTE'].fillna(df['EDAD CLIENTE'].mean(), inplace= True)
df['CUOTA_INICIAL'].fillna(df['VALOR TERRENO']*0.1, inplace = True)
df['VALOR_M2'] = df['VALOR TERRENO']/df['M2 TERRENO']
df['PCT_CINICIAL'] = df['CUOTA_INICIAL']/df['VALOR TERRENO']

df.head()
```

	EDAD CLIENTE	SEXO CLIENTE	EST.CIVIL	ZONA PROYECTO	VALOR TERRENO	M2 TERRENO	CUOTA_INICIAL	DESIS
0	39.0	M	SOLTERO(A)	MOTUPE	16012.0	90.0	1601.2	
1	23.0	M	SOLTERO(A)	MOTUPE	16000.0	90.0	1600.0	
2	31.0	M	SOLTERO(A)	MOTUPE	14400.0	90.0	1440.0	
3	33.0	F	SOLTERO(A)	MOTUPE	20000.0	90.0	2000.0	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 985 entries, 0 to 984
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   EDAD CLIENTE   985 non-null    float64
 1   SEXO CLIENTE   985 non-null    object
 2   EST.CIVIL      985 non-null    object
 3   ZONA PROYECTO  985 non-null    object
 4   VALOR TERRENO  985 non-null    float64
 5   M2 TERRENO     985 non-null    float64
 6   CUOTA_INICIAL  985 non-null    float64
 7   DESISTIMIENTO  985 non-null    int64
 8   VALOR_M2       985 non-null    float64
 9   PCT_CINICIAL   985 non-null    float64
dtypes: float64(6), int64(1), object(3)
memory usage: 77.1+ KB
```

4. Homogenizar valores de columnas

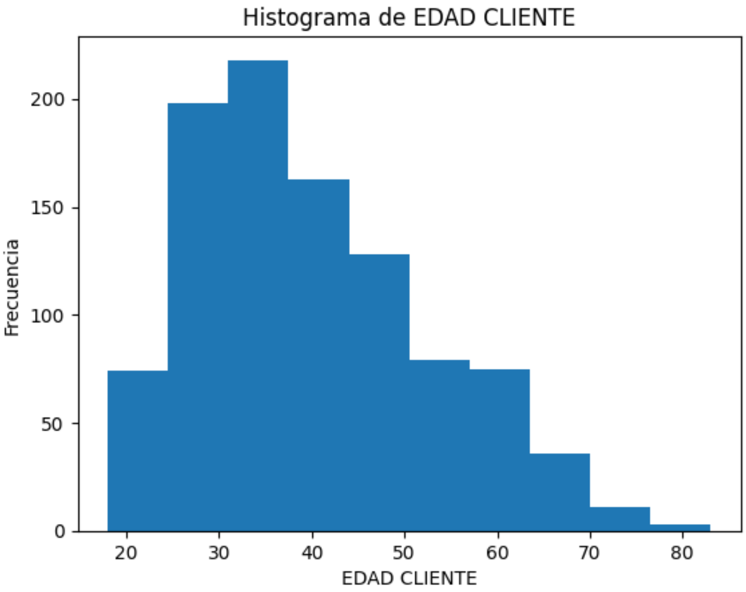
```
# Mostrar las categorías únicas de cada columna

print(df['SEXO CLIENTE'].unique())
#print(df['DEPARTAMENTO'].unique())
#print(df['PROVINCIA'].unique())
```

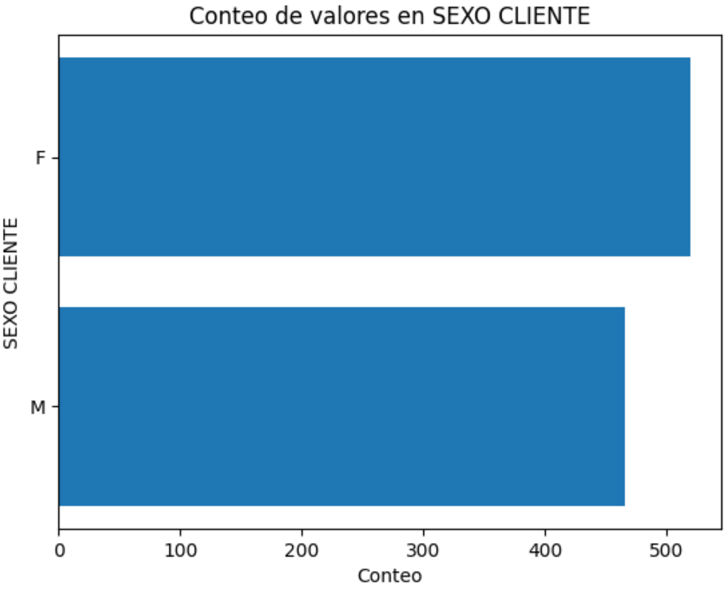
```
print(df['ZONA PROYECTO'].unique())
print(df['EST.CIVIL'].unique())
```

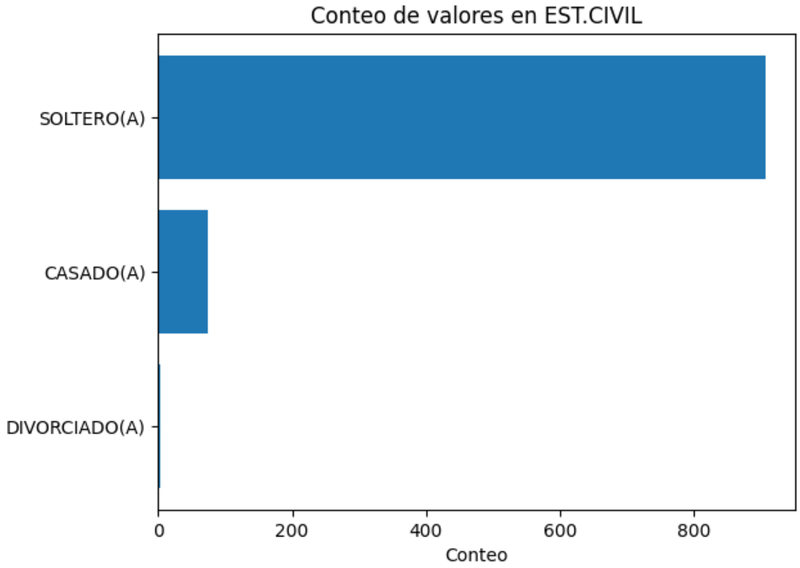
```
['M' 'F']
['MOTUPE' 'PIMENTEL' 'MORROPE' 'PICSI']
['SOLTERO(A)' 'DIVORCIADO(A)' 'CASADO(A)']
```

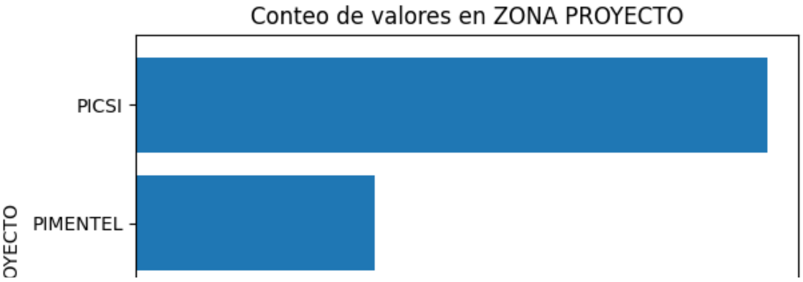
```
# Reempazar los nombres de los Proyectos para un mejor reconocimiento

#df['PROYECTO'] = np.where(df['PROYECTO'] == 'LA PLANICIE DE SANTA CATALINA', 'STA_CATALINA',
#                 np.where(df['PROYECTO'] == 'VALPARAISO DE PIMENTEL'       , 'PIMENTEL',
#                 np.where(df['PROYECTO'] == 'LA ENSENADA DE MÓRROPE'       , 'MORROPE',
#                 np.where(df['PROYECTO'] == 'NUEVO PICSI IV ETAPA'         , 'PICSI_IV',
#                 np.where(df['PROYECTO'] == 'NUEVO PICSI III ETAPA'        , 'PICSI_III',
#                 np.where(df['PROYECTO'] == 'NUEVO PICSI II ETAPA'         , 'PICSI_II',
#                 np.where(df['PROYECTO'] == 'NUEVO PICSI V ETAPA'          , 'PICSI_V', 'PICSI_I')))))))
```

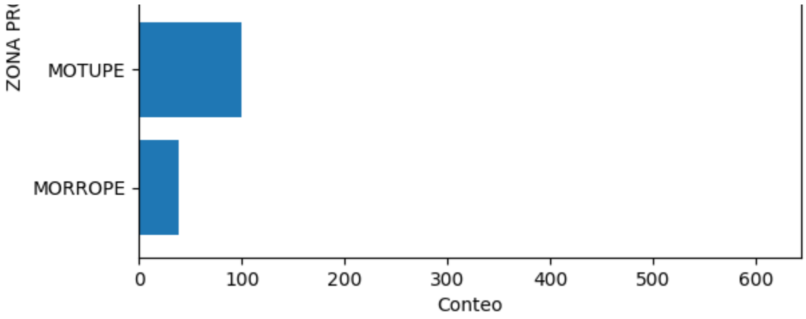
```
for columna in df.columns:
    # Verificar el tipo de la columna
    if df[columna].dtype == 'object':
        # Para variables de tipo "object", generar gráfico de barras horizontales
        conteo_valores = df[columna].value_counts()
        conteo_valores = conteo_valores.sort_values(ascending = True)
        plt.figure()
        plt.barh(conteo_valores.index, conteo_valores.values)
        plt.xlabel('Conteo')
        plt.ylabel(columna)
        plt.title(f'Conteo de valores en {columna}')
        plt.show()
    elif df[columna].dtype in ['int64', 'float64']:
        # Para variables de tipo "int" y "float", generar histograma o gráfico de columnas verticales
        plt.figure()
        plt.hist(df[columna], bins=10)
        plt.xlabel(columna)
        plt.ylabel('Frecuencia')
        plt.title(f'Histograma de {columna}')
        plt.show()
```

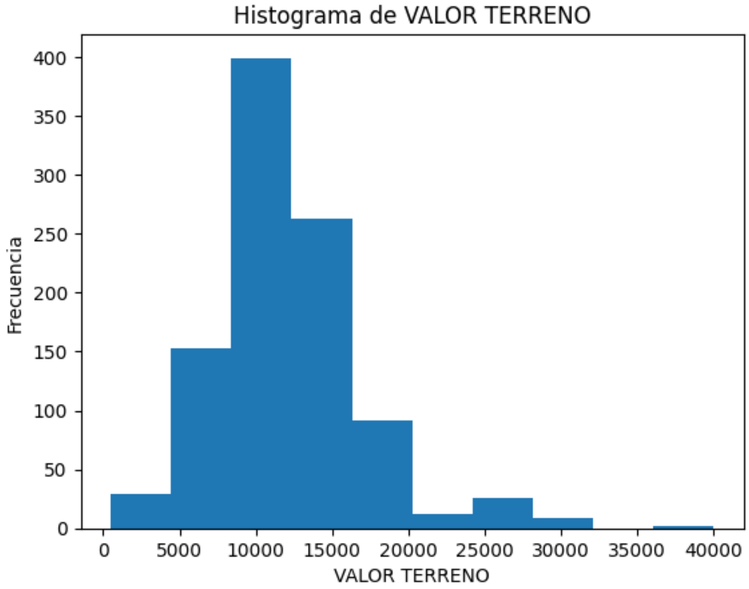

Histograma de EDAD CLIENTE
Frecuencia
0
50
100
150
200
20
30
40
50
60
70
80
EDAD CLIENTE

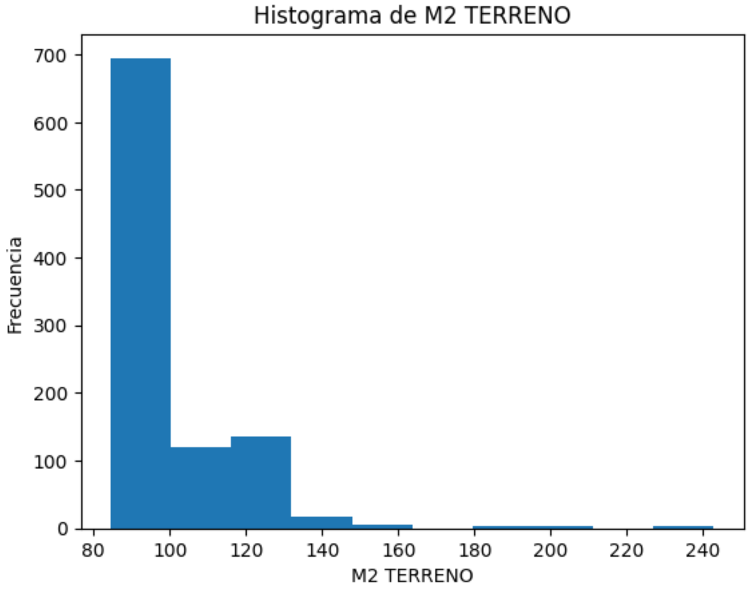

Conteo de valores en SEXO CLIENTE
SEXO CLIENTE
F
M
0
100
200
300
400
500
Conteo

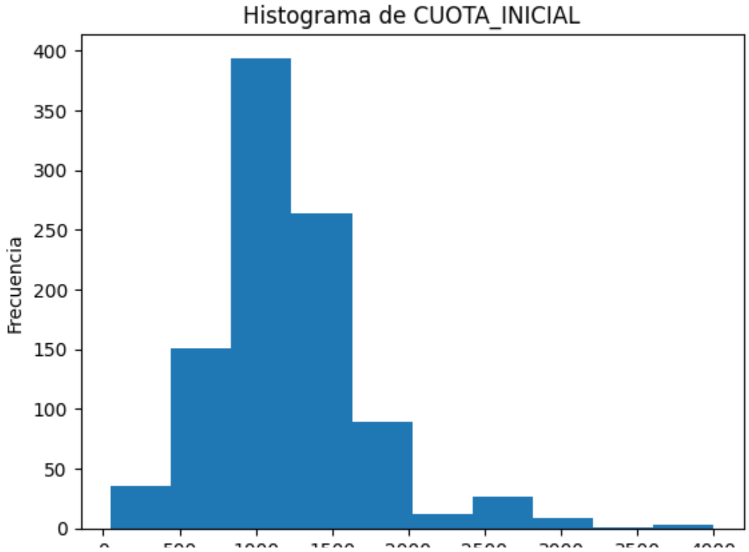

Conteo de valores en EST.CIVIL
EST.CIVIL
SOLTERO(A)
CASADO(A)
DIVORCIADO(A)
0
200
400
600
800
Conteo

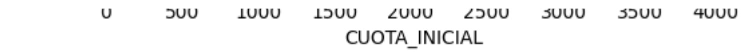

Conteo de valores en ZONA PROYECTO
OYECTO
PICSI
PIMENTEL

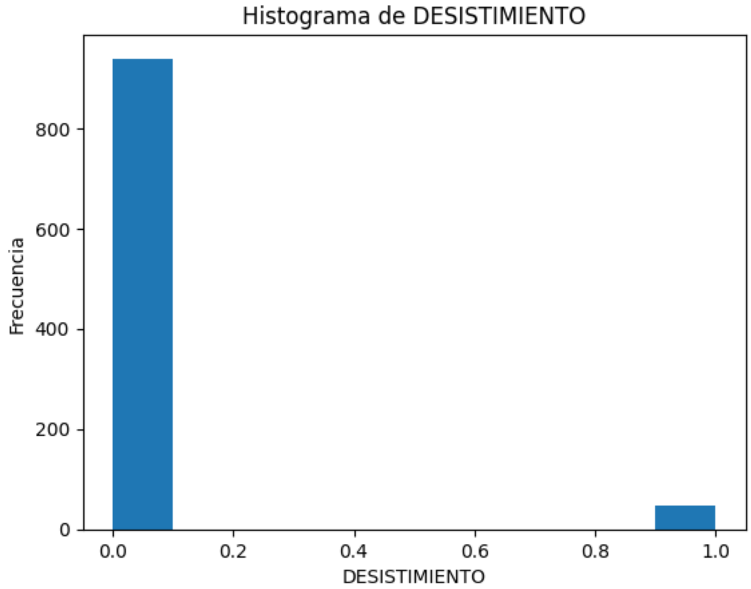



Histograma de VALOR TERRENO


Histograma de M2 TERRENO


Histograma de CUOTA_INICIAL





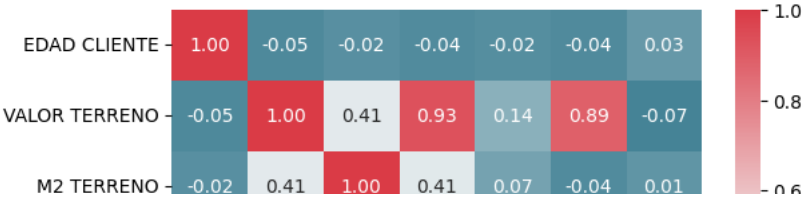
5. Elección de variables que ingresarán al modelo

```
# Listamos las variables resultantes
df.columns
```

```
     Index(['EDAD CLIENTE', 'SEXO CLIENTE', 'EST.CIVIL', 'ZONA PROYECTO',
            'VALOR TERRENO', 'M2 TERRENO', 'CUOTA_INICIAL', 'DESISTIMIENTO',
            'VALOR_M2', 'PCT_CINICIAL'],
           dtype='object')
```

```
columns = ['EDAD CLIENTE', 'SEXO CLIENTE', 'EST.CIVIL', 'ZONA PROYECTO', 'VALOR TERRENO', 'M2 TERRENO', 'DESISTIMIENTO', 'VALOR_
```

```
# Calcular la matriz de correlación
correlation_matrix = df.corr()

# Crear un mapa de colores personalizado
cmap = sns.diverging_palette(220, 10, as_cmap = True)

# Dibujar la matriz de correlación con colores
sns.heatmap(correlation_matrix, cmap = cmap, annot = True, fmt = ".2f")

# Mostrar el gráfico
plt.show()
```



Se obtiene que los campos de VALOR_TERRENO y VALOR_M2 tiene una alta correlación lo cual es normal ya que dependen uno de otro, por lo que al ejecutar la importancia de variables si ambos tienen una importancia considerable solo tomaremos una de ellas

La variable de PROVINCIA no se incluirá en el modelo ya que es una variable categórica con muchos valores distintos pero acumulandose en el valor CHICLAYO

6. Definición del dataframe para el modelo

```
# Variables resultantes para el modelo
features = ['EDAD CLIENTE',
            'SEXO CLIENTE',
            'EST.CIVIL',
            'ZONA PROYECTO',
            'VALOR TERRENO',
            'M2 TERRENO',
            'DESISTIMIENTO',
            'VALOR_M2']

df_modelo = df[features]
df_modelo.head()
```

	EDAD CLIENTE	SEXO CLIENTE	EST.CIVIL	ZONA PROYECTO	VALOR TERRENO	M2 TERRENO	DESISTIMIENTO	VALC
0	39.0	M	SOLTERO(A)	MOTUPE	16012.0	90.0	0	177.9
1	23.0	M	SOLTERO(A)	MOTUPE	16000.0	90.0	0	177.77
2	31.0	M	SOLTERO(A)	MOTUPE	14400.0	90.0	0	160.0C
3	33.0	F	SOLTERO(A)	MOTUPE	20000.0	90.0	0	222.22

```
df_modelo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 985 entries, 0 to 984
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   EDAD CLIENTE   985 non-null    float64
 1   SEXO CLIENTE   985 non-null    object 
 2   EST.CIVIL      985 non-null    object 
 3   ZONA PROYECTO  985 non-null    object 
 4   VALOR TERRENO  985 non-null    float64
 5   M2 TERRENO     985 non-null    float64
 6   DESISTIMIENTO  985 non-null    int64  
 7   VALOR_M2       985 non-null    float64
dtypes: float64(4), int64(1), object(3)
memory usage: 61.7+ KB
```

```
# Variables numéricas
# ==============================================================================
df_modelo.select_dtypes(include=['float64', 'int64']).describe()
```

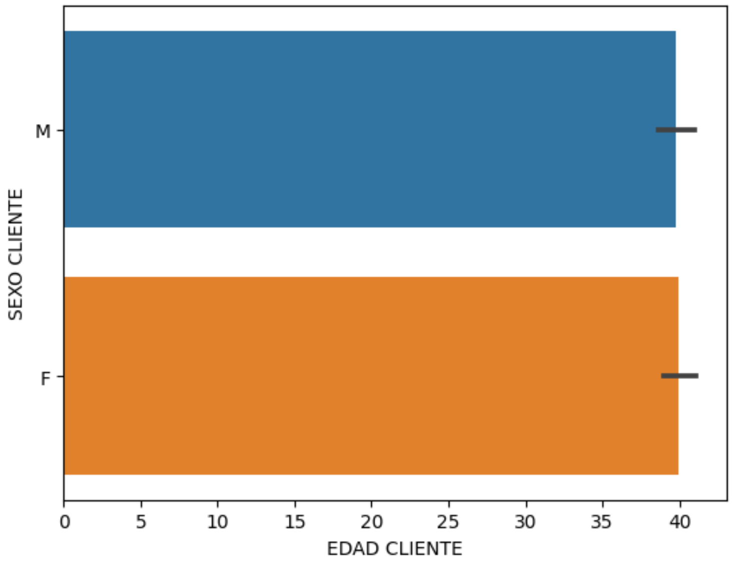
	EDAD CLIENTE	VALOR TERRENO	M2 TERRENO	DESISTIMIENTO	VALOR_M2
count	985.000000	985.000000	985.000000	985.000000	985.000000

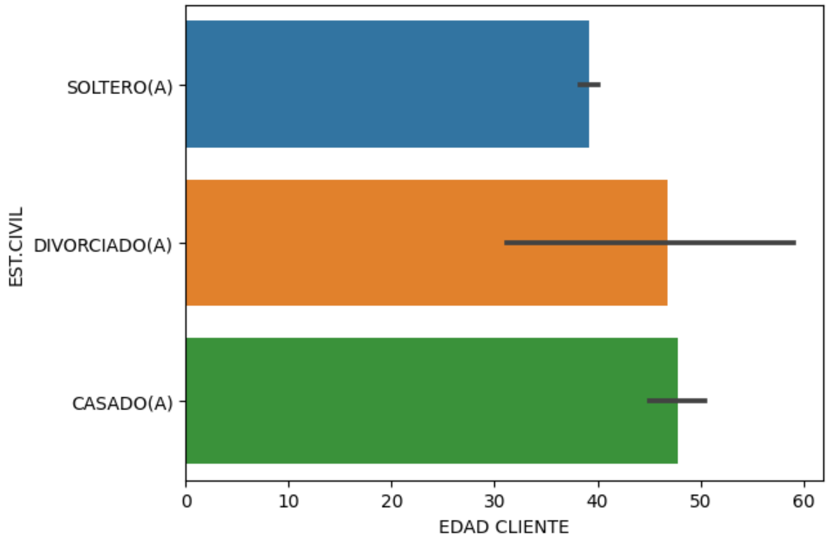
```
# Variables cualitativas (tipo object)
# ==============================================================================
df_modelo.select_dtypes(include=['object']).describe()
```

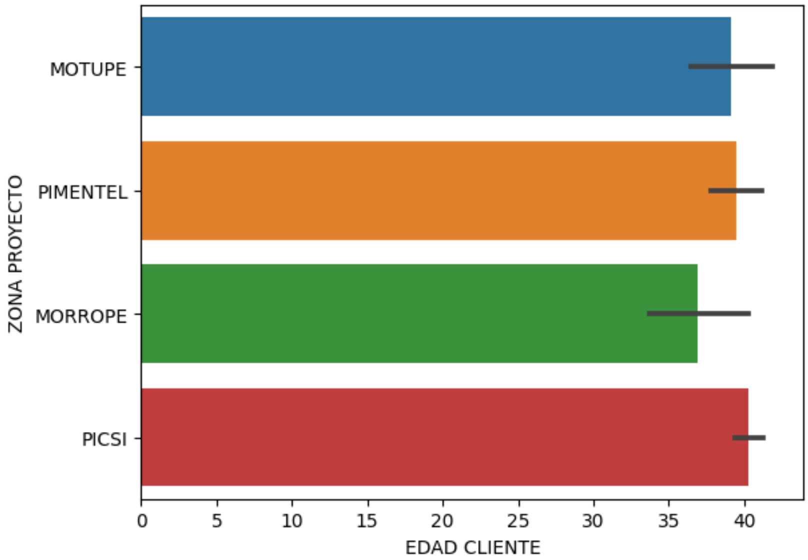
	SEXO CLIENTE	EST.CIVIL	ZONA PROYECTO
count	985	985	985
unique	2	3	4
top	F	SOLTERO(A)	PICSI
freq	519	906	614

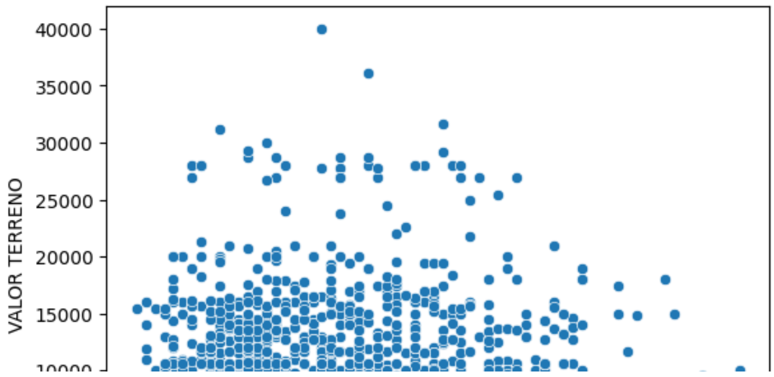
7. Analisis Bivariado de las variables del modelo

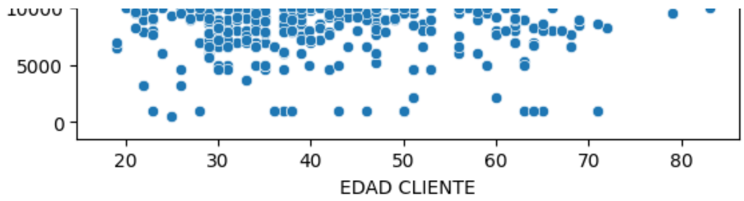
```
# Obtener la lista de variables en el dataframe
variables = df_modelo.columns

# Iterar sobre cada par de variables
for i in range(len(variables)):
    for j in range(i+1, len(variables)):
        variable_1 = variables[i]
        variable_2 = variables[j]

        # Verificar los tipos de las variables
        if df_modelo[variable_1].dtype == 'int64' and df_modelo[variable_2].dtype == 'int64':
            # Realizar un gráfico de dispersión para variables enteras
            sns.scatterplot(x=variable_1, y=variable_2, data=df_modelo)
            plt.show()
        elif df_modelo[variable_1].dtype == 'float64' and df_modelo[variable_2].dtype == 'float64':
            # Realizar un gráfico de dispersión para variables continuas
            sns.scatterplot(x=variable_1, y=variable_2, data=df_modelo)
            plt.show()
        elif df_modelo[variable_1].dtype == 'object' and df_modelo[variable_2].dtype == 'object':
            # Realizar un gráfico de barras para variables categóricas
            crosstab = pd.crosstab(df_modelo[variable_1], df_modelo[variable_2])
            crosstab.plot(kind='bar', stacked=True)
            plt.show()
        else:
            # Realizar un gráfico de barras para variables mixtas
            sns.barplot(x=variable_1, y=variable_2, data=df_modelo)
            plt.show()
```

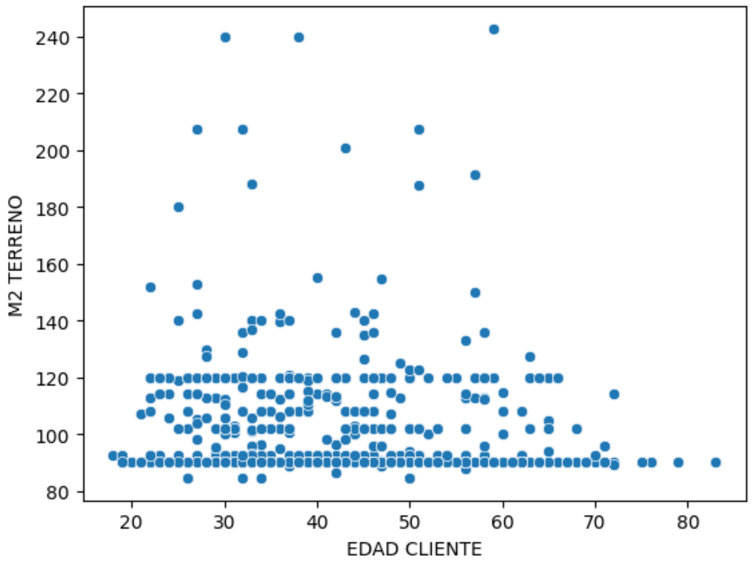

SEXO CLIENTE
M
F
0
5
10
15
20
25
30
35
40
EDAD CLIENTE

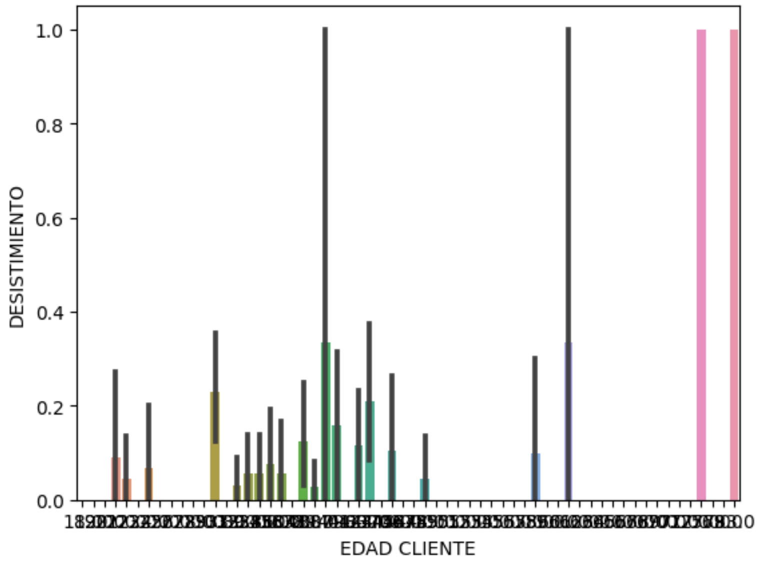

EST.CIVIL
SOLTERO(A)
DIVORCIADO(A)
CASADO(A)
0
10
20
30
40
50
60
EDAD CLIENTE

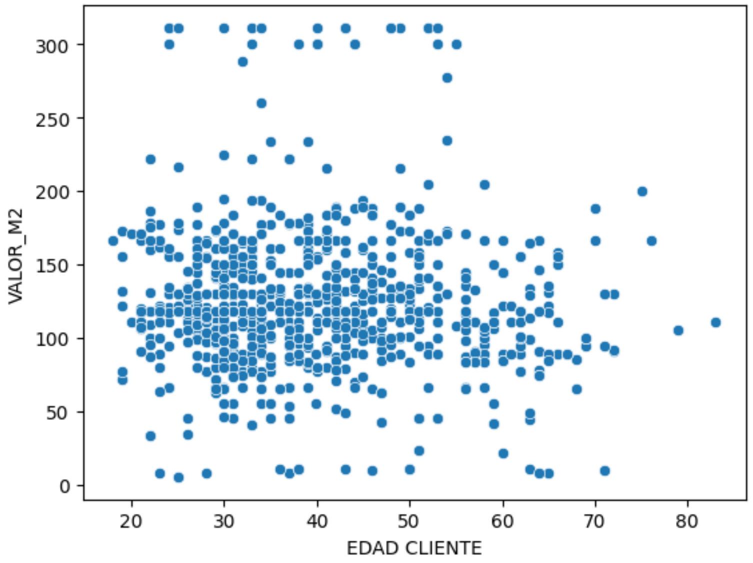

ZONA PROYECTO
MOTUPE
PIMENTEL
MORROPE
PICSI
0
5
10
15
20
25
30
35
40
EDAD CLIENTE


VALOR TERRENO
40000
35000
30000
25000
20000
15000
10000

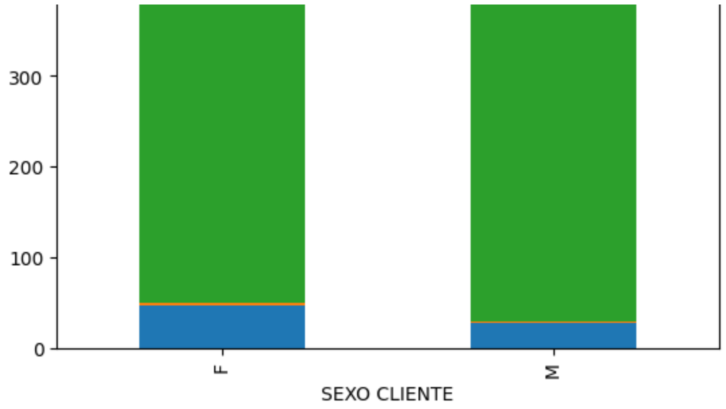

10000
5000
0
20
30
40
50
60
70
80
EDAD CLIENTE

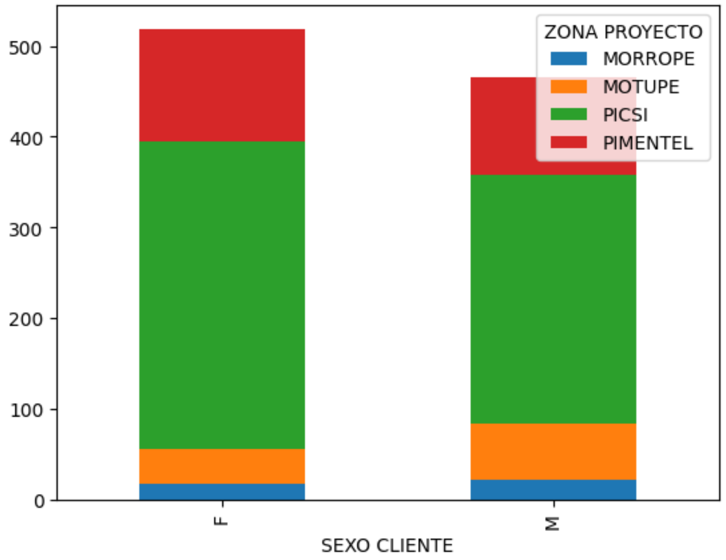

M2 TERRENO
240
220
200
180
160
140
120
100
80
20
30
40
50
60
70
80
EDAD CLIENTE

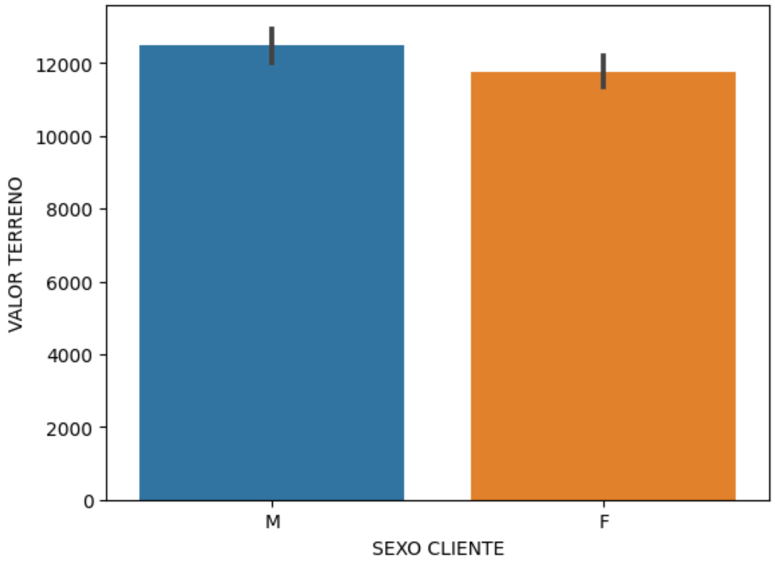

DESISTIMIENTO
1.0
0.8
0.6
0.4
0.2
0.0
EDAD CLIENTE

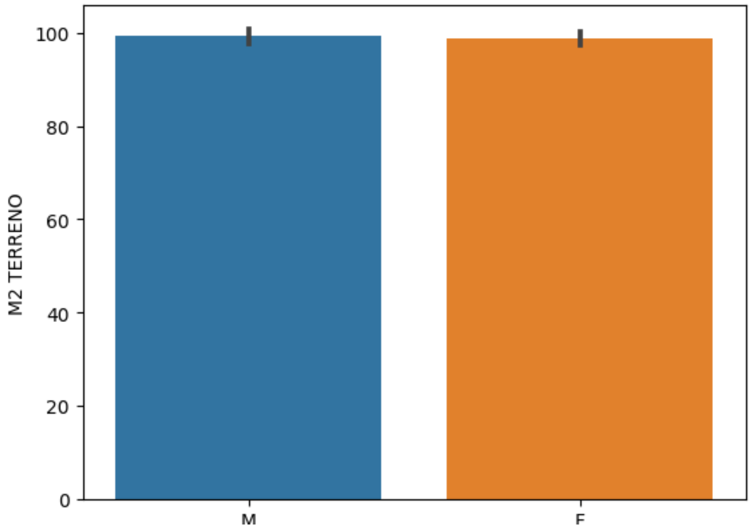

VALOR_M2
300
250
200
150
100
50
0
20
30
40
50
60
70
80
EDAD CLIENTE

500
400
EST.CIVIL
CASADO(A)
DIVORCIADO(A)
SOLTERO(A)

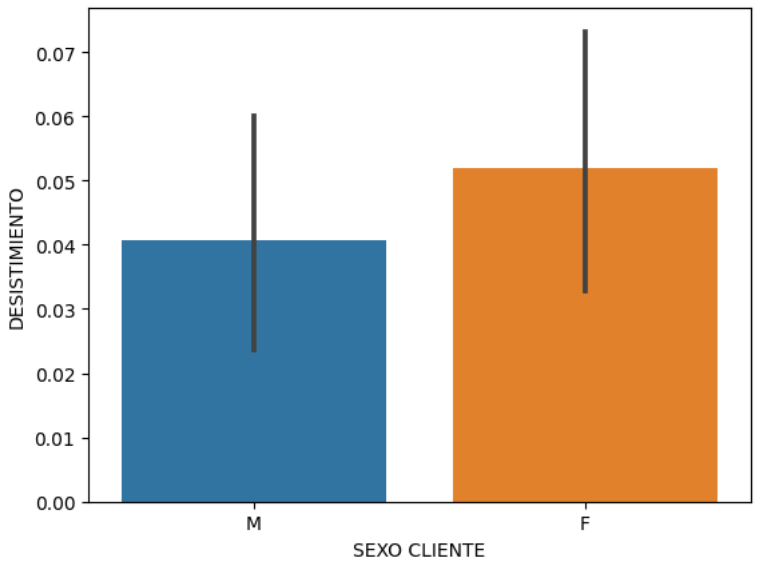

300
200
100
0
F
M
SEXO CLIENTE

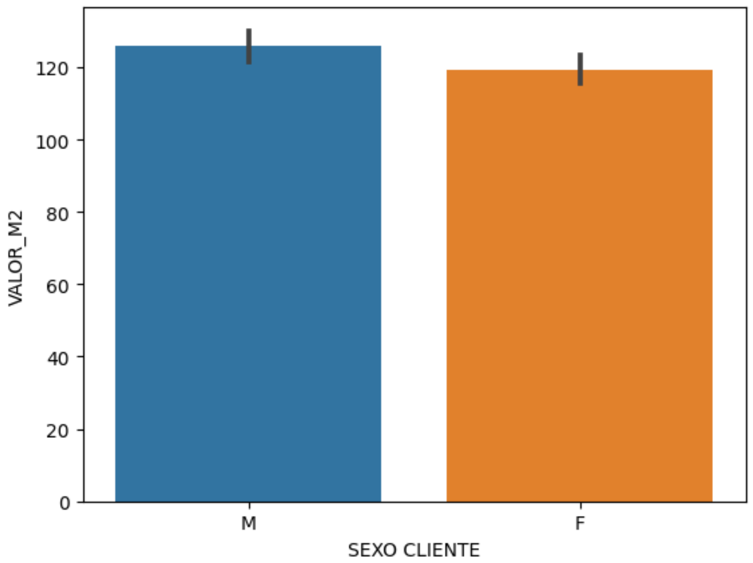

500
400
300
200
100
0
ZONA PROYECTO
MORROPE
MOTUPE
PICSI
PIMENTEL
F
M
SEXO CLIENTE

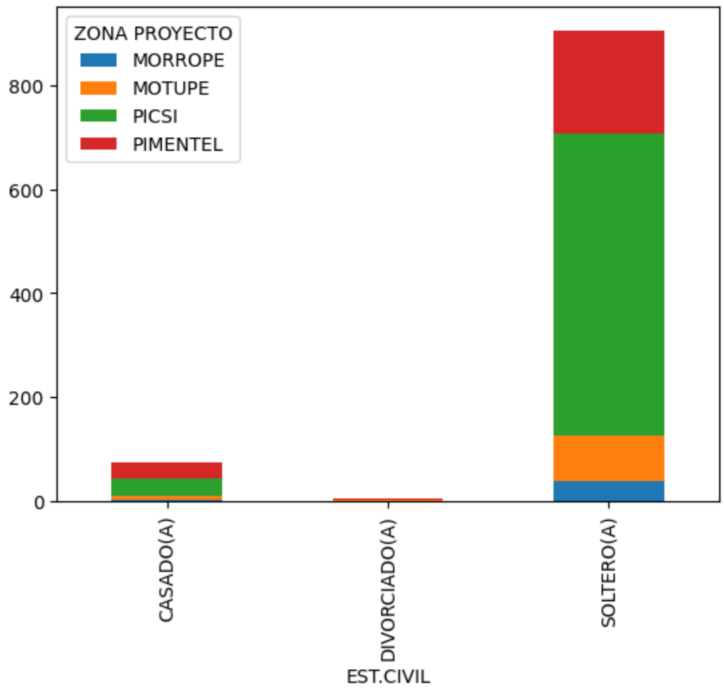

VALOR TERRENO
12000
10000
8000
6000
4000
2000
0
M
F
SEXO CLIENTE

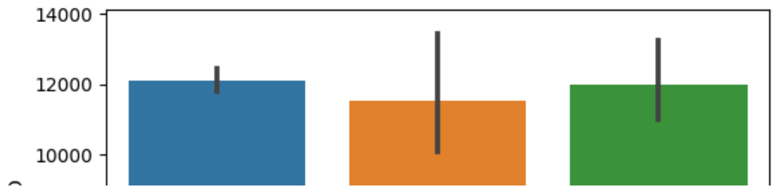

M2 TERRENO
100
80
60
40
20
0
M
F

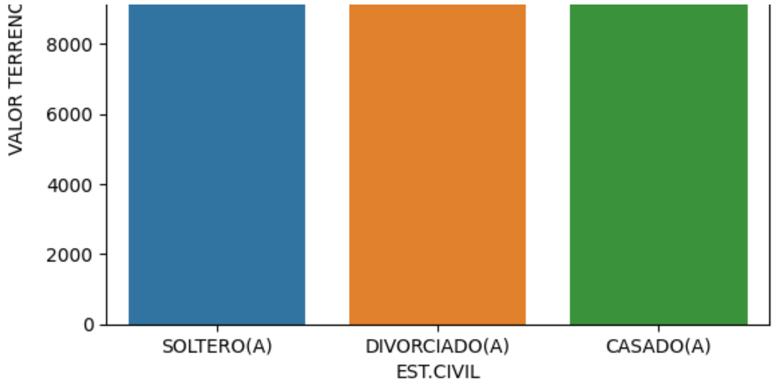
SEXO CLIENTE



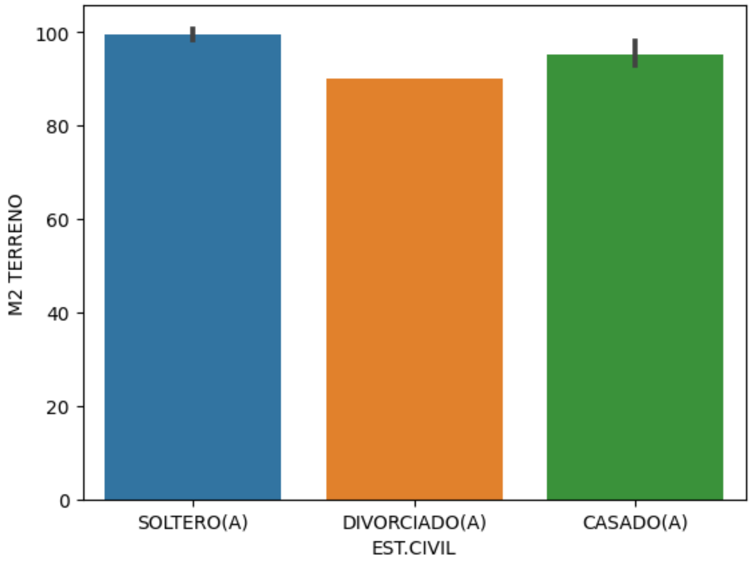




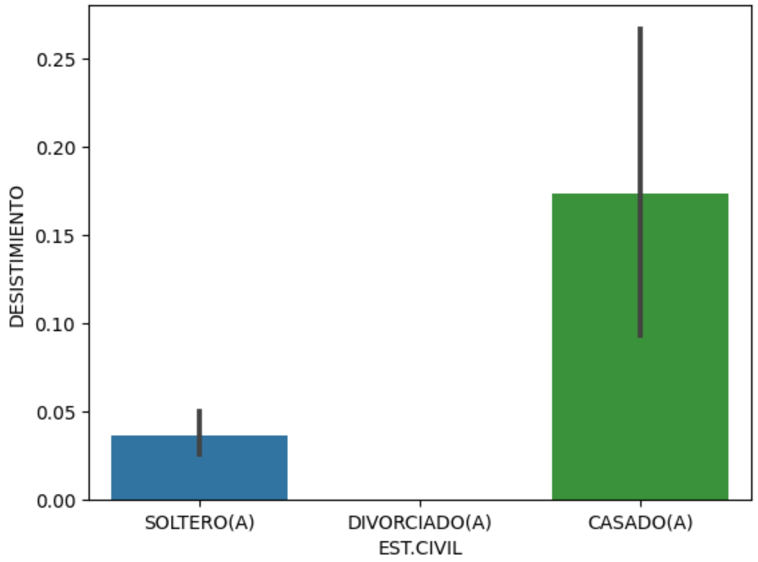




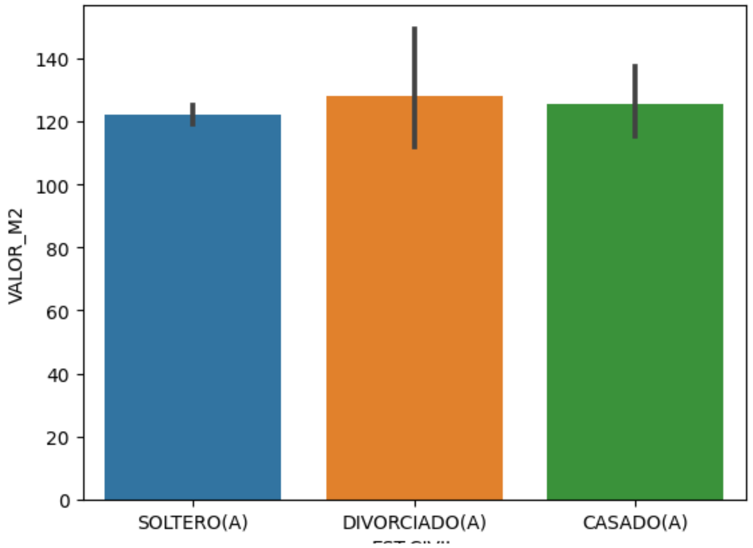
VALOR TERRENC
0
2000
4000
6000
8000
SOLTERO(A)
DIVORCIADO(A)
CASADO(A)
EST.CIVIL



M2 TERRENO
0
20
40
60
80
100
SOLTERO(A)
DIVORCIADO(A)
CASADO(A)
EST.CIVIL

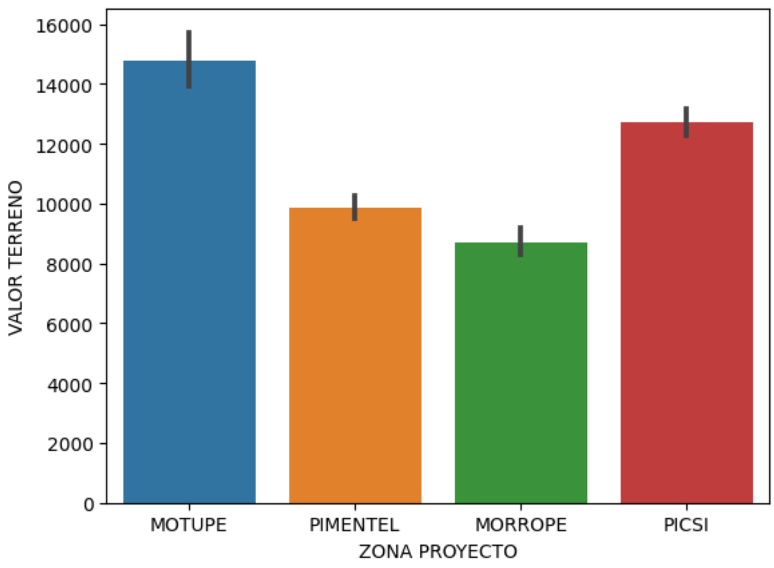


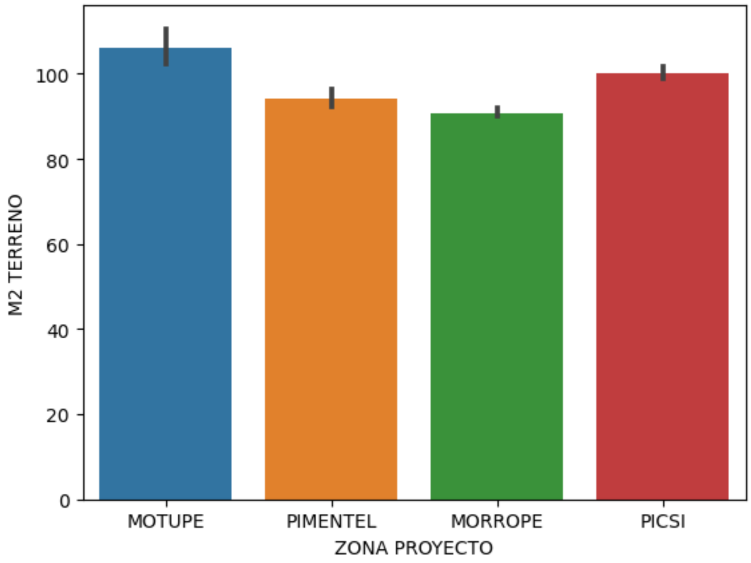
DESISTIMIENTO
0.00
0.05
0.10
0.15
0.20
0.25
SOLTERO(A)
DIVORCIADO(A)
CASADO(A)
EST.CIVIL

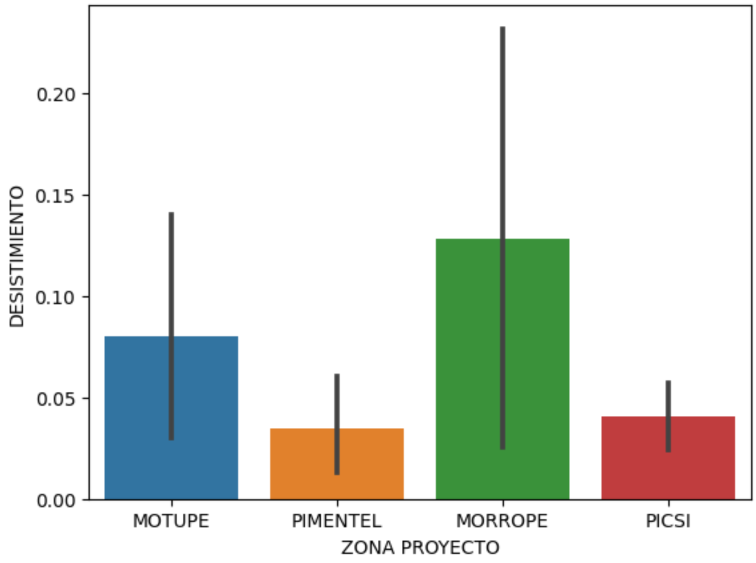


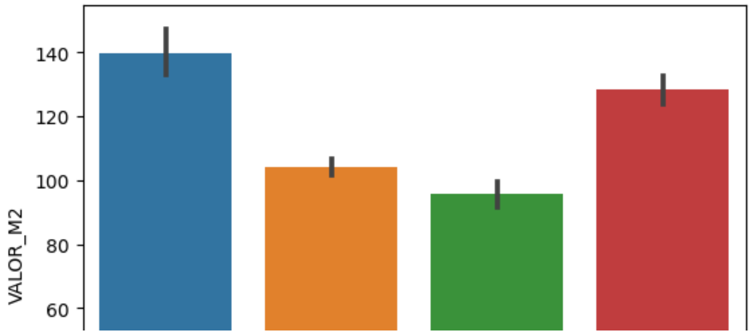
VALOR_M2
0
20
40
60
80
100
120
140
SOLTERO(A)
DIVORCIADO(A)
CASADO(A)

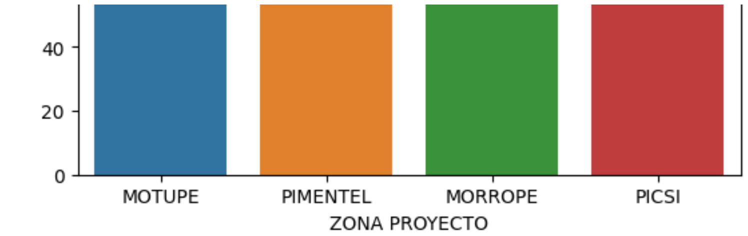
EST.CIVIL

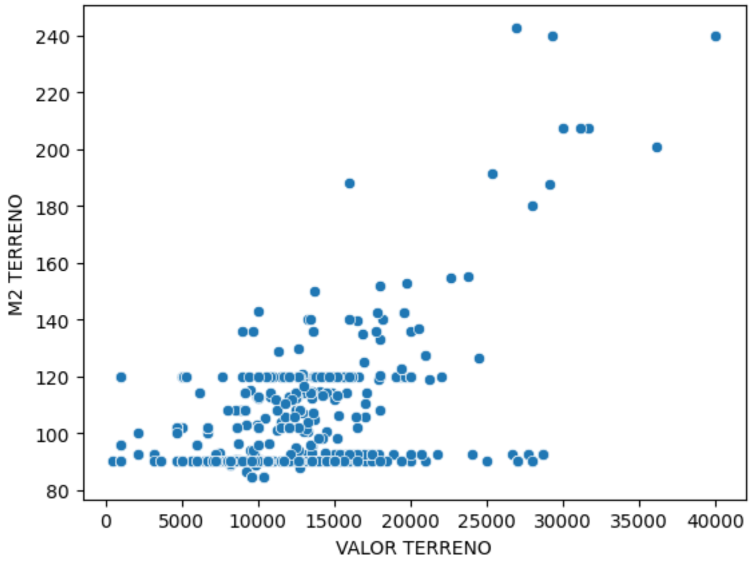

VALOR TERRENO
16000
14000
12000
10000
8000
6000
4000
2000
0
MOTUPE
PIMENTEL
MORROPE
PICSI
ZONA PROYECTO

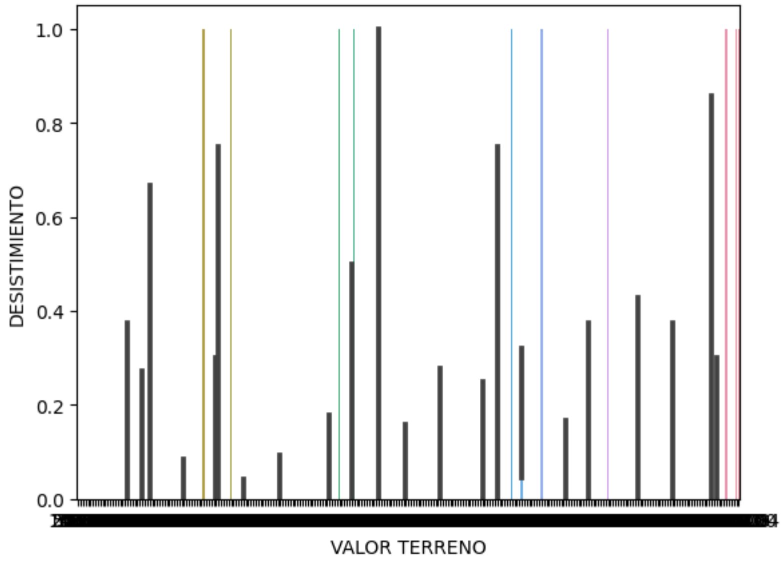

M2 TERRENO
100
80
60
40
20
0
MOTUPE
PIMENTEL
MORROPE
PICSI
ZONA PROYECTO

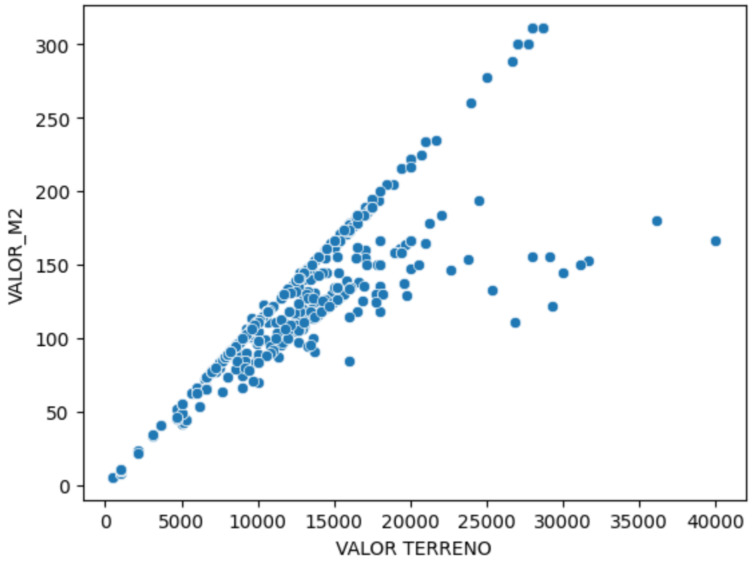

DESISTIMIENTO
0.20
0.15
0.10
0.05
0.00
MOTUPE
PIMENTEL
MORROPE
PICSI
ZONA PROYECTO


VALOR_M2
140
120
100
80
60

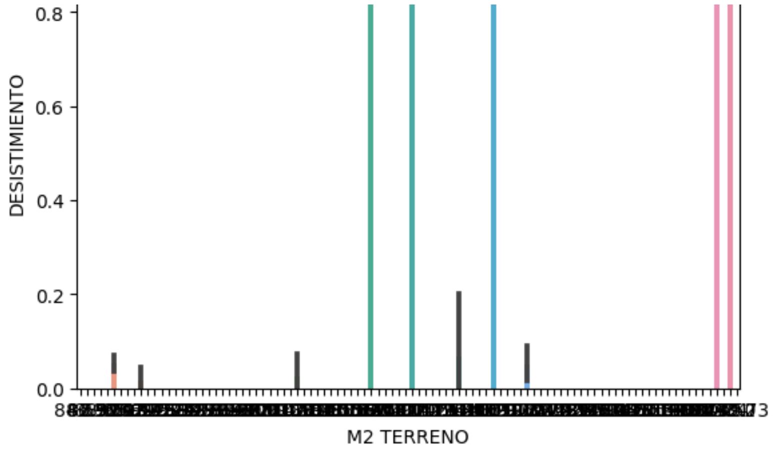

40
20
0
MOTUPE
PIMENTEL
MORROPE
PICSI
ZONA PROYECTO

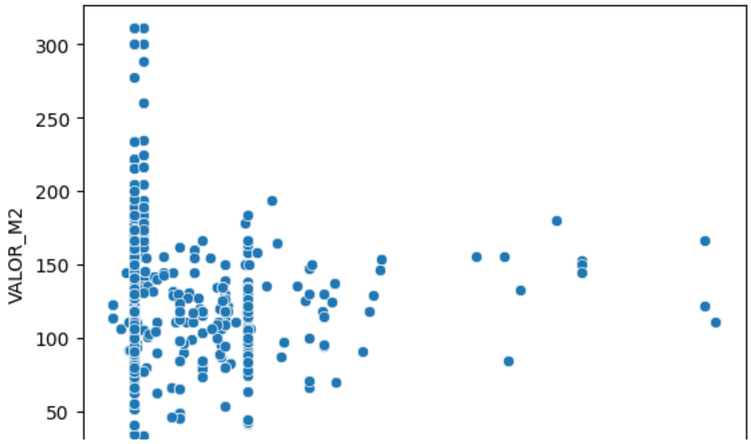

240
220
200
180
160
140
120
100
80
M2 TERRENO
0
5000
10000
15000
20000
25000
30000
35000
40000
VALOR TERRENO


1.0
0.8
0.6
0.4
0.2
0.0
DESISTIMIENTO
VALOR TERRENO


300
250
200
150
100
50
0
VALOR_M2
0
5000
10000
15000
20000
25000
30000
35000
40000
VALOR TERRENO

1.0





```
#Conversión de datos categóricos a numéricos
df_dummies = pd.get_dummies(df_modelo, columns = [
    'SEXO CLIENTE',
    'EST.CIVIL',
    'ZONA PROYECTO'
])

#Vemos el dataframe
df_dummies.columns
```

```
Index(['EDAD CLIENTE', 'VALOR TERRENO', 'M2 TERRENO', 'DESISTIMIENTO',
       'VALOR_M2', 'SEXO CLIENTE_F', 'SEXO CLIENTE_M', 'EST.CIVIL_CASADO(A)',
       'EST.CIVIL_DIVORCIADO(A)', 'EST.CIVIL_SOLTERO(A)',
       'ZONA PROYECTO_MORROPE', 'ZONA PROYECTO_MOTUPE', 'ZONA PROYECTO_PICSI',
       'ZONA PROYECTO_PIMENTEL'],
      dtype='object')
```

```
# Reindexar los registros finales
df_dummies.reset_index(drop=True, inplace=True)
df_dummies.head()
```

	EDAD CLIENTE	VALOR TERRENO	M2 TERRENO	DESISTIMIENTO	VALOR_M2	SEXO CLIENTE_F	SEXO CLIENTE_M	EST.
0	39.0	16012.0	90.0	0	177.911111	0	1	
1	23.0	16000.0	90.0	0	177.777778	0	1	
2	31.0	14400.0	90.0	0	160.000000	0	1	
3	33.0	20000.0	90.0	0	222.222222	1	0	
4	51.0	15000.0	90.0	0	166.666667	1	0	

```
features_mod = ['EDAD CLIENTE',
                'VALOR TERRENO',
                'M2 TERRENO',
                #'DESISTIMIENTO',
                'VALOR_M2',
                'SEXO CLIENTE_F',
                'SEXO CLIENTE_M',
                'EST.CIVIL_CASADO(A)',
                'EST.CIVIL_DIVORCIADO(A)',
                'EST.CIVIL_SOLTERO(A)',
                'ZONA PROYECTO_MORROPE',
                'ZONA PROYECTO_MOTUPE',
                'ZONA PROYECTO_PICSI',
                'ZONA PROYECTO_PIMENTEL']
```

8. Desarrollo del modelo

```
# Definimos las variables que comprenderán el X o los estimadores y el Y que se buscará predecir

X = df_dummies[features_mod]
Y = df_dummies['DESISTIMIENTO']
```

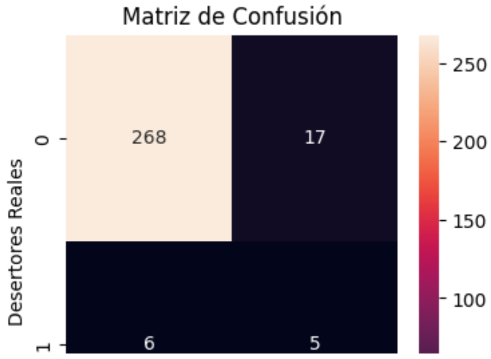
```
#dividimos en sets de entrenamiento y test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.3, random_state = 666)
```

```
from sklearn.utils import class_weight

class_weights = class_weight.compute_class_weight(class_weight = 'balanced', classes = np.unique(y_train), y = y_train)
class_weights_dict = dict(zip(np.unique(y_train), class_weights))
print(class_weights)
print(class_weights_dict)
```

```
    [0.52675841 9.84285714]
    {0: 0.5267584097859327, 1: 9.842857142857143}
```

```
#===================================================================================#
#=================================ARBOL DE DECISION=================================#
#===================================================================================#

#creamos una función que crea el modelo que usaremos cada vez
def run_model(X_train, X_test, y_train, y_test):
    clf_base = DecisionTreeClassifier(criterion = 'gini',
                                      #min_samples_split = 10,
                                      #class_weight = class_weights_dict,
                                      class_weight = {0:1, 1:2.5},
                                      random_state = 666,
                                      splitter = 'best')
    clf_base.fit(X_train, y_train)
    return clf_base

#ejecutamos el modelo "tal cual"
model_tree = run_model(X_train, X_test, y_train, y_test)

#definimos funcion para mostrar los resultados
def mostrar_resultados(y_test, pred_y):
    conf_matrix = confusion_matrix(y_test, pred_y)
    plt.figure(figsize=(4, 4))
    sns.heatmap(conf_matrix, annot=True, fmt="d");
    plt.title("Matriz de Confusión")
    plt.ylabel('Desertores Reales')
    plt.xlabel('Desertores Calculados')
    plt.show()
    print (classification_report(y_test, pred_y))


print(f"Profundidad del árbol: {model_tree.get_depth()}")
print(f"Número de nodos terminales: {model_tree.get_n_leaves()}")
pred_y = model_tree.predict(X_test)
mostrar_resultados(y_test, pred_y)
```

```
Profundidad del árbol: 16
Número de nodos terminales: 69
```



```
# Importancia de variables de cualquier modelo

importancia_predictores = pd.DataFrame(
                            {'predictor': features_mod,
                             'importancia': model_tree.feature_importances_}
                            )
print("Importancia de los predictores en el modelo")
print("-------------------------------------------")
top_var = importancia_predictores.sort_values('importancia', ascending = False)

top_var.reset_index(drop = True, inplace = True)
top_var.head(50)
```

```
Importancia de los predictores en el modelo
-------------------------------------------
```

	predictor	importancia
0	EDAD CLIENTE	0.329787
1	VALOR TERRENO	0.226476
2	VALOR_M2	0.197319
3	M2 TERRENO	0.078632
4	SEXO CLIENTE_F	0.048697
5	ZONA PROYECTO_MOTUPE	0.037898
6	EST.CIVIL_CASADO(A)	0.031103
7	ZONA PROYECTO_MORROPE	0.025507
8	ZONA PROYECTO_PICSI	0.016089
9	SEXO CLIENTE_M	0.008493
10	EST.CIVIL_DIVORCIADO(A)	0.000000
11	EST.CIVIL_SOLTERO(A)	0.000000
12	ZONA PROYECTO_PIMENTEL	0.000000

```
!pip install graphviz
!pip install pydotplus
```

```
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (0.20.1)
Requirement already satisfied: pydotplus in /usr/local/lib/python3.10/dist-packages (2.0.2)
Requirement already satisfied: pyparsing>=2.0.1 in /usr/local/lib/python3.10/dist-packages (from pydotplus) (3.1.0)
```

```
from sklearn.tree import export_graphviz
import graphviz
import pydotplus

# Crear una representación gráfica del árbol
dot_data = export_graphviz(model_tree, out_file = None,
                           feature_names = features_mod,
                           class_names = ['NoDefault', 'Default'],
                           filled = True, rounded = True,
                           special_characters = True,
                           impurity = False,
                           proportion = True,
                           precision = 2,
                           label = 'all')

graph = graphviz.Source(dot_data)
```

```
# Crear una imagen PNG del árbol
graph.render("Model_Tree_Inmobiliaria")

# Crear una representación gráfica del árbol solo con los casos proyectados correctamente
dot_data_acc = export_graphviz(model_tree,
                               out_file = None,
                               feature_names = features_mod,
                               class_names = ['NoDefault', 'Default'],
                               filled = True,
                               rounded = True,
                               special_characters = True,
                               impurity = False,
                               proportion = False,
                               #precision = 1,
                               label = 'all',
                               leaves_parallel = True,
                               #max_depth = 7,
                               node_ids = True,
                               rotate = False)

graph_acc = graphviz.Source(dot_data_acc)

# Crear una imagen PNG del árbol solo con los casos proyectados correctamente
graph_acc.render("Model_Tree_Inmobiliaria_ACC")
```

```
     'Model_Tree_Inmobiliaria_ACC.pdf'
```

```
from sklearn.tree import plot_tree

# Estructura del árbol creado
# ------------------------------------------------------------------------------
fig, ax = plt.subplots(figsize = (70, 30))

print(f"Profundidad del árbol: {model_tree.get_depth()}")
print(f"Número de nodos terminales: {model_tree.get_n_leaves()}")

plot = plot_tree(
            decision_tree = model_tree,
            feature_names = features_mod,
            #max_depth     = 7,
            class_names   = ['NoDefault', 'Default'],
            filled        = True,
            impurity      = False,
            rounded       = True,
            fontsize      = 12,
            precision     = 2,
            ax            = ax
       )
plt.show()
```

```
Profundidad del árbol: 16
Número de nodos terminales: 69
```

9. Aplicación de modelos BOOST

```
# Crear un objeto Dataset de LightGBM
train_data = lgb.Dataset(X_train, label = y_train)

# Definir los parámetros del modelo
params = {
    'objective': 'binary',
    'metric': 'binary_logloss'
}

# Entrenar el modelo de LightGBM
model = lgb.train(params, train_data, num_boost_round = 100)
```

```
[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Number of positive: 35, number of negative: 654
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.000347 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 326
[LightGBM] [Info] Number of data points in the train set: 689, number of used features: 12
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.050798 -> initscore=-2.927759
[LightGBM] [Info] Start training from score -2.927759
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

10. Importancia de variables segun LigthGBM

```
# Obtener la importancia de las variables
feature_importance = model.feature_importance()

# Obtener el nombre de las variables
feature_names = X.columns

# Crear un DataFrame con la importancia de las variables
importance_df = pd.DataFrame({'Variable': feature_names, 'Importance': feature_importance})

# Ordenar el DataFrame por importancia descendente
importance_df = importance_df.sort_values('Importance', ascending = False)

# Reindexar los registros finales
importance_df.reset_index(drop=True, inplace=True)

# Listar la importancia de variables
importance_df.head(50)
```

	Variable	Importance
0	EDAD CLIENTE	947
1	VALOR TERRENO	758
2	VALOR_M2	593
3	M2 TERRENO	155
4	EST.CIVIL_CASADO(A)	104
5	SEXO CLIENTE_F	86
6	ZONA PROYECTO_PICSI	80
7	ZONA PROYECTO_PIMENTEL	64
8	ZONA PROYECTO_MORROPE	50
9	SEXO CLIENTE_M	35

```
# Obtener la importancia de las variables
importancia_predictores = pd.DataFrame({
    'predictor': feature_names,
    'importancia': model.feature_importance(importance_type = 'gain')
})

print("Importancia de los predictores en el modelo")
print("-------------------------------------------")
top_var = importancia_predictores.sort_values('importancia', ascending = False)
top_var.reset_index(drop = True, inplace = True)
top_var.head(15)
```

```
Importancia de los predictores en el modelo
-------------------------------------------
```

	predictor	importancia
0	EDAD CLIENTE	475.200364
1	VALOR TERRENO	332.178709
2	VALOR_M2	298.930417
3	M2 TERRENO	70.603167
4	EST.CIVIL_CASADO(A)	66.830220
5	SEXO CLIENTE_F	33.321491
6	ZONA PROYECTO_PICSI	30.431785
7	ZONA PROYECTO_PIMENTEL	28.986277
8	ZONA PROYECTO_MORROPE	24.851121
9	SEXO CLIENTE_M	15.208981
10	EST.CIVIL_SOLTERO(A)	4.102297
11	ZONA PROYECTO_MOTUPE	2.730236
12	EST.CIVIL_DIVORCIADO(A)	0.000000

```
# Filtrar los registros del DataFrame
registros_filtrados_1 = importance_df.loc[importance_df['Importance'] >= 10]

registros_filtrados_2 = top_var.loc[top_var['importancia'] >= 10]

# Mostrar los registros filtrados
print(registros_filtrados_1)
print('#############################')
print(registros_filtrados_2)
```

```
                  Variable  Importance
0             EDAD CLIENTE         947
1            VALOR TERRENO         758
2                 VALOR_M2         593
3               M2 TERRENO         155
4      EST.CIVIL_CASADO(A)         104
5           SEXO CLIENTE_F          86
6      ZONA PROYECTO_PICSI          80
7   ZONA PROYECTO_PIMENTEL          64
8    ZONA PROYECTO_MORROPE          50
9           SEXO CLIENTE_M          35
10    ZONA PROYECTO_MOTUPE          31
11    EST.CIVIL_SOLTERO(A)          13
#############################
                 predictor  importancia
```

```
     0           EDAD CLIENTE   475.200364
     1          VALOR TERRENO   332.178709
     2               VALOR_M2   298.930417
     3             M2 TERRENO    70.603167
     4     EST.CIVIL_CASADO(A)    66.830220
     5          SEXO CLIENTE_F    33.321491
     6     ZONA PROYECTO_PICSI    30.431785
     7  ZONA PROYECTO_PIMENTEL    28.986277
     8   ZONA PROYECTO_MORROPE    24.851121
     9          SEXO CLIENTE_M    15.208981
```

```
features_importance = registros_filtrados_1['Variable'].unique().tolist()
features_importance
```

```
     ['EDAD CLIENTE',
      'VALOR TERRENO',
      'VALOR_M2',
      'M2 TERRENO',
      'EST.CIVIL_CASADO(A)',
      'SEXO CLIENTE_F',
      'ZONA PROYECTO_PICSI',
      'ZONA PROYECTO_PIMENTEL',
      'ZONA PROYECTO_MORROPE',
      'SEXO CLIENTE_M',
      'ZONA PROYECTO_MOTUPE',
      'EST.CIVIL_SOLTERO(A)']
```

11. Evaluación de los modelos BOOSTING

```
# Definir los parámetros de cada modelo
lgb_params = {
    'objective': 'binary',
    'metric': 'binary_logloss'
}

cat_params = {
    'loss_function': 'Logloss',
    'eval_metric': 'Logloss'
}

xgb_params = {
    'objective': 'binary:logistic',
    'eval_metric': 'logloss'
}

# Entrenar los modelos
lgb_model = lgb.LGBMClassifier(**lgb_params)
lgb_model.fit(X_train, y_train)

cat_model = CatBoostClassifier(**cat_params)
cat_model.fit(X_train, y_train)

xgb_model = xgb.XGBClassifier(**xgb_params)
xgb_model.fit(X_train, y_train)
```

```
Learning rate set to 0.008787
0:      learn: 0.6806003        total: 49.6ms   remaining: 49.5s
1:      learn: 0.6688052        total: 51ms     remaining: 25.5s
2:      learn: 0.6572793        total: 53ms     remaining: 17.6s
3:      learn: 0.6456499        total: 56ms     remaining: 13.9s
4:      learn: 0.6338418        total: 59.2ms   remaining: 11.8s
5:      learn: 0.6227366        total: 61.5ms   remaining: 10.2s
6:      learn: 0.6106921        total: 63.9ms   remaining: 9.07s
7:      learn: 0.6004273        total: 67.3ms   remaining: 8.34s
8:      learn: 0.5900435        total: 70.5ms   remaining: 7.76s
9:      learn: 0.5798075        total: 73.8ms   remaining: 7.3s
10:     learn: 0.5697453        total: 77.1ms   remaining: 6.93s
11:     learn: 0.5609059        total: 79.9ms   remaining: 6.58s
12:     learn: 0.5516582        total: 82.6ms   remaining: 6.27s
13:     learn: 0.5430355        total: 86ms     remaining: 6.06s
14:     learn: 0.5337131        total: 89.2ms   remaining: 5.86s
15:     learn: 0.5253804        total: 92.7ms   remaining: 5.7s
16:     learn: 0.5169766        total: 97.7ms   remaining: 5.65s
17:     learn: 0.5087136        total: 102ms    remaining: 5.54s
18:     learn: 0.5011074        total: 110ms    remaining: 5.66s
19:     learn: 0.4935670        total: 114ms    remaining: 5.57s
20:     learn: 0.4858525        total: 118ms    remaining: 5.5s
21:     learn: 0.4780787        total: 122ms    remaining: 5.4s
22:     learn: 0.4707516        total: 125ms    remaining: 5.3s
23:     learn: 0.4638068        total: 128ms    remaining: 5.22s
24:     learn: 0.4574120        total: 131ms    remaining: 5.1s
25:     learn: 0.4505275        total: 133ms    remaining: 4.97s
26:     learn: 0.4438619        total: 135ms    remaining: 4.88s
27:     learn: 0.4372580        total: 139ms    remaining: 4.82s
28:     learn: 0.4312048        total: 142ms    remaining: 4.76s
29:     learn: 0.4254363        total: 145ms    remaining: 4.7s
30:     learn: 0.4196361        total: 149ms    remaining: 4.65s
31:     learn: 0.4138263        total: 152ms    remaining: 4.6s
32:     learn: 0.4079558        total: 155ms    remaining: 4.55s
33:     learn: 0.4020067        total: 159ms    remaining: 4.51s
34:     learn: 0.3965236        total: 162ms    remaining: 4.46s
35:     learn: 0.3916190        total: 165ms    remaining: 4.42s
36:     learn: 0.3862966        total: 169ms    remaining: 4.39s
37:     learn: 0.3814466        total: 178ms    remaining: 4.51s
38:     learn: 0.3767872        total: 181ms    remaining: 4.47s
39:     learn: 0.3721808        total: 187ms    remaining: 4.48s
40:     learn: 0.3675867        total: 191ms    remaining: 4.47s
41:     learn: 0.3632170        total: 195ms    remaining: 4.45s
42:     learn: 0.3590124        total: 199ms    remaining: 4.42s
43:     learn: 0.3550781        total: 202ms    remaining: 4.4s
44:     learn: 0.3508654        total: 206ms    remaining: 4.37s
45:     learn: 0.3471225        total: 209ms    remaining: 4.34s
46:     learn: 0.3425168        total: 213ms    remaining: 4.32s
47:     learn: 0.3384495        total: 217ms    remaining: 4.3s
48:     learn: 0.3348788        total: 220ms    remaining: 4.28s
49:     learn: 0.3314146        total: 224ms    remaining: 4.25s
50:     learn: 0.3278320        total: 227ms    remaining: 4.22s
51:     learn: 0.3231996        total: 230ms    remaining: 4.2s
52:     learn: 0.3201745        total: 233ms    remaining: 4.15s
53:     learn: 0.3168771        total: 235ms    remaining: 4.11s
54:     learn: 0.3138736        total: 238ms    remaining: 4.09s
55:     learn: 0.3105197        total: 242ms    remaining: 4.07s
56:     learn: 0.3074146        total: 245ms    remaining: 4.05s
57:     learn: 0.3043740        total: 248ms    remaining: 4.03s
58:     learn: 0.3017405        total: 250ms    remaining: 3.99s
59:     learn: 0.2986836        total: 252ms    remaining: 3.96s
60:     learn: 0.2960496        total: 256ms    remaining: 3.94s
61:     learn: 0.2930283        total: 259ms    remaining: 3.92s
62:     learn: 0.2901948        total: 264ms    remaining: 3.92s
63:     learn: 0.2869203        total: 266ms    remaining: 3.89s
64:     learn: 0.2847009        total: 269ms    remaining: 3.87s
65:     learn: 0.2823793        total: 271ms    remaining: 3.83s
66:     learn: 0.2799974        total: 274ms    remaining: 3.81s
67:     learn: 0.2774788        total: 277ms    remaining: 3.79s
68:     learn: 0.2750986        total: 280ms    remaining: 3.78s
69:     learn: 0.2729807        total: 282ms    remaining: 3.75s
70:     learn: 0.2701872        total: 285ms    remaining: 3.73s
71:     learn: 0.2682720        total: 289ms    remaining: 3.72s
72:     learn: 0.2656625        total: 291ms    remaining: 3.69s
73:     learn: 0.2635050        total: 292ms    remaining: 3.65s
74:     learn: 0.2616309        total: 295ms    remaining: 3.64s
75:     learn: 0.2592971        total: 297ms    remaining: 3.62s
76:     learn: 0.2573204        total: 301ms    remaining: 3.6s
77:     learn: 0.2554768        total: 303ms    remaining: 3.58s
78:     learn: 0.2538749        total: 304ms    remaining: 3.55s
79:     learn: 0.2520578        total: 306ms    remaining: 3.52s
80:     learn: 0.2503382        total: 309ms    remaining: 3.51s
81:     learn: 0.2487030        total: 313ms    remaining: 3.5s
82:     learn: 0.2468657        total: 316ms    remaining: 3.49s
83:     learn: 0.2452338        total: 320ms    remaining: 3.49s
84:     learn: 0.2435871        total: 331ms    remaining: 3.56s
85:     learn: 0.2419526        total: 336ms    remaining: 3.57s
86:     learn: 0.2405095        total: 338ms    remaining: 3.55s
87:     learn: 0.2388977        total: 344ms    remaining: 3.57s
88:     learn: 0.2374762        total: 353ms    remaining: 3.62s
```

```
89:     learn: 0.2359762        total: 361ms    remaining: 3.65s
90:     learn: 0.2347028        total: 364ms    remaining: 3.63s
91:     learn: 0.2332876        total: 367ms    remaining: 3.62s
92:     learn: 0.2319448        total: 370ms    remaining: 3.61s
93:     learn: 0.2305904        total: 373ms    remaining: 3.6s
94:     learn: 0.2294345        total: 376ms    remaining: 3.58s
95:     learn: 0.2281237        total: 378ms    remaining: 3.56s
96:     learn: 0.2266354        total: 382ms    remaining: 3.55s
97:     learn: 0.2250822        total: 385ms    remaining: 3.54s
98:     learn: 0.2239951        total: 389ms    remaining: 3.54s
99:     learn: 0.2226589        total: 393ms    remaining: 3.54s
100:    learn: 0.2214223        total: 397ms    remaining: 3.53s
101:    learn: 0.2200210        total: 401ms    remaining: 3.53s
102:    learn: 0.2189365        total: 404ms    remaining: 3.52s
103:    learn: 0.2175407        total: 408ms    remaining: 3.51s
104:    learn: 0.2161273        total: 414ms    remaining: 3.53s
105:    learn: 0.2149155        total: 417ms    remaining: 3.52s
106:    learn: 0.2139814        total: 419ms    remaining: 3.5s
107:    learn: 0.2128662        total: 425ms    remaining: 3.51s
108:    learn: 0.2118583        total: 429ms    remaining: 3.5s
109:    learn: 0.2107315        total: 431ms    remaining: 3.48s
110:    learn: 0.2094628        total: 435ms    remaining: 3.48s
111:    learn: 0.2085279        total: 439ms    remaining: 3.48s
112:    learn: 0.2075109        total: 442ms    remaining: 3.47s
113:    learn: 0.2065916        total: 447ms    remaining: 3.47s
114:    learn: 0.2056701        total: 451ms    remaining: 3.47s
115:    learn: 0.2046972        total: 454ms    remaining: 3.46s
116:    learn: 0.2039639        total: 461ms    remaining: 3.48s
117:    learn: 0.2033052        total: 462ms    remaining: 3.45s
118:    learn: 0.2025221        total: 464ms    remaining: 3.44s
119:    learn: 0.2015343        total: 470ms    remaining: 3.45s
120:    learn: 0.2007947        total: 473ms    remaining: 3.44s
121:    learn: 0.1999661        total: 479ms    remaining: 3.44s
122:    learn: 0.1989687        total: 482ms    remaining: 3.44s
123:    learn: 0.1982414        total: 485ms    remaining: 3.43s
124:    learn: 0.1974401        total: 488ms    remaining: 3.41s
125:    learn: 0.1966092        total: 492ms    remaining: 3.41s
126:    learn: 0.1959633        total: 495ms    remaining: 3.4s
127:    learn: 0.1951788        total: 499ms    remaining: 3.4s
128:    learn: 0.1942235        total: 501ms    remaining: 3.38s
129:    learn: 0.1935134        total: 505ms    remaining: 3.38s
130:    learn: 0.1929415        total: 507ms    remaining: 3.36s
131:    learn: 0.1922758        total: 510ms    remaining: 3.35s
132:    learn: 0.1916552        total: 513ms    remaining: 3.34s
133:    learn: 0.1908043        total: 516ms    remaining: 3.34s
134:    learn: 0.1902038        total: 520ms    remaining: 3.33s
135:    learn: 0.1892662        total: 526ms    remaining: 3.34s
136:    learn: 0.1884593        total: 529ms    remaining: 3.33s
137:    learn: 0.1877956        total: 535ms    remaining: 3.34s
138:    learn: 0.1870694        total: 537ms    remaining: 3.33s
139:    learn: 0.1865338        total: 542ms    remaining: 3.33s
140:    learn: 0.1859862        total: 544ms    remaining: 3.31s
141:    learn: 0.1853954        total: 547ms    remaining: 3.31s
142:    learn: 0.1849194        total: 551ms    remaining: 3.3s
143:    learn: 0.1843259        total: 554ms    remaining: 3.29s
144:    learn: 0.1835474        total: 557ms    remaining: 3.28s
145:    learn: 0.1829124        total: 560ms    remaining: 3.28s
146:    learn: 0.1822582        total: 564ms    remaining: 3.27s
147:    learn: 0.1816811        total: 567ms    remaining: 3.26s
148:    learn: 0.1811275        total: 570ms    remaining: 3.25s
149:    learn: 0.1805116        total: 573ms    remaining: 3.25s
150:    learn: 0.1799954        total: 576ms    remaining: 3.24s
151:    learn: 0.1795014        total: 579ms    remaining: 3.23s
152:    learn: 0.1788112        total: 583ms    remaining: 3.23s
153:    learn: 0.1783252        total: 586ms    remaining: 3.22s
154:    learn: 0.1777379        total: 589ms    remaining: 3.21s
155:    learn: 0.1771684        total: 592ms    remaining: 3.2s
156:    learn: 0.1766813        total: 596ms    remaining: 3.2s
157:    learn: 0.1761171        total: 599ms    remaining: 3.19s
158:    learn: 0.1756723        total: 601ms    remaining: 3.18s
159:    learn: 0.1752463        total: 605ms    remaining: 3.18s
160:    learn: 0.1746603        total: 609ms    remaining: 3.17s
161:    learn: 0.1741193        total: 612ms    remaining: 3.16s
162:    learn: 0.1736628        total: 615ms    remaining: 3.16s
163:    learn: 0.1733006        total: 618ms    remaining: 3.15s
164:    learn: 0.1729352        total: 622ms    remaining: 3.15s
165:    learn: 0.1725203        total: 626ms    remaining: 3.14s
166:    learn: 0.1719620        total: 630ms    remaining: 3.14s
167:    learn: 0.1713884        total: 633ms    remaining: 3.13s
168:    learn: 0.1710637        total: 636ms    remaining: 3.13s
169:    learn: 0.1707124        total: 640ms    remaining: 3.12s
170:    learn: 0.1702127        total: 647ms    remaining: 3.13s
171:    learn: 0.1697025        total: 649ms    remaining: 3.12s
172:    learn: 0.1692030        total: 652ms    remaining: 3.12s
173:    learn: 0.1688561        total: 656ms    remaining: 3.11s
174:    learn: 0.1685376        total: 659ms    remaining: 3.1s
175:    learn: 0.1683550        total: 661ms    remaining: 3.1s
176:    learn: 0.1681483        total: 663ms    remaining: 3.08s
177:    learn: 0.1678474        total: 665ms    remaining: 3.07s
178:    learn: 0.1675846        total: 666ms    remaining: 3.06s
```

```
179:	learn: 0.1672462	total: 668ms	remaining: 3.04s
180:	learn: 0.1667759	total: 670ms	remaining: 3.03s
181:	learn: 0.1665186	total: 673ms	remaining: 3.03s
182:	learn: 0.1662279	total: 676ms	remaining: 3.02s
183:	learn: 0.1658602	total: 679ms	remaining: 3.01s
184:	learn: 0.1655072	total: 682ms	remaining: 3s
185:	learn: 0.1651806	total: 685ms	remaining: 3s
186:	learn: 0.1648855	total: 692ms	remaining: 3.01s
187:	learn: 0.1645204	total: 694ms	remaining: 3s
188:	learn: 0.1641847	total: 697ms	remaining: 2.99s
189:	learn: 0.1639681	total: 701ms	remaining: 2.99s
190:	learn: 0.1637088	total: 704ms	remaining: 2.98s
191:	learn: 0.1633578	total: 707ms	remaining: 2.98s
192:	learn: 0.1629611	total: 711ms	remaining: 2.97s
193:	learn: 0.1623889	total: 714ms	remaining: 2.97s
194:	learn: 0.1619598	total: 717ms	remaining: 2.96s
195:	learn: 0.1615636	total: 720ms	remaining: 2.95s
196:	learn: 0.1611112	total: 724ms	remaining: 2.95s
197:	learn: 0.1607277	total: 727ms	remaining: 2.94s
198:	learn: 0.1604199	total: 731ms	remaining: 2.94s
199:	learn: 0.1601319	total: 734ms	remaining: 2.94s
200:	learn: 0.1598774	total: 738ms	remaining: 2.93s
201:	learn: 0.1595689	total: 741ms	remaining: 2.93s
202:	learn: 0.1593459	total: 744ms	remaining: 2.92s
203:	learn: 0.1590868	total: 747ms	remaining: 2.92s
204:	learn: 0.1588434	total: 751ms	remaining: 2.91s
205:	learn: 0.1585276	total: 754ms	remaining: 2.9s
206:	learn: 0.1582945	total: 757ms	remaining: 2.9s
207:	learn: 0.1581622	total: 760ms	remaining: 2.89s
208:	learn: 0.1578834	total: 762ms	remaining: 2.88s
209:	learn: 0.1576706	total: 766ms	remaining: 2.88s
210:	learn: 0.1573729	total: 770ms	remaining: 2.88s
211:	learn: 0.1572188	total: 773ms	remaining: 2.87s
212:	learn: 0.1569634	total: 777ms	remaining: 2.87s
213:	learn: 0.1567705	total: 780ms	remaining: 2.87s
214:	learn: 0.1566407	total: 783ms	remaining: 2.86s
215:	learn: 0.1565273	total: 785ms	remaining: 2.85s
216:	learn: 0.1563581	total: 787ms	remaining: 2.84s
217:	learn: 0.1561157	total: 791ms	remaining: 2.83s
218:	learn: 0.1558575	total: 794ms	remaining: 2.83s
219:	learn: 0.1556132	total: 797ms	remaining: 2.83s
220:	learn: 0.1553920	total: 800ms	remaining: 2.82s
221:	learn: 0.1550516	total: 803ms	remaining: 2.81s
222:	learn: 0.1548535	total: 806ms	remaining: 2.81s
223:	learn: 0.1546204	total: 810ms	remaining: 2.81s
224:	learn: 0.1544993	total: 813ms	remaining: 2.8s
225:	learn: 0.1542518	total: 816ms	remaining: 2.79s
226:	learn: 0.1539267	total: 820ms	remaining: 2.79s
227:	learn: 0.1537461	total: 823ms	remaining: 2.79s
228:	learn: 0.1534009	total: 827ms	remaining: 2.78s
229:	learn: 0.1531186	total: 830ms	remaining: 2.78s
230:	learn: 0.1528641	total: 833ms	remaining: 2.77s
231:	learn: 0.1526974	total: 837ms	remaining: 2.77s
232:	learn: 0.1525149	total: 840ms	remaining: 2.77s
233:	learn: 0.1524267	total: 842ms	remaining: 2.76s
234:	learn: 0.1522085	total: 845ms	remaining: 2.75s
235:	learn: 0.1521030	total: 848ms	remaining: 2.75s
236:	learn: 0.1516701	total: 851ms	remaining: 2.74s
237:	learn: 0.1514345	total: 854ms	remaining: 2.73s
238:	learn: 0.1512365	total: 857ms	remaining: 2.73s
239:	learn: 0.1510325	total: 862ms	remaining: 2.73s
240:	learn: 0.1508607	total: 866ms	remaining: 2.73s
241:	learn: 0.1507172	total: 869ms	remaining: 2.72s
242:	learn: 0.1505563	total: 873ms	remaining: 2.72s
243:	learn: 0.1503907	total: 876ms	remaining: 2.71s
244:	learn: 0.1501381	total: 879ms	remaining: 2.71s
245:	learn: 0.1498321	total: 883ms	remaining: 2.71s
246:	learn: 0.1495206	total: 886ms	remaining: 2.7s
247:	learn: 0.1492991	total: 890ms	remaining: 2.7s
248:	learn: 0.1491594	total: 893ms	remaining: 2.69s
249:	learn: 0.1490089	total: 896ms	remaining: 2.69s
250:	learn: 0.1487319	total: 899ms	remaining: 2.68s
251:	learn: 0.1484973	total: 903ms	remaining: 2.68s
252:	learn: 0.1483977	total: 905ms	remaining: 2.67s
253:	learn: 0.1482292	total: 908ms	remaining: 2.67s
254:	learn: 0.1480126	total: 911ms	remaining: 2.66s
255:	learn: 0.1478626	total: 914ms	remaining: 2.66s
256:	learn: 0.1477198	total: 918ms	remaining: 2.65s
257:	learn: 0.1474674	total: 921ms	remaining: 2.65s
258:	learn: 0.1471920	total: 925ms	remaining: 2.65s
259:	learn: 0.1470173	total: 929ms	remaining: 2.64s
260:	learn: 0.1468250	total: 933ms	remaining: 2.64s
261:	learn: 0.1466898	total: 936ms	remaining: 2.64s
262:	learn: 0.1465903	total: 939ms	remaining: 2.63s
263:	learn: 0.1463974	total: 943ms	remaining: 2.63s
264:	learn: 0.1462989	total: 946ms	remaining: 2.62s
265:	learn: 0.1460534	total: 949ms	remaining: 2.62s
266:	learn: 0.1458876	total: 952ms	remaining: 2.61s
267:	learn: 0.1457815	total: 956ms	remaining: 2.61s
268:	learn: 0.1455856	total: 959ms	remaining: 2.6s
269:	learn: 0.1455316	total: 962ms	remaining: 2.6s
```

```
269:    learn: 0.1455510        total: 962ms    remaining: 2.6s
270:    learn: 0.1454705        total: 965ms    remaining: 2.6s
271:    learn: 0.1451447        total: 968ms    remaining: 2.59s
272:    learn: 0.1448626        total: 972ms    remaining: 2.59s
273:    learn: 0.1447586        total: 975ms    remaining: 2.58s
274:    learn: 0.1445739        total: 978ms    remaining: 2.58s
275:    learn: 0.1444668        total: 982ms    remaining: 2.57s
276:    learn: 0.1442834        total: 985ms    remaining: 2.57s
277:    learn: 0.1439650        total: 988ms    remaining: 2.57s
278:    learn: 0.1439381        total: 991ms    remaining: 2.56s
279:    learn: 0.1437235        total: 993ms    remaining: 2.55s
280:    learn: 0.1434872        total: 1000ms   remaining: 2.56s
281:    learn: 0.1432785        total: 1.01s    remaining: 2.56s
282:    learn: 0.1430194        total: 1.01s    remaining: 2.56s
283:    learn: 0.1429742        total: 1.01s    remaining: 2.56s
284:    learn: 0.1427949        total: 1.02s    remaining: 2.55s
285:    learn: 0.1427056        total: 1.02s    remaining: 2.55s
286:    learn: 0.1425595        total: 1.03s    remaining: 2.55s
287:    learn: 0.1424272        total: 1.03s    remaining: 2.55s
288:    learn: 0.1422969        total: 1.04s    remaining: 2.55s
289:    learn: 0.1421977        total: 1.04s    remaining: 2.55s
290:    learn: 0.1420189        total: 1.04s    remaining: 2.54s
291:    learn: 0.1418577        total: 1.04s    remaining: 2.53s
292:    learn: 0.1416463        total: 1.05s    remaining: 2.53s
293:    learn: 0.1414894        total: 1.05s    remaining: 2.52s
294:    learn: 0.1413785        total: 1.05s    remaining: 2.52s
295:    learn: 0.1412923        total: 1.06s    remaining: 2.52s
296:    learn: 0.1409532        total: 1.06s    remaining: 2.51s
297:    learn: 0.1407998        total: 1.06s    remaining: 2.51s
298:    learn: 0.1405858        total: 1.07s    remaining: 2.5s
299:    learn: 0.1404856        total: 1.07s    remaining: 2.5s
300:    learn: 0.1402029        total: 1.07s    remaining: 2.49s
301:    learn: 0.1399654        total: 1.08s    remaining: 2.49s
302:    learn: 0.1398020        total: 1.08s    remaining: 2.48s
303:    learn: 0.1396515        total: 1.08s    remaining: 2.48s
304:    learn: 0.1395054        total: 1.08s    remaining: 2.47s
305:    learn: 0.1392857        total: 1.08s    remaining: 2.46s
306:    learn: 0.1391551        total: 1.09s    remaining: 2.46s
307:    learn: 0.1389194        total: 1.09s    remaining: 2.45s
308:    learn: 0.1388771        total: 1.09s    remaining: 2.44s
309:    learn: 0.1387439        total: 1.09s    remaining: 2.43s
310:    learn: 0.1386541        total: 1.09s    remaining: 2.42s
311:    learn: 0.1385175        total: 1.09s    remaining: 2.41s
312:    learn: 0.1383489        total: 1.1s     remaining: 2.41s
313:    learn: 0.1382379        total: 1.1s     remaining: 2.4s
314:    learn: 0.1381610        total: 1.1s     remaining: 2.39s
315:    learn: 0.1380360        total: 1.1s     remaining: 2.38s
316:    learn: 0.1378613        total: 1.1s     remaining: 2.38s
317:    learn: 0.1377124        total: 1.1s     remaining: 2.37s
318:    learn: 0.1376728        total: 1.1s     remaining: 2.36s
319:    learn: 0.1374618        total: 1.11s    remaining: 2.35s
320:    learn: 0.1373110        total: 1.11s    remaining: 2.34s
321:    learn: 0.1372869        total: 1.11s    remaining: 2.33s
322:    learn: 0.1370690        total: 1.11s    remaining: 2.33s
323:    learn: 0.1369889        total: 1.11s    remaining: 2.32s
324:    learn: 0.1368477        total: 1.11s    remaining: 2.31s
325:    learn: 0.1367503        total: 1.11s    remaining: 2.31s
326:    learn: 0.1366218        total: 1.12s    remaining: 2.3s
327:    learn: 0.1365272        total: 1.12s    remaining: 2.29s
328:    learn: 0.1362821        total: 1.12s    remaining: 2.28s
329:    learn: 0.1361522        total: 1.12s    remaining: 2.28s
330:    learn: 0.1359606        total: 1.12s    remaining: 2.27s
331:    learn: 0.1358953        total: 1.12s    remaining: 2.26s
332:    learn: 0.1358079        total: 1.13s    remaining: 2.26s
333:    learn: 0.1357632        total: 1.13s    remaining: 2.25s
334:    learn: 0.1357026        total: 1.13s    remaining: 2.24s
335:    learn: 0.1354942        total: 1.13s    remaining: 2.23s
336:    learn: 0.1353266        total: 1.13s    remaining: 2.23s
337:    learn: 0.1352528        total: 1.13s    remaining: 2.22s
338:    learn: 0.1351993        total: 1.14s    remaining: 2.21s
339:    learn: 0.1350123        total: 1.14s    remaining: 2.21s
340:    learn: 0.1348457        total: 1.14s    remaining: 2.2s
341:    learn: 0.1346764        total: 1.14s    remaining: 2.19s
342:    learn: 0.1344904        total: 1.14s    remaining: 2.19s
343:    learn: 0.1344099        total: 1.14s    remaining: 2.18s
344:    learn: 0.1342649        total: 1.15s    remaining: 2.17s
345:    learn: 0.1342122        total: 1.15s    remaining: 2.17s
346:    learn: 0.1340741        total: 1.15s    remaining: 2.16s
347:    learn: 0.1339902        total: 1.15s    remaining: 2.15s
348:    learn: 0.1339228        total: 1.15s    remaining: 2.15s
349:    learn: 0.1338306        total: 1.15s    remaining: 2.14s
350:    learn: 0.1336808        total: 1.15s    remaining: 2.13s
351:    learn: 0.1335970        total: 1.16s    remaining: 2.13s
352:    learn: 0.1332814        total: 1.16s    remaining: 2.12s
353:    learn: 0.1331411        total: 1.16s    remaining: 2.12s
354:    learn: 0.1330239        total: 1.16s    remaining: 2.12s
355:    learn: 0.1329856        total: 1.17s    remaining: 2.11s
356:    learn: 0.1328486        total: 1.17s    remaining: 2.11s
357:    learn: 0.1326897        total: 1.17s    remaining: 2.1s
358:    learn: 0.1326421        total: 1.18s    remaining: 2.1s
359:    learn: 0.1325297        total: 1.18s    remaining: 2.1s
```

```
360:	learn: 0.1324456	total: 1.18s	remaining: 2.09s
361:	learn: 0.1323531	total: 1.19s	remaining: 2.09s
362:	learn: 0.1322892	total: 1.19s	remaining: 2.09s
363:	learn: 0.1322288	total: 1.19s	remaining: 2.09s
364:	learn: 0.1321716	total: 1.2s	remaining: 2.08s
365:	learn: 0.1320154	total: 1.2s	remaining: 2.08s
366:	learn: 0.1318852	total: 1.2s	remaining: 2.08s
367:	learn: 0.1318204	total: 1.21s	remaining: 2.08s
368:	learn: 0.1317947	total: 1.21s	remaining: 2.07s
369:	learn: 0.1316186	total: 1.22s	remaining: 2.07s
370:	learn: 0.1315121	total: 1.22s	remaining: 2.07s
371:	learn: 0.1314388	total: 1.22s	remaining: 2.06s
372:	learn: 0.1313732	total: 1.23s	remaining: 2.06s
373:	learn: 0.1311862	total: 1.23s	remaining: 2.06s
374:	learn: 0.1309184	total: 1.23s	remaining: 2.05s
375:	learn: 0.1306457	total: 1.24s	remaining: 2.05s
376:	learn: 0.1305920	total: 1.24s	remaining: 2.05s
377:	learn: 0.1304248	total: 1.24s	remaining: 2.04s
378:	learn: 0.1302877	total: 1.25s	remaining: 2.05s
379:	learn: 0.1301617	total: 1.25s	remaining: 2.05s
380:	learn: 0.1301295	total: 1.26s	remaining: 2.05s
381:	learn: 0.1299797	total: 1.26s	remaining: 2.04s
382:	learn: 0.1296996	total: 1.27s	remaining: 2.04s
383:	learn: 0.1296506	total: 1.27s	remaining: 2.04s
384:	learn: 0.1295863	total: 1.27s	remaining: 2.04s
385:	learn: 0.1295438	total: 1.28s	remaining: 2.03s
386:	learn: 0.1293666	total: 1.28s	remaining: 2.03s
387:	learn: 0.1292838	total: 1.28s	remaining: 2.03s
388:	learn: 0.1290797	total: 1.29s	remaining: 2.02s
389:	learn: 0.1290156	total: 1.29s	remaining: 2.02s
390:	learn: 0.1289237	total: 1.3s	remaining: 2.02s
391:	learn: 0.1287982	total: 1.3s	remaining: 2.02s
392:	learn: 0.1286104	total: 1.3s	remaining: 2.01s
393:	learn: 0.1284510	total: 1.31s	remaining: 2.01s
394:	learn: 0.1283675	total: 1.31s	remaining: 2.01s
395:	learn: 0.1282911	total: 1.31s	remaining: 2s
396:	learn: 0.1280920	total: 1.32s	remaining: 2s
397:	learn: 0.1278934	total: 1.32s	remaining: 2s
398:	learn: 0.1277971	total: 1.32s	remaining: 2s
399:	learn: 0.1277042	total: 1.33s	remaining: 1.99s
400:	learn: 0.1275989	total: 1.33s	remaining: 1.99s
401:	learn: 0.1275604	total: 1.34s	remaining: 1.99s
402:	learn: 0.1274405	total: 1.34s	remaining: 1.98s
403:	learn: 0.1272776	total: 1.34s	remaining: 1.98s
404:	learn: 0.1272056	total: 1.35s	remaining: 1.98s
405:	learn: 0.1271355	total: 1.35s	remaining: 1.98s
406:	learn: 0.1269730	total: 1.35s	remaining: 1.98s
407:	learn: 0.1269292	total: 1.36s	remaining: 1.97s
408:	learn: 0.1268036	total: 1.36s	remaining: 1.96s
409:	learn: 0.1266840	total: 1.36s	remaining: 1.96s
410:	learn: 0.1265806	total: 1.36s	remaining: 1.95s
411:	learn: 0.1264323	total: 1.36s	remaining: 1.95s
412:	learn: 0.1262675	total: 1.37s	remaining: 1.94s
413:	learn: 0.1261255	total: 1.37s	remaining: 1.94s
414:	learn: 0.1259447	total: 1.37s	remaining: 1.93s
415:	learn: 0.1257280	total: 1.37s	remaining: 1.93s
416:	learn: 0.1255341	total: 1.37s	remaining: 1.92s
417:	learn: 0.1254041	total: 1.37s	remaining: 1.91s
418:	learn: 0.1252243	total: 1.38s	remaining: 1.91s
419:	learn: 0.1251176	total: 1.38s	remaining: 1.9s
420:	learn: 0.1250497	total: 1.38s	remaining: 1.9s
421:	learn: 0.1248855	total: 1.38s	remaining: 1.89s
422:	learn: 0.1247459	total: 1.38s	remaining: 1.88s
423:	learn: 0.1246411	total: 1.38s	remaining: 1.88s
424:	learn: 0.1244748	total: 1.38s	remaining: 1.87s
425:	learn: 0.1242802	total: 1.39s	remaining: 1.87s
426:	learn: 0.1241403	total: 1.39s	remaining: 1.86s
427:	learn: 0.1241170	total: 1.39s	remaining: 1.85s
428:	learn: 0.1240127	total: 1.39s	remaining: 1.85s
429:	learn: 0.1239458	total: 1.39s	remaining: 1.84s
430:	learn: 0.1238526	total: 1.39s	remaining: 1.84s
431:	learn: 0.1238038	total: 1.4s	remaining: 1.83s
432:	learn: 0.1236929	total: 1.4s	remaining: 1.83s
433:	learn: 0.1236439	total: 1.4s	remaining: 1.82s
434:	learn: 0.1234607	total: 1.4s	remaining: 1.82s
435:	learn: 0.1233452	total: 1.4s	remaining: 1.81s
436:	learn: 0.1232377	total: 1.4s	remaining: 1.81s
437:	learn: 0.1230883	total: 1.4s	remaining: 1.8s
438:	learn: 0.1228722	total: 1.41s	remaining: 1.8s
439:	learn: 0.1227638	total: 1.41s	remaining: 1.79s
440:	learn: 0.1226285	total: 1.41s	remaining: 1.78s
441:	learn: 0.1225367	total: 1.41s	remaining: 1.78s
442:	learn: 0.1224541	total: 1.41s	remaining: 1.77s
443:	learn: 0.1223797	total: 1.41s	remaining: 1.77s
444:	learn: 0.1222827	total: 1.42s	remaining: 1.76s
445:	learn: 0.1221761	total: 1.42s	remaining: 1.76s
446:	learn: 0.1220352	total: 1.42s	remaining: 1.75s
447:	learn: 0.1219843	total: 1.42s	remaining: 1.75s
448:	learn: 0.1218706	total: 1.42s	remaining: 1.74s
449:	learn: 0.1216723	total: 1.42s	remaining: 1.74s
```

```
450:	learn: 0.1215648	total: 1.43s	remaining: 1.74s
451:	learn: 0.1215351	total: 1.43s	remaining: 1.73s
452:	learn: 0.1213556	total: 1.43s	remaining: 1.73s
453:	learn: 0.1212521	total: 1.44s	remaining: 1.73s
454:	learn: 0.1211838	total: 1.44s	remaining: 1.72s
455:	learn: 0.1209797	total: 1.44s	remaining: 1.72s
456:	learn: 0.1209231	total: 1.45s	remaining: 1.72s
457:	learn: 0.1208120	total: 1.45s	remaining: 1.72s
458:	learn: 0.1207379	total: 1.45s	remaining: 1.71s
459:	learn: 0.1206349	total: 1.45s	remaining: 1.71s
460:	learn: 0.1205324	total: 1.46s	remaining: 1.7s
461:	learn: 0.1203921	total: 1.46s	remaining: 1.7s
462:	learn: 0.1202600	total: 1.46s	remaining: 1.7s
463:	learn: 0.1199869	total: 1.46s	remaining: 1.69s
464:	learn: 0.1198946	total: 1.47s	remaining: 1.69s
465:	learn: 0.1197957	total: 1.47s	remaining: 1.68s
466:	learn: 0.1196632	total: 1.47s	remaining: 1.68s
467:	learn: 0.1194269	total: 1.47s	remaining: 1.67s
468:	learn: 0.1193705	total: 1.47s	remaining: 1.67s
469:	learn: 0.1191321	total: 1.47s	remaining: 1.66s
470:	learn: 0.1189975	total: 1.48s	remaining: 1.66s
471:	learn: 0.1189272	total: 1.48s	remaining: 1.65s
472:	learn: 0.1187448	total: 1.48s	remaining: 1.65s
473:	learn: 0.1187369	total: 1.48s	remaining: 1.64s
474:	learn: 0.1187310	total: 1.48s	remaining: 1.64s
475:	learn: 0.1186764	total: 1.48s	remaining: 1.63s
476:	learn: 0.1185418	total: 1.48s	remaining: 1.63s
477:	learn: 0.1183083	total: 1.49s	remaining: 1.62s
478:	learn: 0.1182452	total: 1.49s	remaining: 1.62s
479:	learn: 0.1181870	total: 1.49s	remaining: 1.61s
480:	learn: 0.1180811	total: 1.49s	remaining: 1.61s
481:	learn: 0.1180752	total: 1.49s	remaining: 1.6s
482:	learn: 0.1180304	total: 1.49s	remaining: 1.6s
483:	learn: 0.1178854	total: 1.5s	remaining: 1.59s
484:	learn: 0.1177904	total: 1.5s	remaining: 1.59s
485:	learn: 0.1176922	total: 1.5s	remaining: 1.58s
486:	learn: 0.1175398	total: 1.5s	remaining: 1.58s
487:	learn: 0.1175194	total: 1.5s	remaining: 1.57s
488:	learn: 0.1173671	total: 1.5s	remaining: 1.57s
489:	learn: 0.1172551	total: 1.5s	remaining: 1.57s
490:	learn: 0.1171811	total: 1.51s	remaining: 1.56s
491:	learn: 0.1170463	total: 1.51s	remaining: 1.56s
492:	learn: 0.1169169	total: 1.51s	remaining: 1.55s
493:	learn: 0.1167027	total: 1.51s	remaining: 1.55s
494:	learn: 0.1165319	total: 1.52s	remaining: 1.55s
495:	learn: 0.1164589	total: 1.52s	remaining: 1.54s
496:	learn: 0.1163964	total: 1.52s	remaining: 1.54s
497:	learn: 0.1162667	total: 1.52s	remaining: 1.53s
498:	learn: 0.1160647	total: 1.52s	remaining: 1.53s
499:	learn: 0.1160024	total: 1.52s	remaining: 1.52s
500:	learn: 0.1158768	total: 1.52s	remaining: 1.52s
501:	learn: 0.1157359	total: 1.53s	remaining: 1.51s
502:	learn: 0.1154662	total: 1.53s	remaining: 1.51s
503:	learn: 0.1153697	total: 1.53s	remaining: 1.5s
504:	learn: 0.1153077	total: 1.53s	remaining: 1.5s
505:	learn: 0.1152691	total: 1.53s	remaining: 1.5s
506:	learn: 0.1152630	total: 1.53s	remaining: 1.49s
507:	learn: 0.1151548	total: 1.53s	remaining: 1.49s
508:	learn: 0.1150512	total: 1.54s	remaining: 1.48s
509:	learn: 0.1149178	total: 1.54s	remaining: 1.48s
510:	learn: 0.1147788	total: 1.54s	remaining: 1.47s
511:	learn: 0.1147145	total: 1.54s	remaining: 1.47s
512:	learn: 0.1146059	total: 1.54s	remaining: 1.46s
513:	learn: 0.1143656	total: 1.54s	remaining: 1.46s
514:	learn: 0.1141737	total: 1.55s	remaining: 1.46s
515:	learn: 0.1141293	total: 1.55s	remaining: 1.45s
516:	learn: 0.1140307	total: 1.55s	remaining: 1.45s
517:	learn: 0.1138678	total: 1.55s	remaining: 1.44s
518:	learn: 0.1137246	total: 1.55s	remaining: 1.44s
519:	learn: 0.1136322	total: 1.55s	remaining: 1.43s
520:	learn: 0.1134440	total: 1.55s	remaining: 1.43s
521:	learn: 0.1133498	total: 1.55s	remaining: 1.42s
522:	learn: 0.1132459	total: 1.56s	remaining: 1.42s
523:	learn: 0.1131367	total: 1.56s	remaining: 1.42s
524:	learn: 0.1129397	total: 1.56s	remaining: 1.41s
525:	learn: 0.1128210	total: 1.56s	remaining: 1.41s
526:	learn: 0.1126896	total: 1.57s	remaining: 1.41s
527:	learn: 0.1126402	total: 1.57s	remaining: 1.4s
528:	learn: 0.1123927	total: 1.57s	remaining: 1.4s
529:	learn: 0.1122566	total: 1.57s	remaining: 1.39s
530:	learn: 0.1121825	total: 1.57s	remaining: 1.39s
531:	learn: 0.1121474	total: 1.57s	remaining: 1.38s
532:	learn: 0.1119896	total: 1.57s	remaining: 1.38s
533:	learn: 0.1119229	total: 1.58s	remaining: 1.38s
534:	learn: 0.1118489	total: 1.58s	remaining: 1.37s
535:	learn: 0.1117289	total: 1.58s	remaining: 1.37s
536:	learn: 0.1115837	total: 1.58s	remaining: 1.36s
537:	learn: 0.1115215	total: 1.58s	remaining: 1.36s
538:	learn: 0.1114475	total: 1.58s	remaining: 1.35s
539:	learn: 0.1113848	total: 1.59s	remaining: 1.35s
540:	learn: 0.1113293	total: 1.59s	remaining: 1.35s
```

```
540:    learn: 0.1113293        total: 1.59s    remaining: 1.35s
541:    learn: 0.1112446        total: 1.59s    remaining: 1.34s
542:    learn: 0.1112228        total: 1.59s    remaining: 1.34s
543:    learn: 0.1111887        total: 1.59s    remaining: 1.33s
544:    learn: 0.1110781        total: 1.59s    remaining: 1.33s
545:    learn: 0.1109783        total: 1.59s    remaining: 1.33s
546:    learn: 0.1108673        total: 1.6s     remaining: 1.32s
547:    learn: 0.1107572        total: 1.6s     remaining: 1.32s
548:    learn: 0.1106188        total: 1.6s     remaining: 1.31s
549:    learn: 0.1105509        total: 1.6s     remaining: 1.31s
550:    learn: 0.1104772        total: 1.61s    remaining: 1.31s
551:    learn: 0.1103676        total: 1.61s    remaining: 1.3s
552:    learn: 0.1101977        total: 1.61s    remaining: 1.3s
553:    learn: 0.1100088        total: 1.61s    remaining: 1.3s
554:    learn: 0.1099245        total: 1.61s    remaining: 1.29s
555:    learn: 0.1097303        total: 1.61s    remaining: 1.29s
556:    learn: 0.1096126        total: 1.62s    remaining: 1.29s
557:    learn: 0.1095050        total: 1.62s    remaining: 1.28s
558:    learn: 0.1094444        total: 1.62s    remaining: 1.28s
559:    learn: 0.1093324        total: 1.62s    remaining: 1.27s
560:    learn: 0.1092194        total: 1.62s    remaining: 1.27s
561:    learn: 0.1090561        total: 1.62s    remaining: 1.26s
562:    learn: 0.1088816        total: 1.63s    remaining: 1.26s
563:    learn: 0.1087166        total: 1.63s    remaining: 1.26s
564:    learn: 0.1085866        total: 1.63s    remaining: 1.25s
565:    learn: 0.1085435        total: 1.63s    remaining: 1.25s
566:    learn: 0.1084840        total: 1.63s    remaining: 1.25s
567:    learn: 0.1084428        total: 1.63s    remaining: 1.24s
568:    learn: 0.1083144        total: 1.64s    remaining: 1.24s
569:    learn: 0.1081753        total: 1.64s    remaining: 1.24s
570:    learn: 0.1080834        total: 1.64s    remaining: 1.23s
571:    learn: 0.1080162        total: 1.64s    remaining: 1.23s
572:    learn: 0.1078693        total: 1.64s    remaining: 1.22s
573:    learn: 0.1077951        total: 1.64s    remaining: 1.22s
574:    learn: 0.1077210        total: 1.65s    remaining: 1.22s
575:    learn: 0.1076356        total: 1.65s    remaining: 1.21s
576:    learn: 0.1075786        total: 1.65s    remaining: 1.21s
577:    learn: 0.1074938        total: 1.66s    remaining: 1.21s
578:    learn: 0.1074511        total: 1.66s    remaining: 1.21s
579:    learn: 0.1073949        total: 1.67s    remaining: 1.21s
580:    learn: 0.1073147        total: 1.67s    remaining: 1.2s
581:    learn: 0.1072228        total: 1.67s    remaining: 1.2s
582:    learn: 0.1071549        total: 1.68s    remaining: 1.2s
583:    learn: 0.1071266        total: 1.68s    remaining: 1.2s
584:    learn: 0.1070436        total: 1.68s    remaining: 1.19s
585:    learn: 0.1070005        total: 1.69s    remaining: 1.19s
586:    learn: 0.1069928        total: 1.69s    remaining: 1.19s
587:    learn: 0.1068033        total: 1.69s    remaining: 1.18s
588:    learn: 0.1066780        total: 1.69s    remaining: 1.18s
589:    learn: 0.1066151        total: 1.7s     remaining: 1.18s
590:    learn: 0.1065472        total: 1.7s     remaining: 1.18s
591:    learn: 0.1064510        total: 1.7s     remaining: 1.17s
592:    learn: 0.1063435        total: 1.71s    remaining: 1.17s
593:    learn: 0.1062474        total: 1.71s    remaining: 1.17s
594:    learn: 0.1061903        total: 1.71s    remaining: 1.17s
595:    learn: 0.1060858        total: 1.72s    remaining: 1.16s
596:    learn: 0.1058901        total: 1.72s    remaining: 1.16s
597:    learn: 0.1058295        total: 1.72s    remaining: 1.16s
598:    learn: 0.1056718        total: 1.73s    remaining: 1.16s
599:    learn: 0.1056055        total: 1.73s    remaining: 1.15s
600:    learn: 0.1055948        total: 1.73s    remaining: 1.15s
601:    learn: 0.1055609        total: 1.74s    remaining: 1.15s
602:    learn: 0.1055077        total: 1.74s    remaining: 1.15s
603:    learn: 0.1054568        total: 1.75s    remaining: 1.14s
604:    learn: 0.1053827        total: 1.75s    remaining: 1.14s
605:    learn: 0.1052878        total: 1.75s    remaining: 1.14s
606:    learn: 0.1051984        total: 1.75s    remaining: 1.13s
607:    learn: 0.1051431        total: 1.75s    remaining: 1.13s
608:    learn: 0.1050871        total: 1.76s    remaining: 1.13s
609:    learn: 0.1050409        total: 1.76s    remaining: 1.13s
610:    learn: 0.1050079        total: 1.76s    remaining: 1.12s
611:    learn: 0.1049564        total: 1.77s    remaining: 1.12s
612:    learn: 0.1049194        total: 1.77s    remaining: 1.12s
613:    learn: 0.1048639        total: 1.77s    remaining: 1.11s
614:    learn: 0.1047870        total: 1.78s    remaining: 1.11s
615:    learn: 0.1046488        total: 1.78s    remaining: 1.11s
616:    learn: 0.1045799        total: 1.79s    remaining: 1.11s
617:    learn: 0.1044624        total: 1.79s    remaining: 1.11s
618:    learn: 0.1043775        total: 1.79s    remaining: 1.1s
619:    learn: 0.1043280        total: 1.8s     remaining: 1.1s
620:    learn: 0.1041398        total: 1.8s     remaining: 1.1s
621:    learn: 0.1040492        total: 1.8s     remaining: 1.1s
622:    learn: 0.1038824        total: 1.81s    remaining: 1.09s
623:    learn: 0.1038203        total: 1.81s    remaining: 1.09s
624:    learn: 0.1036996        total: 1.81s    remaining: 1.09s
625:    learn: 0.1036304        total: 1.82s    remaining: 1.09s
626:    learn: 0.1035009        total: 1.82s    remaining: 1.08s
627:    learn: 0.1034169        total: 1.83s    remaining: 1.08s
628:    learn: 0.1033590        total: 1.83s    remaining: 1.08s
629:    learn: 0.1032621        total: 1.83s    remaining: 1.08s
630:    learn: 0.1032209        total: 1.84s    remaining: 1.07s
```

```
631:    learn: 0.1031727        total: 1.84s    remaining: 1.07s
632:    learn: 0.1030643        total: 1.84s    remaining: 1.07s
633:    learn: 0.1030300        total: 1.85s    remaining: 1.06s
634:    learn: 0.1029665        total: 1.85s    remaining: 1.06s
635:    learn: 0.1028746        total: 1.85s    remaining: 1.06s
636:    learn: 0.1027437        total: 1.85s    remaining: 1.06s
637:    learn: 0.1025988        total: 1.86s    remaining: 1.05s
638:    learn: 0.1024981        total: 1.86s    remaining: 1.05s
639:    learn: 0.1024310        total: 1.87s    remaining: 1.05s
640:    learn: 0.1024234        total: 1.87s    remaining: 1.05s
641:    learn: 0.1023412        total: 1.87s    remaining: 1.04s
642:    learn: 0.1022232        total: 1.88s    remaining: 1.04s
643:    learn: 0.1021110        total: 1.88s    remaining: 1.04s
644:    learn: 0.1019151        total: 1.88s    remaining: 1.04s
645:    learn: 0.1018303        total: 1.89s    remaining: 1.03s
646:    learn: 0.1017702        total: 1.89s    remaining: 1.03s
647:    learn: 0.1016262        total: 1.89s    remaining: 1.03s
648:    learn: 0.1015872        total: 1.9s     remaining: 1.02s
649:    learn: 0.1014135        total: 1.9s     remaining: 1.02s
650:    learn: 0.1013359        total: 1.9s     remaining: 1.02s
651:    learn: 0.1013107        total: 1.9s     remaining: 1.02s
652:    learn: 0.1012072        total: 1.91s    remaining: 1.01s
653:    learn: 0.1011687        total: 1.91s    remaining: 1.01s
654:    learn: 0.1011284        total: 1.91s    remaining: 1.01s
655:    learn: 0.1009831        total: 1.92s    remaining: 1s
656:    learn: 0.1009188        total: 1.92s    remaining: 1s
657:    learn: 0.1008602        total: 1.92s    remaining: 1s
658:    learn: 0.1006648        total: 1.93s    remaining: 998ms
659:    learn: 0.1004293        total: 1.93s    remaining: 995ms
660:    learn: 0.1004020        total: 1.93s    remaining: 992ms
661:    learn: 0.1002978        total: 1.94s    remaining: 990ms
662:    learn: 0.1000739        total: 1.94s    remaining: 987ms
663:    learn: 0.0999485        total: 1.95s    remaining: 984ms
664:    learn: 0.0998055        total: 1.95s    remaining: 982ms
665:    learn: 0.0997119        total: 1.95s    remaining: 978ms
666:    learn: 0.0995660        total: 1.95s    remaining: 975ms
667:    learn: 0.0994809        total: 1.95s    remaining: 972ms
668:    learn: 0.0994254        total: 1.96s    remaining: 968ms
669:    learn: 0.0993404        total: 1.96s    remaining: 965ms
670:    learn: 0.0992508        total: 1.96s    remaining: 961ms
671:    learn: 0.0992206        total: 1.96s    remaining: 957ms
672:    learn: 0.0992007        total: 1.96s    remaining: 954ms
673:    learn: 0.0991426        total: 1.96s    remaining: 950ms
674:    learn: 0.0991082        total: 1.97s    remaining: 947ms
675:    learn: 0.0990485        total: 1.97s    remaining: 943ms
676:    learn: 0.0988952        total: 1.97s    remaining: 940ms
677:    learn: 0.0988368        total: 1.97s    remaining: 936ms
678:    learn: 0.0987273        total: 1.97s    remaining: 932ms
679:    learn: 0.0986644        total: 1.97s    remaining: 929ms
680:    learn: 0.0985469        total: 1.98s    remaining: 925ms
681:    learn: 0.0984542        total: 1.98s    remaining: 922ms
682:    learn: 0.0983746        total: 1.98s    remaining: 919ms
683:    learn: 0.0982242        total: 1.98s    remaining: 916ms
684:    learn: 0.0980581        total: 1.99s    remaining: 913ms
685:    learn: 0.0979506        total: 1.99s    remaining: 910ms
686:    learn: 0.0978947        total: 1.99s    remaining: 908ms
687:    learn: 0.0978265        total: 2s       remaining: 906ms
688:    learn: 0.0976866        total: 2s       remaining: 903ms
689:    learn: 0.0975395        total: 2s       remaining: 901ms
690:    learn: 0.0974234        total: 2.01s    remaining: 898ms
691:    learn: 0.0972996        total: 2.01s    remaining: 896ms
692:    learn: 0.0972008        total: 2.02s    remaining: 894ms
693:    learn: 0.0971666        total: 2.02s    remaining: 891ms
694:    learn: 0.0970370        total: 2.02s    remaining: 888ms
695:    learn: 0.0970236        total: 2.03s    remaining: 885ms
696:    learn: 0.0969559        total: 2.03s    remaining: 883ms
697:    learn: 0.0967952        total: 2.04s    remaining: 881ms
698:    learn: 0.0966373        total: 2.04s    remaining: 878ms
699:    learn: 0.0965316        total: 2.04s    remaining: 876ms
700:    learn: 0.0964508        total: 2.05s    remaining: 873ms
701:    learn: 0.0963082        total: 2.05s    remaining: 871ms
702:    learn: 0.0962239        total: 2.05s    remaining: 868ms
703:    learn: 0.0961683        total: 2.06s    remaining: 865ms
704:    learn: 0.0961136        total: 2.06s    remaining: 863ms
705:    learn: 0.0960062        total: 2.06s    remaining: 860ms
706:    learn: 0.0958807        total: 2.07s    remaining: 857ms
707:    learn: 0.0957802        total: 2.07s    remaining: 854ms
708:    learn: 0.0956443        total: 2.07s    remaining: 851ms
709:    learn: 0.0955663        total: 2.08s    remaining: 849ms
710:    learn: 0.0955066        total: 2.08s    remaining: 846ms
711:    learn: 0.0953597        total: 2.08s    remaining: 843ms
712:    learn: 0.0952649        total: 2.09s    remaining: 840ms
713:    learn: 0.0951931        total: 2.09s    remaining: 838ms
714:    learn: 0.0951441        total: 2.09s    remaining: 835ms
715:    learn: 0.0950697        total: 2.1s     remaining: 832ms
716:    learn: 0.0948982        total: 2.1s     remaining: 829ms
717:    learn: 0.0947306        total: 2.1s     remaining: 826ms
718:    learn: 0.0947020        total: 2.11s    remaining: 824ms
719:    learn: 0.0946953        total: 2.11s    remaining: 820ms
720:    learn: 0.0945317        total: 2.11s    remaining: 817ms
```

```
721:    learn: 0.0944328        total: 2.12s    remaining: 815ms
722:    learn: 0.0942565        total: 2.12s    remaining: 812ms
723:    learn: 0.0941533        total: 2.12s    remaining: 809ms
724:    learn: 0.0941223        total: 2.13s    remaining: 806ms
725:    learn: 0.0940875        total: 2.13s    remaining: 803ms
726:    learn: 0.0940838        total: 2.13s    remaining: 800ms
727:    learn: 0.0939383        total: 2.13s    remaining: 797ms
728:    learn: 0.0938436        total: 2.14s    remaining: 795ms
729:    learn: 0.0937799        total: 2.15s    remaining: 794ms
730:    learn: 0.0936805        total: 2.15s    remaining: 790ms
731:    learn: 0.0935080        total: 2.15s    remaining: 789ms
732:    learn: 0.0934797        total: 2.16s    remaining: 786ms
733:    learn: 0.0933783        total: 2.17s    remaining: 785ms
734:    learn: 0.0932241        total: 2.17s    remaining: 782ms
735:    learn: 0.0931378        total: 2.17s    remaining: 780ms
736:    learn: 0.0930974        total: 2.18s    remaining: 777ms
737:    learn: 0.0930350        total: 2.18s    remaining: 775ms
738:    learn: 0.0928532        total: 2.18s    remaining: 772ms
739:    learn: 0.0927580        total: 2.19s    remaining: 769ms
740:    learn: 0.0927251        total: 2.19s    remaining: 766ms
741:    learn: 0.0926062        total: 2.19s    remaining: 763ms
742:    learn: 0.0924879        total: 2.2s     remaining: 760ms
743:    learn: 0.0924270        total: 2.2s     remaining: 758ms
744:    learn: 0.0923641        total: 2.21s    remaining: 755ms
745:    learn: 0.0922446        total: 2.21s    remaining: 752ms
746:    learn: 0.0921347        total: 2.21s    remaining: 749ms
747:    learn: 0.0921104        total: 2.22s    remaining: 747ms
748:    learn: 0.0920597        total: 2.22s    remaining: 744ms
749:    learn: 0.0919547        total: 2.22s    remaining: 741ms
750:    learn: 0.0919174        total: 2.23s    remaining: 738ms
751:    learn: 0.0917980        total: 2.23s    remaining: 735ms
752:    learn: 0.0917018        total: 2.23s    remaining: 733ms
753:    learn: 0.0915761        total: 2.24s    remaining: 730ms
754:    learn: 0.0915385        total: 2.24s    remaining: 727ms
755:    learn: 0.0914984        total: 2.25s    remaining: 725ms
756:    learn: 0.0914112        total: 2.25s    remaining: 722ms
757:    learn: 0.0913000        total: 2.25s    remaining: 719ms
758:    learn: 0.0911454        total: 2.26s    remaining: 716ms
759:    learn: 0.0910794        total: 2.26s    remaining: 714ms
760:    learn: 0.0909975        total: 2.26s    remaining: 711ms
761:    learn: 0.0909423        total: 2.27s    remaining: 708ms
762:    learn: 0.0908696        total: 2.27s    remaining: 705ms
763:    learn: 0.0908420        total: 2.27s    remaining: 702ms
764:    learn: 0.0906649        total: 2.28s    remaining: 700ms
765:    learn: 0.0905942        total: 2.28s    remaining: 696ms
766:    learn: 0.0905170        total: 2.28s    remaining: 693ms
767:    learn: 0.0904925        total: 2.29s    remaining: 690ms
768:    learn: 0.0904603        total: 2.29s    remaining: 687ms
769:    learn: 0.0903081        total: 2.29s    remaining: 685ms
770:    learn: 0.0901992        total: 2.29s    remaining: 682ms
771:    learn: 0.0901195        total: 2.3s     remaining: 679ms
772:    learn: 0.0899324        total: 2.3s     remaining: 676ms
773:    learn: 0.0898687        total: 2.31s    remaining: 673ms
774:    learn: 0.0897830        total: 2.31s    remaining: 671ms
775:    learn: 0.0897028        total: 2.31s    remaining: 668ms
776:    learn: 0.0896393        total: 2.32s    remaining: 665ms
777:    learn: 0.0896041        total: 2.32s    remaining: 661ms
778:    learn: 0.0895445        total: 2.32s    remaining: 658ms
779:    learn: 0.0894716        total: 2.32s    remaining: 655ms
780:    learn: 0.0894403        total: 2.33s    remaining: 652ms
781:    learn: 0.0893345        total: 2.33s    remaining: 649ms
782:    learn: 0.0892995        total: 2.33s    remaining: 645ms
783:    learn: 0.0891618        total: 2.33s    remaining: 642ms
784:    learn: 0.0891307        total: 2.33s    remaining: 639ms
785:    learn: 0.0890606        total: 2.33s    remaining: 635ms
786:    learn: 0.0889611        total: 2.33s    remaining: 632ms
787:    learn: 0.0887910        total: 2.34s    remaining: 629ms
788:    learn: 0.0886685        total: 2.34s    remaining: 625ms
789:    learn: 0.0885137        total: 2.34s    remaining: 622ms
790:    learn: 0.0884352        total: 2.34s    remaining: 619ms
791:    learn: 0.0883120        total: 2.34s    remaining: 615ms
792:    learn: 0.0881308        total: 2.34s    remaining: 612ms
793:    learn: 0.0879960        total: 2.35s    remaining: 609ms
794:    learn: 0.0879311        total: 2.35s    remaining: 605ms
795:    learn: 0.0878445        total: 2.35s    remaining: 602ms
796:    learn: 0.0877474        total: 2.35s    remaining: 599ms
797:    learn: 0.0876364        total: 2.35s    remaining: 595ms
798:    learn: 0.0875549        total: 2.35s    remaining: 592ms
799:    learn: 0.0875111        total: 2.35s    remaining: 589ms
800:    learn: 0.0874601        total: 2.36s    remaining: 585ms
801:    learn: 0.0874049        total: 2.36s    remaining: 582ms
802:    learn: 0.0873467        total: 2.36s    remaining: 579ms
803:    learn: 0.0872976        total: 2.36s    remaining: 576ms
804:    learn: 0.0871483        total: 2.37s    remaining: 573ms
805:    learn: 0.0870389        total: 2.37s    remaining: 571ms
806:    learn: 0.0869588        total: 2.37s    remaining: 567ms
807:    learn: 0.0868473        total: 2.37s    remaining: 564ms
808:    learn: 0.0867120        total: 2.38s    remaining: 561ms
809:    learn: 0.0865408        total: 2.38s    remaining: 558ms
810:    learn: 0.0864938        total: 2.38s    remaining: 555ms
811:    learn: 0.0864329        total: 2.38s    remaining: 552ms
```

```
811:	learn: 0.0864329	total: 2.38s	remaining: 552ms
812:	learn: 0.0863822	total: 2.38s	remaining: 549ms
813:	learn: 0.0862961	total: 2.39s	remaining: 546ms
814:	learn: 0.0861732	total: 2.39s	remaining: 543ms
815:	learn: 0.0861181	total: 2.4s	remaining: 540ms
816:	learn: 0.0860815	total: 2.4s	remaining: 538ms
817:	learn: 0.0860048	total: 2.4s	remaining: 535ms
818:	learn: 0.0858447	total: 2.41s	remaining: 532ms
819:	learn: 0.0857256	total: 2.41s	remaining: 529ms
820:	learn: 0.0856705	total: 2.41s	remaining: 526ms
821:	learn: 0.0856531	total: 2.42s	remaining: 524ms
822:	learn: 0.0855338	total: 2.42s	remaining: 521ms
823:	learn: 0.0853442	total: 2.43s	remaining: 519ms
824:	learn: 0.0852552	total: 2.43s	remaining: 516ms
825:	learn: 0.0851281	total: 2.44s	remaining: 513ms
826:	learn: 0.0850640	total: 2.44s	remaining: 510ms
827:	learn: 0.0850147	total: 2.44s	remaining: 507ms
828:	learn: 0.0849114	total: 2.44s	remaining: 504ms
829:	learn: 0.0848341	total: 2.45s	remaining: 501ms
830:	learn: 0.0847345	total: 2.45s	remaining: 498ms
831:	learn: 0.0846484	total: 2.45s	remaining: 495ms
832:	learn: 0.0844766	total: 2.46s	remaining: 492ms
833:	learn: 0.0844169	total: 2.46s	remaining: 490ms
834:	learn: 0.0843192	total: 2.46s	remaining: 487ms
835:	learn: 0.0841642	total: 2.46s	remaining: 483ms
836:	learn: 0.0840926	total: 2.47s	remaining: 480ms
837:	learn: 0.0840211	total: 2.47s	remaining: 477ms
838:	learn: 0.0838887	total: 2.47s	remaining: 474ms
839:	learn: 0.0837807	total: 2.48s	remaining: 472ms
840:	learn: 0.0837074	total: 2.48s	remaining: 469ms
841:	learn: 0.0836924	total: 2.48s	remaining: 465ms
842:	learn: 0.0836363	total: 2.48s	remaining: 462ms
843:	learn: 0.0836136	total: 2.48s	remaining: 459ms
844:	learn: 0.0835705	total: 2.48s	remaining: 456ms
845:	learn: 0.0835222	total: 2.48s	remaining: 452ms
846:	learn: 0.0834434	total: 2.49s	remaining: 450ms
847:	learn: 0.0834048	total: 2.49s	remaining: 447ms
848:	learn: 0.0833456	total: 2.49s	remaining: 444ms
849:	learn: 0.0832776	total: 2.5s	remaining: 440ms
850:	learn: 0.0832074	total: 2.5s	remaining: 437ms
851:	learn: 0.0831729	total: 2.5s	remaining: 434ms
852:	learn: 0.0831014	total: 2.5s	remaining: 431ms
853:	learn: 0.0830086	total: 2.5s	remaining: 428ms
854:	learn: 0.0828917	total: 2.5s	remaining: 425ms
855:	learn: 0.0827539	total: 2.5s	remaining: 422ms
856:	learn: 0.0826612	total: 2.51s	remaining: 418ms
857:	learn: 0.0825462	total: 2.51s	remaining: 415ms
858:	learn: 0.0824646	total: 2.51s	remaining: 412ms
859:	learn: 0.0824162	total: 2.51s	remaining: 409ms
860:	learn: 0.0823395	total: 2.51s	remaining: 406ms
861:	learn: 0.0822491	total: 2.52s	remaining: 403ms
862:	learn: 0.0820848	total: 2.52s	remaining: 399ms
863:	learn: 0.0819752	total: 2.52s	remaining: 396ms
864:	learn: 0.0819531	total: 2.52s	remaining: 393ms
865:	learn: 0.0818661	total: 2.52s	remaining: 390ms
866:	learn: 0.0817662	total: 2.52s	remaining: 387ms
867:	learn: 0.0816562	total: 2.52s	remaining: 384ms
868:	learn: 0.0815548	total: 2.53s	remaining: 381ms
869:	learn: 0.0814765	total: 2.53s	remaining: 378ms
870:	learn: 0.0813821	total: 2.53s	remaining: 375ms
871:	learn: 0.0812106	total: 2.53s	remaining: 372ms
872:	learn: 0.0811357	total: 2.53s	remaining: 369ms
873:	learn: 0.0810898	total: 2.53s	remaining: 365ms
874:	learn: 0.0810270	total: 2.54s	remaining: 362ms
875:	learn: 0.0809498	total: 2.54s	remaining: 359ms
876:	learn: 0.0809140	total: 2.54s	remaining: 356ms
877:	learn: 0.0807642	total: 2.54s	remaining: 353ms
878:	learn: 0.0806474	total: 2.54s	remaining: 350ms
879:	learn: 0.0804876	total: 2.55s	remaining: 347ms
880:	learn: 0.0804238	total: 2.55s	remaining: 344ms
881:	learn: 0.0803486	total: 2.55s	remaining: 341ms
882:	learn: 0.0802734	total: 2.55s	remaining: 338ms
883:	learn: 0.0802038	total: 2.55s	remaining: 335ms
884:	learn: 0.0800955	total: 2.56s	remaining: 332ms
885:	learn: 0.0800386	total: 2.56s	remaining: 329ms
886:	learn: 0.0799722	total: 2.56s	remaining: 326ms
887:	learn: 0.0799123	total: 2.56s	remaining: 323ms
888:	learn: 0.0798410	total: 2.56s	remaining: 320ms
889:	learn: 0.0798172	total: 2.57s	remaining: 317ms
890:	learn: 0.0797762	total: 2.57s	remaining: 314ms
891:	learn: 0.0796733	total: 2.57s	remaining: 312ms
892:	learn: 0.0795062	total: 2.58s	remaining: 309ms
893:	learn: 0.0793631	total: 2.58s	remaining: 306ms
894:	learn: 0.0792232	total: 2.58s	remaining: 303ms
895:	learn: 0.0791570	total: 2.58s	remaining: 300ms
896:	learn: 0.0791218	total: 2.59s	remaining: 297ms
897:	learn: 0.0789795	total: 2.59s	remaining: 294ms
898:	learn: 0.0788689	total: 2.59s	remaining: 291ms
899:	learn: 0.0787687	total: 2.59s	remaining: 288ms
900:	learn: 0.0787106	total: 2.59s	remaining: 285ms
901:	learn: 0.0786599	total: 2.59s	remaining: 282ms
```

```
902:	learn: 0.0785775	total: 2.6s	remaining: 279ms
903:	learn: 0.0785496	total: 2.6s	remaining: 276ms
904:	learn: 0.0784949	total: 2.6s	remaining: 273ms
905:	learn: 0.0784205	total: 2.6s	remaining: 270ms
906:	learn: 0.0783742	total: 2.6s	remaining: 267ms
907:	learn: 0.0782726	total: 2.6s	remaining: 264ms
908:	learn: 0.0781082	total: 2.6s	remaining: 261ms
909:	learn: 0.0780354	total: 2.61s	remaining: 258ms
910:	learn: 0.0779239	total: 2.61s	remaining: 255ms
911:	learn: 0.0779034	total: 2.61s	remaining: 252ms
912:	learn: 0.0778179	total: 2.61s	remaining: 249ms
913:	learn: 0.0777444	total: 2.61s	remaining: 246ms
914:	learn: 0.0777251	total: 2.62s	remaining: 243ms
915:	learn: 0.0776585	total: 2.62s	remaining: 240ms
916:	learn: 0.0775876	total: 2.62s	remaining: 237ms
917:	learn: 0.0774913	total: 2.62s	remaining: 234ms
918:	learn: 0.0773368	total: 2.62s	remaining: 231ms
919:	learn: 0.0772025	total: 2.63s	remaining: 228ms
920:	learn: 0.0771414	total: 2.63s	remaining: 225ms
921:	learn: 0.0770956	total: 2.63s	remaining: 222ms
922:	learn: 0.0769996	total: 2.63s	remaining: 219ms
923:	learn: 0.0769035	total: 2.63s	remaining: 216ms
924:	learn: 0.0767897	total: 2.63s	remaining: 213ms
925:	learn: 0.0767521	total: 2.63s	remaining: 211ms
926:	learn: 0.0767065	total: 2.64s	remaining: 208ms
927:	learn: 0.0766463	total: 2.64s	remaining: 205ms
928:	learn: 0.0765922	total: 2.64s	remaining: 202ms
929:	learn: 0.0764747	total: 2.64s	remaining: 199ms
930:	learn: 0.0763665	total: 2.64s	remaining: 196ms
931:	learn: 0.0762115	total: 2.64s	remaining: 193ms
932:	learn: 0.0760858	total: 2.65s	remaining: 190ms
933:	learn: 0.0759594	total: 2.65s	remaining: 187ms
934:	learn: 0.0758602	total: 2.65s	remaining: 184ms
935:	learn: 0.0757910	total: 2.65s	remaining: 181ms
936:	learn: 0.0757019	total: 2.65s	remaining: 178ms
937:	learn: 0.0756252	total: 2.65s	remaining: 175ms
938:	learn: 0.0755688	total: 2.65s	remaining: 173ms
939:	learn: 0.0755087	total: 2.66s	remaining: 170ms
940:	learn: 0.0754579	total: 2.66s	remaining: 167ms
941:	learn: 0.0754002	total: 2.66s	remaining: 164ms
942:	learn: 0.0752739	total: 2.66s	remaining: 161ms
943:	learn: 0.0751798	total: 2.66s	remaining: 158ms
944:	learn: 0.0750295	total: 2.66s	remaining: 155ms
945:	learn: 0.0749157	total: 2.67s	remaining: 152ms
946:	learn: 0.0748613	total: 2.67s	remaining: 149ms
947:	learn: 0.0747205	total: 2.67s	remaining: 147ms
948:	learn: 0.0746675	total: 2.67s	remaining: 144ms
949:	learn: 0.0745480	total: 2.68s	remaining: 141ms
950:	learn: 0.0744959	total: 2.68s	remaining: 138ms
951:	learn: 0.0743695	total: 2.68s	remaining: 135ms
952:	learn: 0.0742967	total: 2.68s	remaining: 132ms
953:	learn: 0.0741953	total: 2.68s	remaining: 129ms
954:	learn: 0.0741282	total: 2.68s	remaining: 127ms
955:	learn: 0.0739977	total: 2.69s	remaining: 124ms
956:	learn: 0.0739443	total: 2.69s	remaining: 121ms
957:	learn: 0.0738576	total: 2.69s	remaining: 118ms
958:	learn: 0.0737688	total: 2.69s	remaining: 115ms
959:	learn: 0.0737203	total: 2.69s	remaining: 112ms
960:	learn: 0.0735865	total: 2.69s	remaining: 109ms
961:	learn: 0.0734862	total: 2.69s	remaining: 106ms
962:	learn: 0.0733840	total: 2.7s	remaining: 104ms
963:	learn: 0.0732947	total: 2.7s	remaining: 101ms
964:	learn: 0.0732414	total: 2.7s	remaining: 97.9ms
965:	learn: 0.0731090	total: 2.7s	remaining: 95.1ms
966:	learn: 0.0729786	total: 2.7s	remaining: 92.2ms
967:	learn: 0.0728586	total: 2.7s	remaining: 89.4ms
968:	learn: 0.0727875	total: 2.71s	remaining: 86.6ms
969:	learn: 0.0727395	total: 2.71s	remaining: 83.7ms
970:	learn: 0.0726457	total: 2.71s	remaining: 80.9ms
971:	learn: 0.0725479	total: 2.71s	remaining: 78.1ms
972:	learn: 0.0724656	total: 2.71s	remaining: 75.2ms
973:	learn: 0.0724214	total: 2.71s	remaining: 72.4ms
974:	learn: 0.0723080	total: 2.71s	remaining: 69.6ms
975:	learn: 0.0722723	total: 2.72s	remaining: 66.8ms
976:	learn: 0.0722230	total: 2.72s	remaining: 64ms
977:	learn: 0.0720937	total: 2.72s	remaining: 61.2ms
978:	learn: 0.0720044	total: 2.72s	remaining: 58.4ms
979:	learn: 0.0719084	total: 2.72s	remaining: 55.5ms
980:	learn: 0.0718354	total: 2.72s	remaining: 52.8ms
981:	learn: 0.0717817	total: 2.73s	remaining: 50ms
982:	learn: 0.0716583	total: 2.73s	remaining: 47.2ms
983:	learn: 0.0715800	total: 2.73s	remaining: 44.4ms
984:	learn: 0.0715350	total: 2.73s	remaining: 41.6ms
985:	learn: 0.0714401	total: 2.73s	remaining: 38.8ms
986:	learn: 0.0713384	total: 2.73s	remaining: 36ms
987:	learn: 0.0712929	total: 2.73s	remaining: 33.2ms
```

```
# Realizar predicciones en el conjunto de prueba
lgb_preds = lgb_model.predict(X_test)
cat_preds = cat_model.predict(X_test)
```

```
xgb_preds = xgb_model.predict(X_test)

# Calcular las métricas de evaluación
lgb_accuracy = accuracy_score(y_test, lgb_preds)
cat_accuracy = accuracy_score(y_test, cat_preds)
xgb_accuracy = accuracy_score(y_test, xgb_preds)

lgb_precision = precision_score(y_test, lgb_preds)
cat_precision = precision_score(y_test, cat_preds)
xgb_precision = precision_score(y_test, xgb_preds)

lgb_recall = recall_score(y_test, lgb_preds)
cat_recall = recall_score(y_test, cat_preds)
xgb_recall = recall_score(y_test, xgb_preds)

lgb_f1 = f1_score(y_test, lgb_preds)
cat_f1 = f1_score(y_test, cat_preds)
xgb_f1 = f1_score(y_test, xgb_preds)

lgb_roc_auc = roc_auc_score(y_test, lgb_preds)
cat_roc_auc = roc_auc_score(y_test, cat_preds)
xgb_roc_auc = roc_auc_score(y_test, xgb_preds)


# Mostrar los valores de las metricas
print(f"Accuracy de LightGBM: {lgb_accuracy}")
print(f"Accuracy de CatBoost: {cat_accuracy}")
print(f"Accuracy de XGBoost:  {xgb_accuracy}")
print('#############################')
print(f"Precision de LightGBM: {lgb_precision}")
print(f"Precision de CatBoost: {cat_precision}")
print(f"Precision de XGBoost:  {xgb_precision}")
print('#############################')
print(f"Recall de LightGBM: {lgb_recall}")
print(f"Recall de CatBoost: {cat_recall}")
print(f"Recall de XGBoost:  {xgb_recall}")
print('#############################')
print(f"F1 de LightGBM: {lgb_f1}")
print(f"F1 de CatBoost: {cat_f1}")
print(f"F1 de XGBoost:  {xgb_f1}")
print('#############################')
print(f"AUC de LightGBM: {lgb_roc_auc}")
print(f"AUC de CatBoost: {cat_roc_auc}")
print(f"AUC de XGBoost:  {xgb_roc_auc}")
```

```
    Accuracy de LightGBM: 0.9594594594594594
    Accuracy de CatBoost: 0.9662162162162162
    Accuracy de XGBoost:  0.956081081081081
    #############################
    Precision de LightGBM: 0.3333333333333333
    Precision de CatBoost: 1.0
    Precision de XGBoost:  0.25
    #############################
    Recall de LightGBM: 0.09090909090909091
    Recall de CatBoost: 0.09090909090909091
    Recall de XGBoost:  0.09090909090909091
    #############################
    F1 de LightGBM: 0.14285714285714288
    F1 de CatBoost: 0.16666666666666669
    F1 de XGBoost:  0.13333333333333333
    #############################
    AUC de LightGBM: 0.5419457735247208
    AUC de CatBoost: 0.5454545454545454
    AUC de XGBoost:  0.5401913875598086
```

```
# Calcular el Gini y el KS
lgb_pred_probs = lgb_model.predict_proba(X_test)[:, 1]
cat_pred_probs = cat_model.predict_proba(X_test)[:, 1]
xgb_pred_probs = xgb_model.predict_proba(X_test)[:, 1]

fpr, tpr, thresholds = roc_curve(y_test, lgb_pred_probs)
lgb_gini = (2 * roc_auc_score(y_test, lgb_pred_probs)) - 1
lgb_ks = max(tpr - fpr)

fpr, tpr, thresholds = roc_curve(y_test, cat_pred_probs)
cat_gini = (2 * roc_auc_score(y_test, cat_pred_probs)) - 1
cat_ks = max(tpr - fpr)

fpr, tpr, thresholds = roc_curve(y_test, xgb_pred_probs)
```

```
xgb_gini = (2 * roc_auc_score(y_test, xgb_pred_probs)) - 1
xgb_ks = max(tpr - fpr)


# Mostrar los valores de las metricas GINI y KS
print(f"GINI de LightGBM: {lgb_gini}")
print(f"GINI de CatBoost: {cat_gini}")
print(f"GINI de XGBoost:  {xgb_gini}")
print('#############################')
print(f"KS de LightGBM: {lgb_ks}")
print(f"KS de CatBoost: {cat_ks}")
print(f"KS de XGBoost:  {xgb_ks}")
```

```
GINI de LightGBM: 0.7403508771929823
GINI de CatBoost: 0.7422647527910684
GINI de XGBoost:  0.7607655502392343
#############################
KS de LightGBM: 0.7196172248803827
KS de CatBoost: 0.712280701754386
KS de XGBoost:  0.7094098883572568
```