fraud-detection-20188

May 24, 2024

#****Proyecto 2 - Security Data Science**** # ****Modelos de Deteccion de Fraude Bancario****

0.1 Nombre: Pedro Arriola

0.2 Carnet: 20188

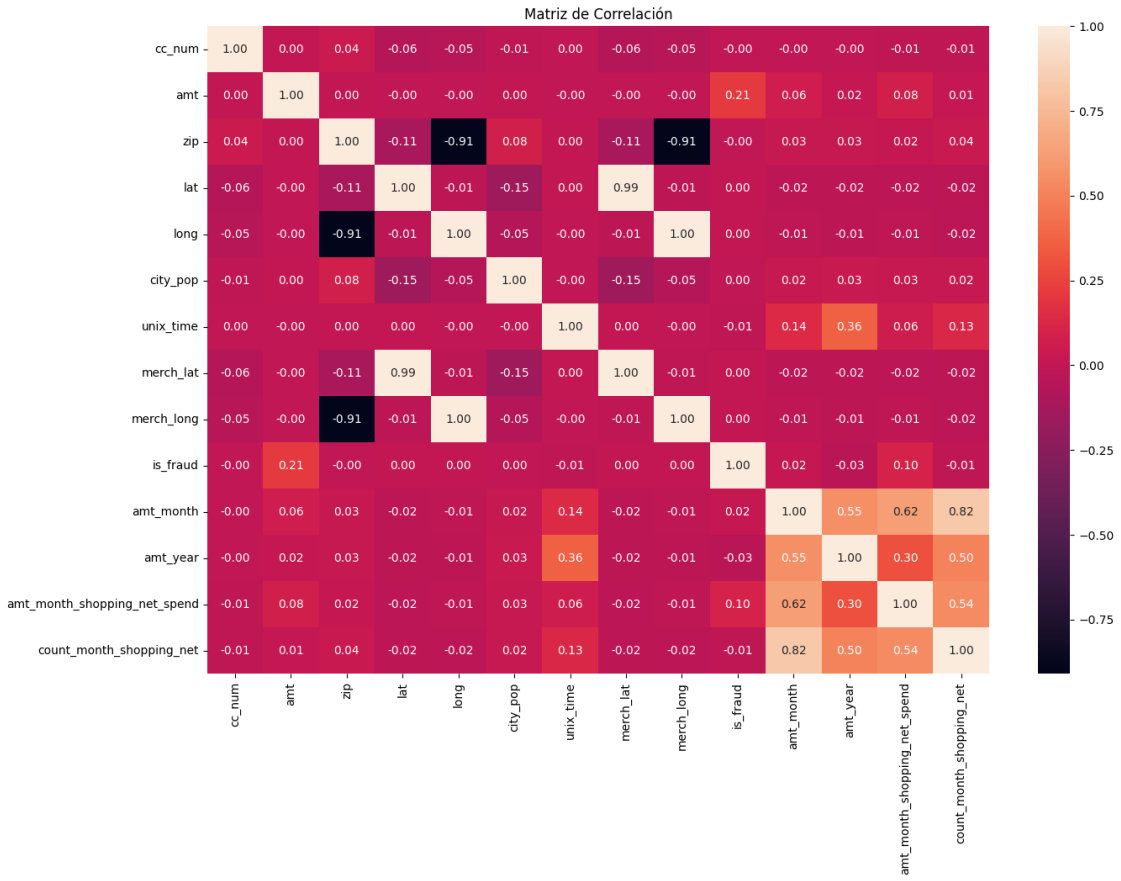
Este trabajo práctico tiene como objetivo investigar la viabilidad del entrenamiento incremental en modelos de aprendizaje automático y profundo, utilizando como estudio de caso un dataset de transacciones de tarjeta de crédito clasificadas en normales y fraudulentas. Los modelos a investigar den incluir 2 de los siguientes algoritmos: Redes Neuronales Artificiales (ANN), LightGBM, XGBoost, Random Forest y Máquinas de Vectores de Soporte (SVM).

0.2.1 Carga del Dataset

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar el dataset
file_path = '/content/drive/MyDrive/fraud/fraud_feature_engineering_example.csv'
df = pd.read_csv(file_path)

# Exploración inicial del dataset
df.head()
df.info()
df.describe()

# Verificar valores nulos
df.isnull().sum()

# Convertir 'trans_date_trans_time' a datetime
df['trans_date_trans_time'] = pd.to_datetime(df['trans_date_trans_time'])

# Seleccionar solo columnas numéricas para la matriz de correlación
numerical_columns = df.select_dtypes(include=[float, int]).columns
corr_matrix = df[numerical_columns].corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, fmt='.2f')
```

```
plt.title('Matriz de Correlación')
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1852394 entries, 0 to 1852393
Data columns (total 27 columns):
 #   Column                         Dtype
---  ------                         -----
 0   trans_date_trans_time          object
 1   cc_num                         int64
 2   merchant                       object
 3   category                       object
 4   amt                            float64
 5   first                          object
 6   last                           object
 7   gender                         object
 8   street                         object
 9   city                           object
 10  state                          object
 11  zip                            int64
 12  lat                            float64
 13  long                           float64
 14  city_pop                       int64
 15  job                            object
 16  dob                            object
 17  trans_num                      object
 18  unix_time                      int64
 19  merch_lat                      float64
 20  merch_long                     float64
 21  is_fraud                       int64
 22  amt_month                      float64
 23  amt_year                       float64
 24  amt_month_shopping_net_spend   float64
 25  count_month_shopping_net       float64
 26  first_time_at_merchant         bool
dtypes: bool(1), float64(9), int64(5), object(12)
memory usage: 369.2+ MB
```



0.2.2 Ingenieria de Variables

[ ]:
```
# Crear variables temporales
df['Transaction_hour'] = df['trans_date_trans_time'].dt.hour
df['Transaction_day'] = df['trans_date_trans_time'].dt.day
df['Transaction_month'] = df['trans_date_trans_time'].dt.month
df['Transaction_year'] = df['trans_date_trans_time'].dt.year
df['Transaction_day_of_week'] = df['trans_date_trans_time'].dt.dayofweek
df['Transaction_weekend'] = df['Transaction_day_of_week'].isin([5, 6]).
 ↪astype(int)
df['Transaction_quarter'] = df['trans_date_trans_time'].dt.quarter

# Crear variables de días desde la última y primera transacción del mismo␣
 ↪cliente
df = df.sort_values(by=['cc_num', 'trans_date_trans_time'])
df['Days_since_last_transaction'] = df.
 ↪groupby('cc_num')['trans_date_trans_time'].diff().dt.days
df['Days_since_last_transaction'].fillna(0, inplace=True)  # Rellenar NaN con 0␣
 ↪para la primera transacción
```

```
df['Days_since_first_transaction'] = df['trans_date_trans_time'] - df.
 ↪groupby('cc_num')['trans_date_trans_time'].transform('min')
df['Days_since_first_transaction'] = df['Days_since_first_transaction'].dt.days

# Verificación de las nuevas características
df_temporal_features = df[['Transaction_hour', 'Transaction_day',␣
 ↪'Transaction_month', 'Transaction_day_of_week',
                           'Transaction_weekend', 'Transaction_quarter',␣
 ↪'Days_since_last_transaction', 'Days_since_first_transaction']]
df_temporal_features.head()
```

[ ]:

	Transaction_hour	Transaction_day	Transaction_month	\
1017	12	1	1	
2724	8	2	1	
2726	8	2	1	
2882	12	2	1	
2907	13	2	1	

	Transaction_day_of_week	Transaction_weekend	Transaction_quarter	\
1017	1	0	1	
2724	2	0	1	
2726	2	0	1	
2882	2	0	1	
2907	2	0	1	

	Days_since_last_transaction	Days_since_first_transaction
1017	0.0	0
2724	0.0	0
2726	0.0	0
2882	0.0	0
2907	0.0	1

[ ]:
```
# Creación de variables basadas en el monto
df['Transaction_amount_log'] = np.log1p(df['amt'])
df['Transaction_amount_square'] = df['amt'] ** 2
df['Transaction_amount_sqrt'] = np.sqrt(df['amt'])
df['Transaction_amount_cubed'] = df['amt'] ** 3

# Variables de estadísticas por cliente
df['Avg_transaction_amount'] = df.groupby('cc_num')['amt'].transform('mean')
df['Std_transaction_amount'] = df.groupby('cc_num')['amt'].transform('std').
 ↪fillna(0)
df['Min_transaction_amount'] = df.groupby('cc_num')['amt'].transform('min')
df['Max_transaction_amount'] = df.groupby('cc_num')['amt'].transform('max')

# Verificación de las nuevas características
```

```
df_monto_features = df[['Transaction_amount_log', 'Transaction_amount_square',␣
 ↪'Transaction_amount_sqrt',
                        'Transaction_amount_cubed', 'Avg_transaction_amount',␣
 ↪'Std_transaction_amount',
                        'Min_transaction_amount', 'Max_transaction_amount']]
df_monto_features.head()
```

[ ]:

	Transaction_amount_log	Transaction_amount_square	\
1017	2.112635	52.8529	
2724	3.987872	2802.6436	
2726	4.419804	6737.1264	
2882	3.577669	1210.3441	
2907	3.338613	738.7524	

	Transaction_amount_sqrt	Transaction_amount_cubed	\
1017	2.696294	384.240583	
2724	7.275988	148371.952184	
2726	9.059801	552983.334912	
2882	5.898305	42107.871239	
2907	5.213444	20079.290232	

	Avg_transaction_amount	Std_transaction_amount	Min_transaction_amount	\
1017	59.257796	142.869746	1.02	
2724	59.257796	142.869746	1.02	
2726	59.257796	142.869746	1.02	
2882	59.257796	142.869746	1.02	
2907	59.257796	142.869746	1.02	

	Max_transaction_amount
1017	3437.46
2724	3437.46
2726	3437.46
2882	3437.46
2907	3437.46

[ ]:
```
# Variables de frecuencia
df['Num_transactions_last_day'] = df.groupby('cc_num')['trans_date_trans_time'].
 ↪transform(lambda x: (x.diff().dt.days.fillna(0) == 0).cumsum())
df['Num_transactions_last_week'] = df.
 ↪groupby('cc_num')['trans_date_trans_time'].transform(lambda x: (x.diff().dt.
 ↪days.fillna(0) < 7).cumsum())
df['Num_transactions_last_month'] = df.
 ↪groupby('cc_num')['trans_date_trans_time'].transform(lambda x: (x.diff().dt.
 ↪days.fillna(0) < 30).cumsum())
```

```
df['Avg_transactions_per_day'] = df.groupby(['cc_num',␣
 ↪df['trans_date_trans_time'].dt.date])['trans_date_trans_time'].
 ↪transform('count') / df.groupby('cc_num')['trans_date_trans_time'].
 ↪transform(lambda x: x.dt.date.nunique())

# Verificación de las nuevas características
df_frecuencia_features = df[['Num_transactions_last_day',␣
 ↪'Num_transactions_last_week', 'Num_transactions_last_month',␣
 ↪'Avg_transactions_per_day']]
df_frecuencia_features.head()
```

[ ]:

	Num_transactions_last_day	Num_transactions_last_week	\
1017	1	1	
2724	2	2	
2726	3	3	
2882	4	4	
2907	5	5	

	Num_transactions_last_month	Avg_transactions_per_day
1017	1	0.001464
2724	2	0.005857
2726	3	0.005857
2882	4	0.005857
2907	5	0.005857

[ ]:
```
# Función para calcular la distancia euclidiana
def haversine(lat1, lon1, lat2, lon2):
    R = 6371  # radio de la Tierra en kilómetros
    phi1 = np.radians(lat1)
    phi2 = np.radians(lat2)
    delta_phi = np.radians(lat2 - lat1)
    delta_lambda = np.radians(lon2 - lon1)
    a = np.sin(delta_phi / 2) ** 2 + np.cos(phi1) * np.cos(phi2) * np.
 ↪sin(delta_lambda / 2) ** 2
    return R * 2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a))

# Calcular la distancia entre el hogar del cliente y el comercio
df['Distance_home_merch'] = haversine(df['lat'], df['long'], df['merch_lat'],␣
 ↪df['merch_long'])

# Calcular la distancia entre el comercio de la última transacción y el␣
 ↪comercio actual
df = df.sort_values(by=['cc_num', 'trans_date_trans_time'])
df['Distance_last_trans_merch'] = df.groupby('cc_num').apply(lambda x:␣
 ↪haversine(x['merch_lat'], x['merch_long'], x['merch_lat'].shift(),␣
 ↪x['merch_long'].shift())).reset_index(level=0, drop=True)
```

```
df['Distance_last_trans_merch'].fillna(0, inplace=True)

# Calcular la distancia promedio y máxima entre el hogar del cliente y los␣
 ↪comercios
df['Avg_distance_home_merch'] = df.groupby('cc_num')['Distance_home_merch'].
 ↪transform('mean')
df['Max_distance_home_merch'] = df.groupby('cc_num')['Distance_home_merch'].
 ↪transform('max')

# Verificación de las nuevas características
df_geograficas_features = df[['Distance_home_merch',␣
 ↪'Distance_last_trans_merch', 'Avg_distance_home_merch',␣
 ↪'Max_distance_home_merch']]
df_geograficas_features.head()
```

[ ]:

	Distance_home_merch	Distance_last_trans_merch	Avg_distance_home_merch	\
1017	127.606239	0.000000	73.500496	
2724	110.308921	224.769219	73.500496	
2726	21.787261	105.220439	73.500496	
2882	87.204215	88.152283	73.500496	
2907	74.212965	132.876773	73.500496	

	Max_distance_home_merch
1017	137.124827
2724	137.124827
2726	137.124827
2882	137.124827
2907	137.124827

[ ]:
```
# Calcular la edad del cliente
df['dob'] = pd.to_datetime(df['dob'])
df['Age'] = df['trans_date_trans_time'].dt.year - df['dob'].dt.year

# Crear variables demográficas
df['Is_senior'] = (df['Age'] >= 65).astype(int)
df['Is_teenager'] = ((df['Age'] >= 13) & (df['Age'] <= 19)).astype(int)

# Verificación de las nuevas características
df_demograficas_features = df[['Age', 'Is_senior', 'Is_teenager']]
df_demograficas_features.head()
```

[ ]:

	Age	Is_senior	Is_teenager
1017	33	0	0
2724	33	0	0
2726	33	0	0
2882	33	0	0
2907	33	0	0

[ ]:
```
# Variables históricas
df['Total_transactions'] = df.groupby('cc_num')['amt'].transform('count')
df['Total_amount_spent'] = df.groupby('cc_num')['amt'].transform('sum')
df['Avg_transaction_amount_category'] = df.groupby(['cc_num',␣
 ↪'category'])['amt'].transform('mean')
df['Std_transaction_amount_category'] = df.groupby(['cc_num',␣
 ↪'category'])['amt'].transform('std').fillna(0)

# Verificación de las nuevas características
df_historicas_features = df[['Total_transactions', 'Total_amount_spent',␣
 ↪'Avg_transaction_amount_category', 'Std_transaction_amount_category']]
df_historicas_features.head()
```

[ ]:

	Total_transactions	Total_amount_spent	Avg_transaction_amount_category	\
1017	2196	130130.12	45.596296	
2724	2196	130130.12	59.779429	
2726	2196	130130.12	59.779429	
2882	2196	130130.12	56.438434	
2907	2196	130130.12	55.924559	

	Std_transaction_amount_category
1017	113.527496
2724	15.758267
2726	15.758267
2882	65.433283
2907	53.243029

[ ]:
```
# Variables de ratio
df['Transaction_amount_to_avg_ratio'] = df['amt'] / df['Avg_transaction_amount']
df['Transaction_amount_to_max_ratio'] = df['amt'] / df['Max_transaction_amount']
df['Transaction_amount_to_min_ratio'] = df['amt'] / df['Min_transaction_amount']

# Verificación de las nuevas características
df_ratio_features = df[['Transaction_amount_to_avg_ratio',␣
 ↪'Transaction_amount_to_max_ratio', 'Transaction_amount_to_min_ratio']]
df_ratio_features.head()
```

[ ]:

	Transaction_amount_to_avg_ratio	Transaction_amount_to_max_ratio	\
1017	0.122684	0.002115	
2724	0.893385	0.015401	
2726	1.385134	0.023878	
2882	0.587096	0.010121	
2907	0.458674	0.007907	

	Transaction_amount_to_min_ratio
1017	7.127451
2724	51.901961

```
2726                        80.470588
2882                        34.107843
2907                        26.647059
```

[ ]:
```
# Variables de comportamiento
df['Is_first_transaction'] = (df.groupby('cc_num')['trans_date_trans_time'].
 ↪rank(method='first') == 1).astype(int)
df['Is_new_merchant'] = (~df.groupby('cc_num')['merchant'].transform(lambda x:␣
 ↪x.duplicated(keep='first'))).astype(int)
df['Is_new_category'] = (~df.groupby('cc_num')['category'].transform(lambda x:␣
 ↪x.duplicated(keep='first'))).astype(int)

# Verificación de las nuevas características
df_comportamiento_features = df[['Is_first_transaction', 'Is_new_merchant',␣
 ↪'Is_new_category']]
df_comportamiento_features.head()
```

[ ]:
```
      Is_first_transaction  Is_new_merchant  Is_new_category
1017                     1                1                1
2724                     0                1                1
2726                     0                1                0
2882                     0                1                1
2907                     0                1                1
```

[ ]:
```
# Variables de interacción
df['Hour_amount_interaction'] = df['Transaction_hour'] * df['amt']
df['Day_amount_interaction'] = df['Transaction_day'] * df['amt']
df['Month_amount_interaction'] = df['Transaction_month'] * df['amt']

# Verificación de las nuevas características
df_interaccion_features = df[['Hour_amount_interaction',␣
 ↪'Day_amount_interaction', 'Month_amount_interaction']]
df_interaccion_features.head()
```

[ ]:
```
      Hour_amount_interaction  Day_amount_interaction  \
1017                    87.24                    7.27
2724                   423.52                  105.88
2726                   656.64                  164.16
2882                   417.48                   69.58
2907                   353.34                   54.36

      Month_amount_interaction
1017                      7.27
2724                     52.94
2726                     82.08
2882                     34.79
2907                     27.18
```

[ ]:
```
# Variables de comercio
df['Num_unique_merchants'] = df.groupby('cc_num')['merchant'].
 ↪transform('nunique')
df['Num_unique_categories'] = df.groupby('cc_num')['category'].
 ↪transform('nunique')
df['Avg_transactions_per_merchant'] = df.groupby('merchant')['amt'].
 ↪transform('count') / df.groupby('merchant')['cc_num'].transform('nunique')

# Verificación de las nuevas características
df_comercio_features = df[['Num_unique_merchants', 'Num_unique_categories',␣
 ↪'Avg_transactions_per_merchant']]
df_comercio_features.head()
```

[ ]:
```
      Num_unique_merchants  Num_unique_categories  \
1017                   642                     14
2724                   642                     14
2726                   642                     14
2882                   642                     14
2907                   642                     14

      Avg_transactions_per_merchant
1017                       2.614826
2724                       4.645399
2726                       4.614144
2882                       3.828947
2907                       4.248175
```

[ ]:
```
# Función para determinar si una fecha es festiva
# Función para determinar si una fecha es festiva (solo mes y día)
def is_holiday(date):
    holidays = [('01-01', '12-25')]  # Ejemplo de días festivos (mes-día)
    return date.strftime('%m-%d') in holidays

# Variables de contexto
df['Is_holiday'] = df['trans_date_trans_time'].apply(is_holiday).astype(int)
df['Is_weekend_transaction'] = df['Transaction_day_of_week'].isin([5, 6]).
 ↪astype(int)
df['Merchant_transaction_count'] = df.groupby('merchant')['amt'].
 ↪transform('count')
df['Category_transaction_count'] = df.groupby('category')['amt'].
 ↪transform('count')
df['Is_high_risk_merchant'] = df['merchant'].isin(['fraud_Rippin, Kub and␣
 ↪Mann', 'fraud_Stokes LLC and Sons']).astype(int)  # Ejemplo de comercios de␣
 ↪alto riesgo

# Verificación de las nuevas características
```

```
df_contexto_features = df[['Is_holiday', 'Is_weekend_transaction',␣
 ↪'Merchant_transaction_count', 'Category_transaction_count',␣
 ↪'Is_high_risk_merchant']]
df_contexto_features.head()
```

[ ]:

	Is_holiday	Is_weekend_transaction	Merchant_transaction_count	Category_transaction_count	Is_high_risk_merchant
1017	0	0	1799	90654	0
2724	0	0	3786	188029	0
2726	0	0	3719	188029	0
2882	0	0	3201	161727	0
2907	0	0	3492	175460	0

[ ]:
```
# Calcular la tendencia del monto de transacción y la frecuencia de transacción
df = df.sort_values(by=['cc_num', 'trans_date_trans_time'])
df['Transaction_amount_trend'] = df.groupby('cc_num')['amt'].transform(lambda x:
 ↪ x.rolling(window=5, min_periods=1).mean())
df['Transaction_frequency_trend'] = df.
 ↪groupby('cc_num')['trans_date_trans_time'].transform(lambda x: x.diff().dt.
 ↪total_seconds().rolling(window=5, min_periods=1).mean())

# Calcular el tiempo desde la última transacción fraudulenta del mismo cliente
df['Time_since_last_fraud'] = df.groupby('cc_num')['trans_date_trans_time'].
 ↪transform(lambda x: x.diff().fillna(pd.Timedelta(seconds=0)).dt.
 ↪total_seconds())

# Calcular el ratio de transacciones fraudulentas del cliente al total de␣
 ↪transacciones
df['Fraud_ratio'] = df.groupby('cc_num')['is_fraud'].transform('mean')

# Calcular el intervalo promedio y la desviación estándar del intervalo entre␣
 ↪transacciones del cliente
df['Avg_transaction_interval'] = df.groupby('cc_num')['trans_date_trans_time'].
 ↪transform(lambda x: x.diff().mean().total_seconds())
df['Std_transaction_interval'] = df.groupby('cc_num')['trans_date_trans_time'].
 ↪transform(lambda x: x.diff().std().total_seconds())

# Indicador de si el monto de la transacción es significativamente mayor que el␣
 ↪promedio del cliente
df['Is_large_amount'] = (df['amt'] > df['Avg_transaction_amount']).astype(int)
```

```
# Indicador de actividad fraudulenta reciente
df['Recent_fraud_activity'] = df.groupby('cc_num')['is_fraud'].transform(lambda␣
 ↪x: x.rolling(window=5, min_periods=1).sum())

# Indicador de si la transacción ocurre durante horas de alto riesgo (ejemplo:␣
 ↪0-6 AM)
df['High_risk_time'] = df['Transaction_hour'].isin([0, 1, 2, 3, 4, 5, 6]).
 ↪astype(int)

# Verificación de las nuevas características
df_avanzadas_features = df[['Transaction_amount_trend',␣
 ↪'Transaction_frequency_trend', 'Time_since_last_fraud',
                             'Fraud_ratio', 'Avg_transaction_interval',␣
 ↪'Std_transaction_interval',
                             'Is_large_amount', 'Recent_fraud_activity',␣
 ↪'High_risk_time']]
df_avanzadas_features.head()
```

[ ]:

	Transaction_amount_trend	Transaction_frequency_trend	Time_since_last_fraud	Fraud_ratio	Avg_transaction_interval	Std_transaction_interval	Is_large_amount	Recent_fraud_activity	High_risk_time
1017	7.270	NaN	0.0	0.004098	28722.084282	32251.693854	0	0.0	0
2724	30.105	71862.000000	71862.0	0.004098	28722.084282	32251.693854	0	0.0	0
2726	47.430	36010.500000	159.0	0.004098	28722.084282	32251.693854	1	0.0	0
2882	44.270	28619.666667	13838.0	0.004098	28722.084282	32251.693854	0	0.0	0
2907	40.852	21952.750000	1952.0	0.004098	28722.084282	32251.693854	0	0.0	0

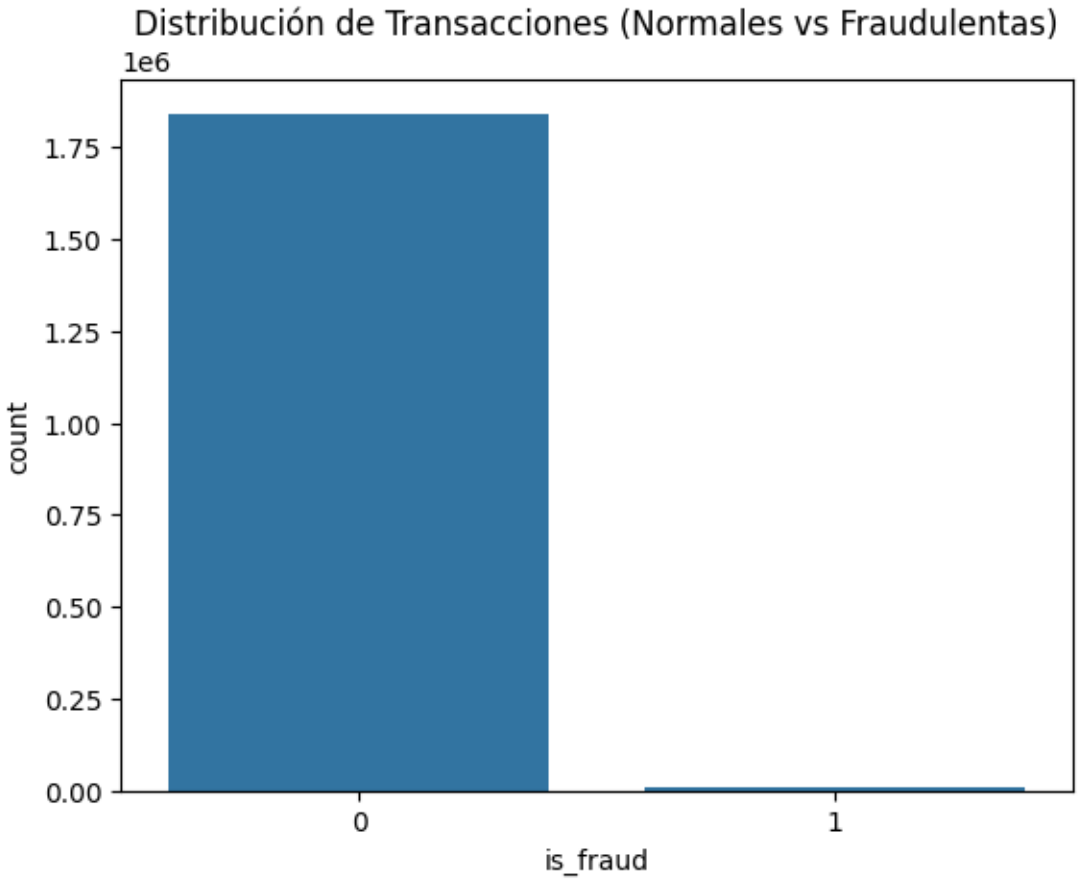
0.2.3 Análisis Exploratorio de Datos

[ ]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Exploración inicial del dataset
df.head()
df.info()
df.describe()

# Verificar valores nulos
df.isnull().sum()

# Visualización de la distribución de las clases
sns.countplot(x='is_fraud', data=df)
plt.title('Distribución de Transacciones (Normales vs Fraudulentas)')
plt.show()
```

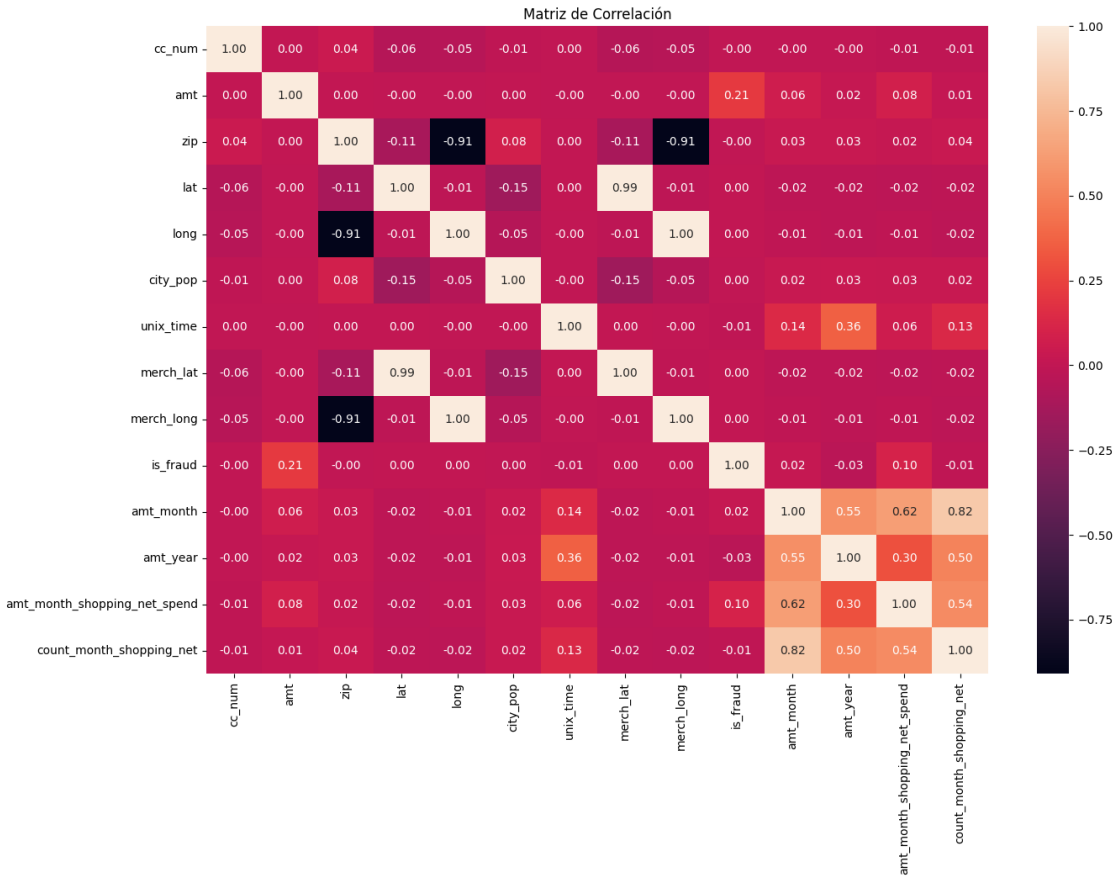
```
<class 'pandas.core.frame.DataFrame'>
Index: 1852394 entries, 1017 to 1850558
Data columns (total 71 columns):
 #   Column                          Dtype
---  ------                          -----
 0   amt                             float64
 1   lat                             float64
 2   long                            float64
 3   city_pop                        float64
 4   unix_time                       float64
 5   merch_lat                       float64
 6   merch_long                      float64
 7   is_fraud                        int64
 8   amt_month                       float64
 9   amt_year                        float64
 10  amt_month_shopping_net_spend    float64
 11  count_month_shopping_net        float64
 12  first_time_at_merchant          float64
 13  Transaction_hour                float64
 14  Transaction_day                 float64
 15  Transaction_month               float64
 16  Transaction_year                int32
 17  Transaction_day_of_week         float64
 18  Transaction_weekend             float64
 19  Transaction_quarter             float64
 20  Days_since_last_transaction     float64
 21  Days_since_first_transaction    float64
 22  Transaction_amount_log          float64
```

23	Transaction_amount_square	float64
24	Transaction_amount_sqrt	float64
25	Transaction_amount_cubed	float64
26	Avg_transaction_amount	float64
27	Std_transaction_amount	float64
28	Min_transaction_amount	float64
29	Max_transaction_amount	float64
30	Num_transactions_last_day	float64
31	Num_transactions_last_week	float64
32	Num_transactions_last_month	float64
33	Avg_transactions_per_day	float64
34	Distance_home_merch	float64
35	Distance_last_trans_merch	float64
36	Avg_distance_home_merch	float64
37	Max_distance_home_merch	float64
38	Age	float64
39	Is_senior	float64
40	Is_teenager	float64
41	Total_transactions	float64
42	Total_amount_spent	float64
43	Avg_transaction_amount_category	float64
44	Std_transaction_amount_category	float64
45	Transaction_amount_to_avg_ratio	float64
46	Transaction_amount_to_max_ratio	float64
47	Transaction_amount_to_min_ratio	float64
48	Is_first_transaction	float64
49	Is_new_merchant	float64
50	Is_new_category	float64
51	Hour_amount_interaction	float64
52	Day_amount_interaction	float64
53	Month_amount_interaction	float64
54	Num_unique_merchants	float64
55	Num_unique_categories	float64
56	Avg_transactions_per_merchant	float64
57	Is_holiday	float64
58	Is_weekend_transaction	float64
59	Merchant_transaction_count	float64
60	Category_transaction_count	float64
61	Is_high_risk_merchant	float64
62	Transaction_amount_trend	float64
63	Transaction_frequency_trend	float64
64	Time_since_last_fraud	float64
65	Fraud_ratio	float64
66	Avg_transaction_interval	float64
67	Std_transaction_interval	float64
68	Is_large_amount	float64
69	Recent_fraud_activity	float64
70	High_risk_time	float64

```
dtypes: float64(69), int32(1), int64(1)
memory usage: 1010.5 MB
```

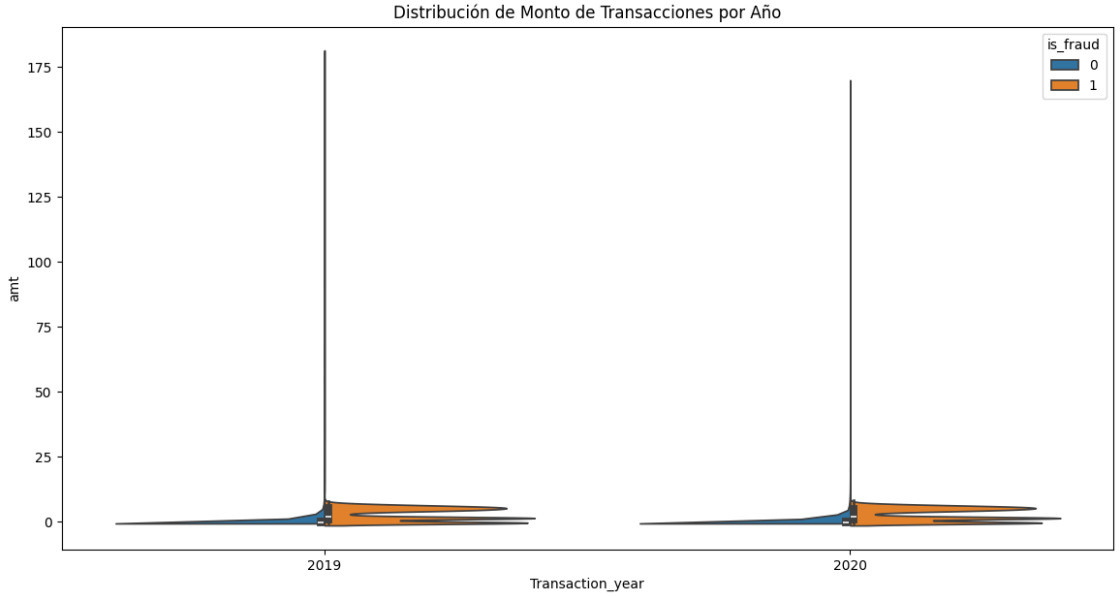


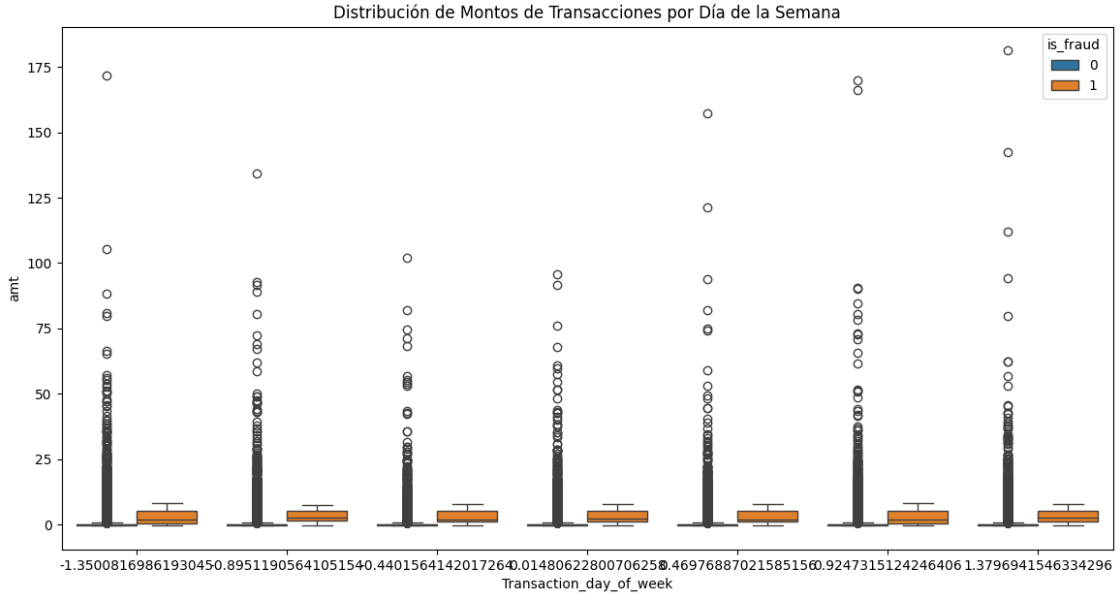
[ ]:
```python
# Matriz de correlación
numerical_columns = df.select_dtypes(include=[float, int]).columns
corr_matrix = df[numerical_columns].corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, fmt='.2f')
plt.title('Matriz de Correlación')
plt.show()
```

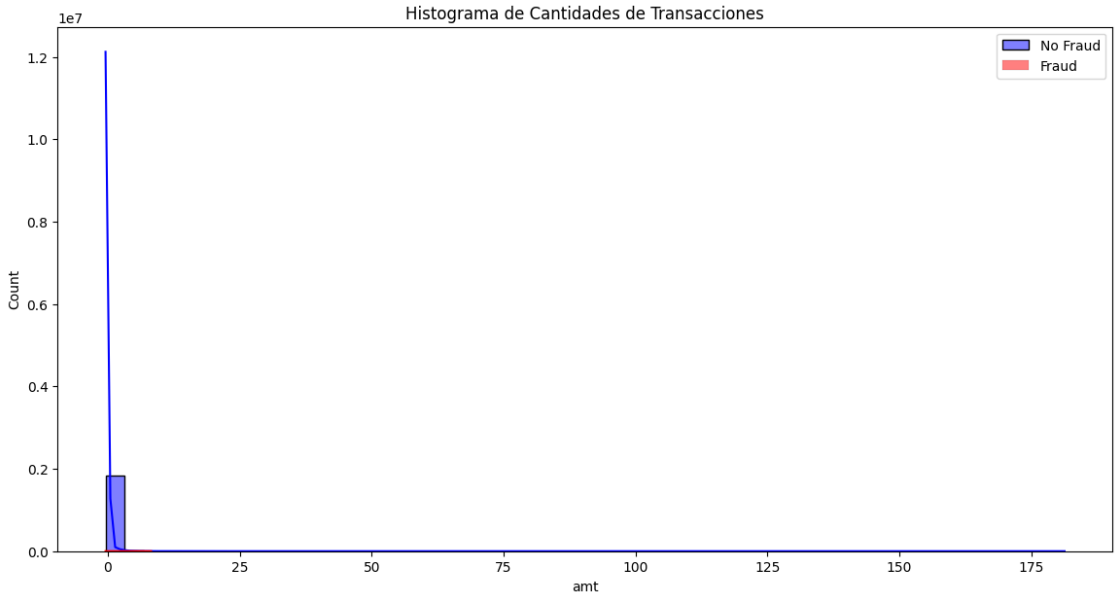


[ ]:
```
# Gráfico de violín para mostrar la distribución de montos por año
plt.figure(figsize=(14, 7))
sns.violinplot(x='Transaction_year', y='amt', hue='is_fraud', data=df,␣
 ↪split=True)
plt.title('Distribución de Monto de Transacciones por Año')
plt.show()

# Distribución de montos de transacciones por día de la semana
plt.figure(figsize=(14, 7))
sns.boxplot(x='Transaction_day_of_week', y='amt', hue='is_fraud', data=df)
plt.title('Distribución de Montos de Transacciones por Día de la Semana')
plt.show()

# Histograma de las cantidades de transacciones
plt.figure(figsize=(14, 7))
sns.histplot(df[df['is_fraud'] == 0]['amt'], bins=50, kde=True, color='blue',␣
 ↪label='No Fraud')
sns.histplot(df[df['is_fraud'] == 1]['amt'], bins=50, kde=True, color='red',␣
 ↪label='Fraud')
```

```
plt.title('Histograma de Cantidades de Transacciones')
plt.legend()
plt.show()
```







0.2.4 Estandarizacion/Normalizacion de los Datos

[ ]:
```
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE

# Llenar valores nulos
for column in df.columns:
    if df[column].dtype == 'object':
        df[column].fillna(df[column].mode()[0], inplace=True)  # Llenar con el␣
 ↪valor más frecuente para categóricas
    else:
        df[column].fillna(df[column].median(), inplace=True)  # Llenar con la␣
 ↪mediana para numéricas

# Eliminar variables no necesarias
df = df.drop(columns=['cc_num', 'trans_date_trans_time', 'merchant',␣
 ↪'category', 'gender', 'first', 'last', 'street', 'city', 'state', 'zip',␣
 ↪'job', 'dob', 'trans_num'])

# Seleccionar características a escalar (excluyendo la columna 'is_fraud' y␣
 ↪'year')
features_to_scale = df.columns.difference(['is_fraud', 'Transaction_year'])

# Inicializar el escalador
scaler = StandardScaler()

# Aplicar el escalador a las características seleccionadas
```

```
df[features_to_scale] = scaler.fit_transform(df[features_to_scale])

# Verificación de los datos normalizados/estandarizados
df[features_to_scale].head()

# Dividir el dataset en características (X) y la variable objetivo (y)
X = df.drop(columns=['is_fraud'])
y = df['is_fraud']

# Asegurarse de que no hay NaNs
df = df.fillna(0)

# Aplicar SMOTE para balancear las clases
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

0.2.5 Preparando el Dataset de Entrenamiento

[ ]:
```
from sklearn.model_selection import train_test_split

# Dividir el dataset en Train, Dev y Test
X_train, X_temp, y_train, y_temp = train_test_split(X_resampled, y_resampled,␣
 ↪test_size=0.4, random_state=42)
X_dev, X_test, y_dev, y_test = train_test_split(X_temp, y_temp, test_size=0.5,␣
 ↪random_state=42)

# Convertir los datos a float (ya deberían estar limpios de NaNs)
X_train = X_train.astype(float)
X_dev = X_dev.astype(float)
X_test = X_test.astype(float)

y_train = y_train.astype(int)
y_dev = y_dev.astype(int)
y_test = y_test.astype(int)

print(f'Tamaño del conjunto de entrenamiento: {X_train.shape[0]}')
print(f'Tamaño del conjunto de validación: {X_dev.shape[0]}')
print(f'Tamaño del conjunto de prueba: {X_test.shape[0]}')
```

```
Tamaño del conjunto de entrenamiento: 2211291
Tamaño del conjunto de validación: 737097
Tamaño del conjunto de prueba: 737098
```

0.2.6 Entrenamiento Normal de los Modelos

[ ]:
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in /usr/local/lib/python3.10/dist-
packages (2.0.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from xgboost) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from xgboost) (1.11.4)
```

[ ]:
```
!nvidia-smi
```

```
Fri May 24 05:57:30 2024
+---------------------------------------------------------------------------
--------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version:
12.2     |
|-----------------------------------------+----------------------+-----------
--------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util
Compute M. |
|                                         |                      |
MIG M. |
|=========================================+======================+===========
========|
|   0  NVIDIA L4                      Off | 00000000:00:03.0 Off |
0 |
| N/A   64C    P8              14W /  72W |      1MiB / 23034MiB |      0%
Default |
|                                         |                      |
N/A |
+-----------------------------------------+----------------------+-----------
--------+

+---------------------------------------------------------------------------
--------+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name                            GPU
Memory |
|        ID   ID
Usage      |
|=========================================================================
========|
|  No running processes found
```

```
|
+-----------------------------------------------------------------------------
--------+
```

[ ]:
```
# This get the RAPIDS-Colab install files and test check your GPU.  Run this␣
 ↪and the next cell only.
# Please read the output of this cell.  If your Colab Instance is not RAPIDS␣
 ↪compatible, it will warn you and give you remediation steps.
!git clone https://github.com/rapidsai/rapidsai-csp-utils.git
!python rapidsai-csp-utils/colab/pip-install.py
```

```
Cloning into 'rapidsai-csp-utils'…
remote: Enumerating objects: 481, done.
remote: Counting objects: 100% (212/212), done.
remote: Compressing objects: 100% (121/121), done.
remote: Total 481 (delta 143), reused 124 (delta 91), pack-reused 269
Receiving objects: 100% (481/481), 133.58 KiB | 759.00 KiB/s, done.
Resolving deltas: 100% (245/245), done.
Collecting pynvml
  Downloading pynvml-11.5.0-py3-none-any.whl (53 kB)
                             53.1/53.1 kB 2.2 MB/s eta 0:00:00
Installing collected packages: pynvml
Successfully installed pynvml-11.5.0
***********************************************************************
Woo! Your instance has a NVIDIA L4 GPU!
We will install the latest stable RAPIDS via pip 24.4.*!  Please stand by,
should be quick…
***********************************************************************

Looking in indexes: https://pypi.org/simple, https://pypi.nvidia.com
Collecting cuml-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cuml-
cu12/cuml_cu12-24.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1200.7 MB)
                             1.2/1.2 GB 1.7 MB/s eta 0:00:00
Collecting cugraph-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cugraph-cu12/cugraph_cu12-24.4.0-cp310-cp3
10-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1429.1 MB)
                             1.4/1.4 GB 1.5 MB/s eta 0:00:00
Collecting cuspatial-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cuspatial-cu12/cuspatial_cu12-24.4.0-cp310
-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (137.8 MB)
                             137.8/137.8 MB 6.7 MB/s eta
0:00:00
Collecting cuproj-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cuproj-cu12/cuproj_cu12-24.4.0-cp310-cp310
-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (920 kB)
                             920.9/920.9 kB 73.6 MB/s eta
```

```
0:00:00
Collecting cuxfilter-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cuxfilter-
cu12/cuxfilter_cu12-24.4.1-py3-none-any.whl (83 kB)
                            83.5/83.5 kB 11.0 MB/s eta 0:00:00
Collecting cucim-cu12==24.4.*
  Downloading https://pypi.nvidia.com/cucim-cu12/cucim_cu12-24.4.0-cp310-cp310-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.8 MB)
                            5.8/5.8 MB 102.2 MB/s eta 0:00:00
Collecting pylibraft-cu12==24.4.*
  Downloading https://pypi.nvidia.com/pylibraft-cu12/pylibraft_cu12-24.4.0-cp310
-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (823.0 MB)
                            823.0/823.0 MB 2.2 MB/s eta
0:00:00
Collecting raft-dask-cu12==24.4.*
  Downloading https://pypi.nvidia.com/raft-dask-cu12/raft_dask_cu12-24.4.0-cp310
-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (170.1 MB)
                            170.1/170.1 MB 9.3 MB/s eta
0:00:00
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-
packages (3.9.5)
Requirement already satisfied: cudf-cu12==24.4.* in
/usr/local/lib/python3.10/dist-packages (from cuml-cu12==24.4.*) (24.4.1)
Requirement already satisfied: cupy-cuda12x>=12.0.0 in
/usr/local/lib/python3.10/dist-packages (from cuml-cu12==24.4.*) (12.2.0)
Collecting dask-cuda==24.4.* (from cuml-cu12==24.4.*)
  Downloading dask_cuda-24.4.0-py3-none-any.whl (126 kB)
                            126.6/126.6 kB 3.5 MB/s eta
0:00:00
Collecting dask-cudf-cu12==24.4.* (from cuml-cu12==24.4.*)
  Downloading https://pypi.nvidia.com/dask-cudf-
cu12/dask_cudf_cu12-24.4.1-py3-none-any.whl (48 kB)
                            48.9/48.9 kB 7.4 MB/s eta 0:00:00
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.10/dist-
packages (from cuml-cu12==24.4.*) (1.4.2)
Requirement already satisfied: numba>=0.57 in /usr/local/lib/python3.10/dist-
packages (from cuml-cu12==24.4.*) (0.58.1)
Collecting rapids-dask-dependency==24.4.* (from cuml-cu12==24.4.*)
  Downloading https://pypi.nvidia.com/rapids-dask-
dependency/rapids_dask_dependency-24.4.1-py3-none-any.whl (15 kB)
Requirement already satisfied: rmm-cu12==24.4.* in
/usr/local/lib/python3.10/dist-packages (from cuml-cu12==24.4.*) (24.4.0)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.10/dist-
packages (from cuml-cu12==24.4.*) (1.11.4)
Collecting treelite==4.1.2 (from cuml-cu12==24.4.*)
  Downloading treelite-4.1.2-py3-none-manylinux2014_x86_64.whl (810 kB)
                            810.9/810.9 kB 9.6 MB/s eta
0:00:00
```

```
Requirement already satisfied: fsspec[http]>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from cugraph-cu12==24.4.*) (2023.6.0)
Requirement already satisfied: numpy<2.0a0,>=1.23 in
/usr/local/lib/python3.10/dist-packages (from cugraph-cu12==24.4.*) (1.25.2)
Collecting pylibcugraph-cu12==24.4.* (from cugraph-cu12==24.4.*)
  Downloading https://pypi.nvidia.com/pylibcugraph-cu12/pylibcugraph_cu12-24.4.0
-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1430.2 MB)
                           1.4/1.4 GB 1.5 MB/s eta 0:00:00
Collecting ucx-py-cu12==0.37.* (from cugraph-cu12==24.4.*)
  Downloading https://pypi.nvidia.com/ucx-py-cu12/ucx_py_cu12-0.37.0-cp310-cp310
-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.7 MB)
                           7.7/7.7 MB 99.5 MB/s eta 0:00:00
Requirement already satisfied: geopandas>=0.11.0 in
/usr/local/lib/python3.10/dist-packages (from cuspatial-cu12==24.4.*) (0.13.2)
Requirement already satisfied: bokeh>=3.1 in /usr/local/lib/python3.10/dist-
packages (from cuxfilter-cu12==24.4.*) (3.3.4)
Collecting datashader>=0.15 (from cuxfilter-cu12==24.4.*)
  Downloading datashader-0.16.1-py2.py3-none-any.whl (18.3 MB)
                           18.3/18.3 MB 54.9 MB/s eta 0:00:00
Requirement already satisfied: holoviews>=1.16.0 in
/usr/local/lib/python3.10/dist-packages (from cuxfilter-cu12==24.4.*) (1.17.1)
Collecting jupyter-server-proxy (from cuxfilter-cu12==24.4.*)
  Downloading jupyter_server_proxy-4.1.2-py3-none-any.whl (34 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from cuxfilter-cu12==24.4.*) (24.0)
Requirement already satisfied: panel>=1.0 in /usr/local/lib/python3.10/dist-
packages (from cuxfilter-cu12==24.4.*) (1.3.8)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from cucim-cu12==24.4.*) (8.1.7)
Requirement already satisfied: lazy-loader>=0.1 in
/usr/local/lib/python3.10/dist-packages (from cucim-cu12==24.4.*) (0.4)
Requirement already satisfied: scikit-image<0.23.0a0,>=0.19.0 in
/usr/local/lib/python3.10/dist-packages (from cucim-cu12==24.4.*) (0.19.3)
Requirement already satisfied: cuda-python<13.0a0,>=12.0 in
/usr/local/lib/python3.10/dist-packages (from pylibraft-cu12==24.4.*) (12.2.1)
Requirement already satisfied: cachetools in /usr/local/lib/python3.10/dist-
packages (from cudf-cu12==24.4.*->cuml-cu12==24.4.*) (5.3.3)
Requirement already satisfied: nvtx>=0.2.1 in /usr/local/lib/python3.10/dist-
packages (from cudf-cu12==24.4.*->cuml-cu12==24.4.*) (0.2.10)
Requirement already satisfied: pandas<2.2.2dev0,>=2.0 in
/usr/local/lib/python3.10/dist-packages (from cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (2.0.3)
Requirement already satisfied: protobuf<5,>=3.20 in
/usr/local/lib/python3.10/dist-packages (from cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (3.20.3)
Requirement already satisfied: pynvjitlink-cu12 in
/usr/local/lib/python3.10/dist-packages (from cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (0.2.3)
```

```
Requirement already satisfied: pyarrow<15.0.0a0,>=14.0.1 in
/usr/local/lib/python3.10/dist-packages (from cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (14.0.2)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from cudf-cu12==24.4.*->cuml-cu12==24.4.*) (13.7.1)
Requirement already satisfied: typing_extensions>=4.0.0 in
/usr/local/lib/python3.10/dist-packages (from cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (4.11.0)
Collecting pynvml<11.5,>=11.0.0 (from dask-cuda==24.4.*->cuml-cu12==24.4.*)
  Downloading pynvml-11.4.1-py3-none-any.whl (46 kB)
                           47.0/47.0 kB 7.4 MB/s eta 0:00:00
Requirement already satisfied: zict>=2.0.0 in /usr/local/lib/python3.10/dist-
packages (from dask-cuda==24.4.*->cuml-cu12==24.4.*) (3.0.0)
Collecting dask==2024.1.1 (from rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*)
  Downloading dask-2024.1.1-py3-none-any.whl (1.2 MB)
                           1.2/1.2 MB 68.1 MB/s eta 0:00:00
Collecting distributed==2024.1.1 (from rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*)
  Downloading distributed-2024.1.1-py3-none-any.whl (1.0 MB)
                           1.0/1.0 MB 74.2 MB/s eta 0:00:00
Collecting dask-expr==0.4.0 (from rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*)
  Downloading dask_expr-0.4.0-py3-none-any.whl (161 kB)
                           161.7/161.7 kB 24.1 MB/s eta
0:00:00
Requirement already satisfied: cloudpickle>=1.5.0 in
/usr/local/lib/python3.10/dist-packages (from dask==2024.1.1->rapids-dask-
dependency==24.4.*->cuml-cu12==24.4.*) (2.2.1)
Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from dask==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (1.4.2)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-
packages (from dask==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (6.0.1)
Requirement already satisfied: toolz>=0.10.0 in /usr/local/lib/python3.10/dist-
packages (from dask==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (0.12.1)
Requirement already satisfied: importlib-metadata>=4.13.0 in
/usr/local/lib/python3.10/dist-packages (from dask==2024.1.1->rapids-dask-
dependency==24.4.*->cuml-cu12==24.4.*) (7.1.0)
Requirement already satisfied: jinja2>=2.10.3 in /usr/local/lib/python3.10/dist-
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (3.1.4)
Requirement already satisfied: locket>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (1.0.0)
Requirement already satisfied: msgpack>=1.0.0 in /usr/local/lib/python3.10/dist-
```

```
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (1.0.8)
Requirement already satisfied: psutil>=5.7.2 in /usr/local/lib/python3.10/dist-
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (5.9.5)
Requirement already satisfied: sortedcontainers>=2.0.5 in
/usr/local/lib/python3.10/dist-packages (from distributed==2024.1.1->rapids-
dask-dependency==24.4.*->cuml-cu12==24.4.*) (2.4.0)
Requirement already satisfied: tblib>=1.6.0 in /usr/local/lib/python3.10/dist-
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (3.0.0)
Requirement already satisfied: tornado>=6.0.4 in /usr/local/lib/python3.10/dist-
packages (from distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (6.3.3)
Requirement already satisfied: urllib3>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from distributed==2024.1.1->rapids-
dask-dependency==24.4.*->cuml-cu12==24.4.*) (2.0.7)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp) (4.0.3)
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist-
packages (from bokeh>=3.1->cuxfilter-cu12==24.4.*) (1.2.1)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-
packages (from bokeh>=3.1->cuxfilter-cu12==24.4.*) (9.4.0)
Requirement already satisfied: xyzservices>=2021.09.1 in
/usr/local/lib/python3.10/dist-packages (from bokeh>=3.1->cuxfilter-
cu12==24.4.*) (2024.4.0)
Requirement already satisfied: cython in /usr/local/lib/python3.10/dist-packages
(from cuda-python<13.0a0,>=12.0->pylibraft-cu12==24.4.*) (3.0.10)
Requirement already satisfied: fastrlock>=0.5 in /usr/local/lib/python3.10/dist-
packages (from cupy-cuda12x>=12.0.0->cuml-cu12==24.4.*) (0.8.2)
Requirement already satisfied: colorcet in /usr/local/lib/python3.10/dist-
packages (from datashader>=0.15->cuxfilter-cu12==24.4.*) (3.1.0)
Requirement already satisfied: multipledispatch in
/usr/local/lib/python3.10/dist-packages (from datashader>=0.15->cuxfilter-
cu12==24.4.*) (1.0.0)
Requirement already satisfied: param in /usr/local/lib/python3.10/dist-packages
(from datashader>=0.15->cuxfilter-cu12==24.4.*) (2.1.0)
Collecting pyct (from datashader>=0.15->cuxfilter-cu12==24.4.*)
```

```
  Downloading pyct-0.5.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from datashader>=0.15->cuxfilter-cu12==24.4.*) (2.31.0)
Requirement already satisfied: xarray in /usr/local/lib/python3.10/dist-packages
(from datashader>=0.15->cuxfilter-cu12==24.4.*) (2023.7.0)
Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-
packages (from geopandas>=0.11.0->cuspatial-cu12==24.4.*) (1.9.6)
Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-
packages (from geopandas>=0.11.0->cuspatial-cu12==24.4.*) (3.6.1)
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-
packages (from geopandas>=0.11.0->cuspatial-cu12==24.4.*) (2.0.4)
Requirement already satisfied: pyviz-comms>=0.7.4 in
/usr/local/lib/python3.10/dist-packages (from holoviews>=1.16.0->cuxfilter-
cu12==24.4.*) (3.0.2)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in
/usr/local/lib/python3.10/dist-packages (from numba>=0.57->cuml-cu12==24.4.*)
(0.41.1)
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-
packages (from panel>=1.0->cuxfilter-cu12==24.4.*) (3.6)
Requirement already satisfied: markdown-it-py in /usr/local/lib/python3.10/dist-
packages (from panel>=1.0->cuxfilter-cu12==24.4.*) (3.0.0)
Requirement already satisfied: linkify-it-py in /usr/local/lib/python3.10/dist-
packages (from panel>=1.0->cuxfilter-cu12==24.4.*) (2.0.3)
Requirement already satisfied: mdit-py-plugins in
/usr/local/lib/python3.10/dist-packages (from panel>=1.0->cuxfilter-
cu12==24.4.*) (0.4.1)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-
packages (from panel>=1.0->cuxfilter-cu12==24.4.*) (4.66.4)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from panel>=1.0->cuxfilter-cu12==24.4.*) (6.1.0)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-
packages (from scikit-image<0.23.0a0,>=0.19.0->cucim-cu12==24.4.*) (3.3)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-image<0.23.0a0,>=0.19.0->cucim-cu12==24.4.*) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image<0.23.0a0,>=0.19.0->cucim-cu12==24.4.*) (2024.5.10)
Requirement already satisfied: PyWavelets>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image<0.23.0a0,>=0.19.0->cucim-cu12==24.4.*) (1.6.0)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-
packages (from yarl<2.0,>=1.0->aiohttp) (3.7)
Requirement already satisfied: jupyter-server>=1.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (1.24.0)
Collecting simpervisor>=1.0 (from jupyter-server-proxy->cuxfilter-cu12==24.4.*)
  Downloading simpervisor-1.0.0-py3-none-any.whl (8.3 kB)
Requirement already satisfied: traitlets>=4.2.1 in
```

```
/usr/local/lib/python3.10/dist-packages (from jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (5.7.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from fiona>=1.8.19->geopandas>=0.11.0->cuspatial-cu12==24.4.*)
(2024.2.2)
Requirement already satisfied: click-plugins>=1.0 in
/usr/local/lib/python3.10/dist-packages (from
fiona>=1.8.19->geopandas>=0.11.0->cuspatial-cu12==24.4.*) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-
packages (from fiona>=1.8.19->geopandas>=0.11.0->cuspatial-cu12==24.4.*) (0.7.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from fiona>=1.8.19->geopandas>=0.11.0->cuspatial-cu12==24.4.*) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from
jinja2>=2.10.3->distributed==2024.1.1->rapids-dask-dependency==24.4.*->cuml-
cu12==24.4.*) (2.1.5)
Requirement already satisfied: anyio<4,>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (3.7.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-
packages (from jupyter-server>=1.0->jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (23.1.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (6.1.12)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (5.7.2)
Requirement already satisfied: nbconvert>=6.4.4 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (6.5.4)
Requirement already satisfied: nbformat>=5.2.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (5.10.4)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (0.20.0)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.10/dist-
packages (from jupyter-server>=1.0->jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (24.0.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.10/dist-
packages (from jupyter-server>=1.0->jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (0.18.1)
Requirement already satisfied: websocket-client in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.0->jupyter-
```

```
server-proxy->cuxfilter-cu12==24.4.*) (1.8.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas<2.2.2dev0,>=2.0->cudf-
cu12==24.4.*->cuml-cu12==24.4.*) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas<2.2.2dev0,>=2.0->cudf-cu12==24.4.*->cuml-cu12==24.4.*)
(2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas<2.2.2dev0,>=2.0->cudf-cu12==24.4.*->cuml-cu12==24.4.*)
(2024.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->panel>=1.0->cuxfilter-cu12==24.4.*) (0.5.1)
Requirement already satisfied: uc-micro-py in /usr/local/lib/python3.10/dist-
packages (from linkify-it-py->panel>=1.0->cuxfilter-cu12==24.4.*) (1.0.3)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py->panel>=1.0->cuxfilter-cu12==24.4.*) (0.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests->datashader>=0.15->cuxfilter-cu12==24.4.*) (3.3.2)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->cudf-cu12==24.4.*->cuml-
cu12==24.4.*) (2.16.1)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (1.2.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-
packages (from importlib-metadata>=4.13.0->dask==2024.1.1->rapids-dask-
dependency==24.4.*->cuml-cu12==24.4.*) (3.18.2)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
core!=5.0.*,>=4.12->jupyter-server>=1.0->jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (4.2.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from nbconvert>=6.4.4->jupyter-server>=1.0->jupyter-server-proxy->cuxfilter-
cu12==24.4.*) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=6.4.4->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (4.12.3)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=6.4.4->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=6.4.4->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (0.4)
Requirement already satisfied: jupyterlab-pygments in
```

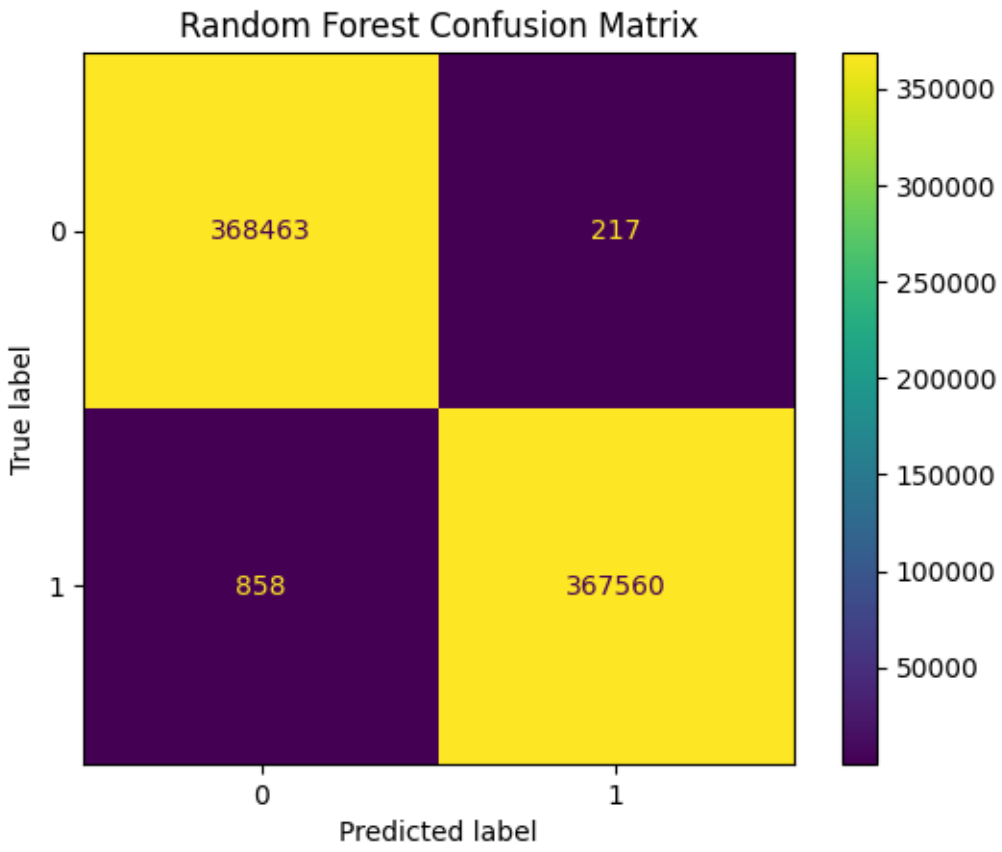
```
/usr/local/lib/python3.10/dist-packages (from nbconvert>=6.4.4->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (0.3.0)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=6.4.4->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=6.4.4->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (0.10.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=6.4.4->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=6.4.4->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (1.3.0)
Requirement already satisfied: fastjsonschema>=2.15 in
/usr/local/lib/python3.10/dist-packages (from nbformat>=5.2.0->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (2.19.1)
Requirement already satisfied: jsonschema>=2.6 in
/usr/local/lib/python3.10/dist-packages (from nbformat>=5.2.0->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (4.19.2)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.10/dist-
packages (from terminado>=0.8.3->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (0.7.0)
Requirement already satisfied: argon2-cffi-bindings in
/usr/local/lib/python3.10/dist-packages (from argon2-cffi->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (21.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.2.0->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.2.0->jupyter-server>=1.0->jupyter-server-
proxy->cuxfilter-cu12==24.4.*) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat>=5.2.0->jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (0.18.1)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-
packages (from argon2-cffi-bindings->argon2-cffi->jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (1.16.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->nbconvert>=6.4.4->jupyter-server>=1.0->jupyter-
server-proxy->cuxfilter-cu12==24.4.*) (2.5)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->jupyter-
server>=1.0->jupyter-server-proxy->cuxfilter-cu12==24.4.*) (2.22)
Installing collected packages: simpervisor, pynvml, pyct, ucx-py-cu12, treelite,
dask, pylibraft-cu12, distributed, dask-expr, cuproj-cu12, cucim-cu12, rapids-
```

```
dask-dependency, pylibcugraph-cu12, datashader, cuspatial-cu12, dask-cudf-cu12,
dask-cuda, raft-dask-cu12, cuml-cu12, cugraph-cu12, jupyter-server-proxy,
cuxfilter-cu12
  Attempting uninstall: pynvml
    Found existing installation: pynvml 11.5.0
    Uninstalling pynvml-11.5.0:
      Successfully uninstalled pynvml-11.5.0
  Attempting uninstall: dask
    Found existing installation: dask 2023.8.1
    Uninstalling dask-2023.8.1:
      Successfully uninstalled dask-2023.8.1
  Attempting uninstall: distributed
    Found existing installation: distributed 2023.8.1
    Uninstalling distributed-2023.8.1:
      Successfully uninstalled distributed-2023.8.1
Successfully installed cucim-cu12-24.4.0 cugraph-cu12-24.4.0 cuml-cu12-24.4.0
cuproj-cu12-24.4.0 cuspatial-cu12-24.4.0 cuxfilter-cu12-24.4.1 dask-2024.1.1
dask-cuda-24.4.0 dask-cudf-cu12-24.4.1 dask-expr-0.4.0 datashader-0.16.1
distributed-2024.1.1 jupyter-server-proxy-4.1.2 pyct-0.5.0 pylibcugraph-
cu12-24.4.0 pylibraft-cu12-24.4.0 pynvml-11.4.1 raft-dask-cu12-24.4.0 rapids-
dask-dependency-24.4.1 simpervisor-1.0.0 treelite-4.1.2 ucx-py-cu12-0.37.0

        ***********************************************************************
        The pip install of RAPIDS is complete.

        Please do not run any further installation from the conda based
installation methods, as they may cause issues!

        Please ensure that you're pulling from the git repo to remain updated
with the latest working install scripts.

        Troubleshooting:
            - If there is an installation failure, please check back on RAPIDSAI
owned templates/notebooks to see how to update your personal files.
            - If an installation failure persists when using the latest script,
please make an issue on https://github.com/rapidsai-community/rapidsai-csp-utils
        ***********************************************************************
```

[ ]:
```python
import cuml
from cuml.ensemble import RandomForestClassifier as cuRF
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
 ↪f1_score, accuracy_score, confusion_matrix, ConfusionMatrixDisplay

# Inicializar el modelo Random Forest de cuML
rf_model = cuRF(
    n_estimators=200,
```

```
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=2,
    max_features='sqrt',
    random_state=42
)

# Entrenamiento del modelo
rf_model.fit(X_train, y_train)

# Predicción y evaluación del modelo Random Forest
y_pred_rf_proba = rf_model.predict_proba(X_test).to_numpy()[:, 1]
y_pred_rf = rf_model.predict(X_test)

rf_roc_auc = roc_auc_score(y_test, y_pred_rf_proba)
rf_precision = precision_score(y_test, y_pred_rf)
rf_recall = recall_score(y_test, y_pred_rf)
rf_f1 = f1_score(y_test, y_pred_rf)
rf_accuracy = accuracy_score(y_test, y_pred_rf)

rf_metrics = {
    'ROC-AUC': rf_roc_auc,
    'Precision': rf_precision,
    'Recall': rf_recall,
    'F1 Score': rf_f1,
    'Accuracy': rf_accuracy
}

print("Random Forest Metrics:", rf_metrics)

# Matriz de Confusión
rf_cm = confusion_matrix(y_test, y_pred_rf)
ConfusionMatrixDisplay(rf_cm).plot()
plt.title("Random Forest Confusion Matrix")
plt.show()
```

```
Random Forest Metrics: {'ROC-AUC': 0.9999898572277597, 'Precision':
0.999409968540719, 'Recall': 0.9976711235607381, 'F1 Score': 0.9985397890504554,
'Accuracy': 0.998541577917726}
```



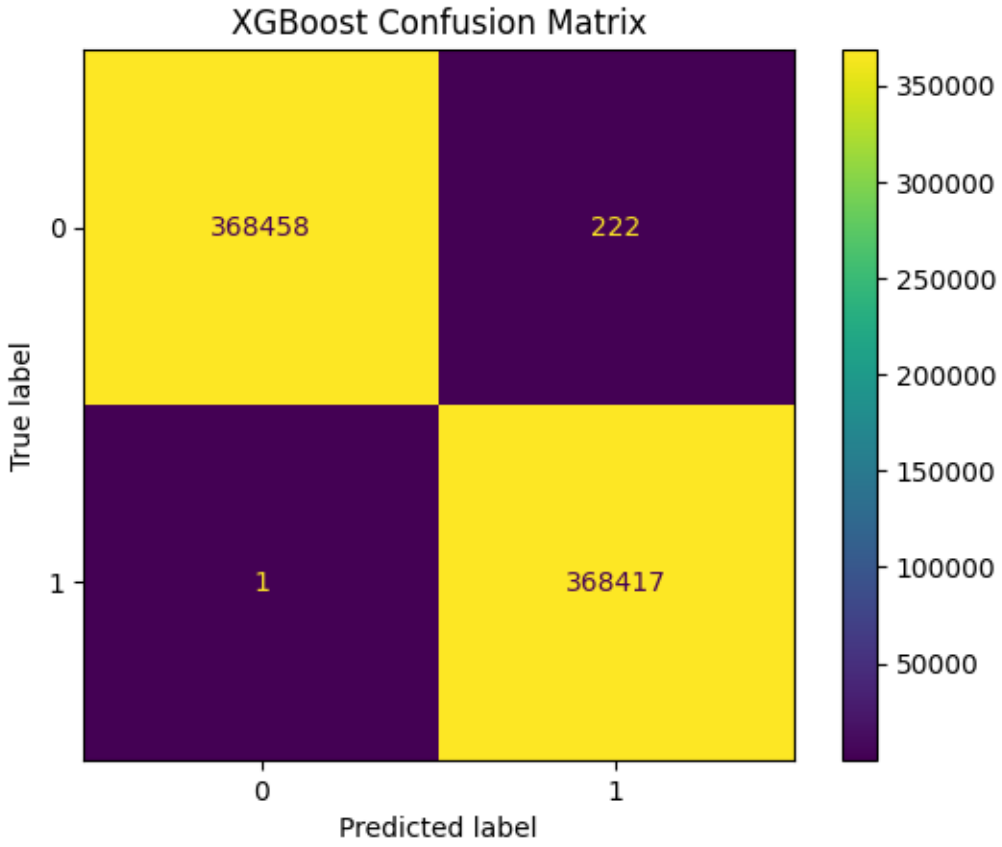
[ ]:
```python
import xgboost as xgb

# Inicializar el modelo XGBoost con GPU
params = {
    'objective': 'binary:logistic',
    'tree_method': 'gpu_hist',
    'eval_metric': 'auc',
    'learning_rate': 0.05,
    'max_depth': 6,
    'min_child_weight': 3,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'gamma': 0.1,
    'alpha': 0.1,
    'lambda': 1,
    'random_state': 42
}
xgb_model = xgb.XGBClassifier(**params)

# Entrenamiento del modelo
```

```
xgb_model.fit(X_train, y_train)

# Predicción y evaluación del modelo XGBoost
y_pred_xgb = xgb_model.predict(X_test)
y_pred_xgb_proba = xgb_model.predict_proba(X_test)[:, 1]

xgb_roc_auc = roc_auc_score(y_test, y_pred_xgb_proba)
xgb_precision = precision_score(y_test, y_pred_xgb)
xgb_recall = recall_score(y_test, y_pred_xgb)
xgb_f1 = f1_score(y_test, y_pred_xgb)
xgb_accuracy = accuracy_score(y_test, y_pred_xgb)

xgb_metrics = {
    'ROC-AUC': xgb_roc_auc,
    'Precision': xgb_precision,
    'Recall': xgb_recall,
    'F1 Score': xgb_f1,
    'Accuracy': xgb_accuracy
}

print("XGBoost Metrics:", xgb_metrics)

# Matriz de Confusión
xgb_cm = confusion_matrix(y_test, y_pred_xgb)
ConfusionMatrixDisplay(xgb_cm).plot()
plt.title("XGBoost Confusion Matrix")
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[06:23:38] WARNING: /workspace/src/common/error_msg.cc:27: The tree method
`gpu_hist` is deprecated since 2.0.0. To use GPU training, set the `device`
parameter to CUDA instead.

    E.g. tree_method = "hist", device = "cuda"

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[06:23:40] WARNING: /workspace/src/common/error_msg.cc:27: The tree method
`gpu_hist` is deprecated since 2.0.0. To use GPU training, set the `device`
parameter to CUDA instead.

    E.g. tree_method = "hist", device = "cuda"

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[06:23:40] WARNING: /workspace/src/common/error_msg.cc:58: Falling back to
prediction using DMatrix due to mismatched devices. This might lead to higher
```

```
memory usage and slower performance. XGBoost is running on: cuda:0, while the
input data is on: cpu.
Potential solutions:
- Use a data structure that matches the device ordinal in the booster.
- Set the device for booster before call to inplace_predict.

This warning will only be shown once.

  warnings.warn(smsg, UserWarning)
```

```
XGBoost Metrics: {'ROC-AUC': 0.9999964878502449, 'Precision':
0.9993977848247201, 'Recall': 0.9999972856917957, 'F1 Score':
0.9996974453807507, 'Accuracy': 0.9996974622099096}
```



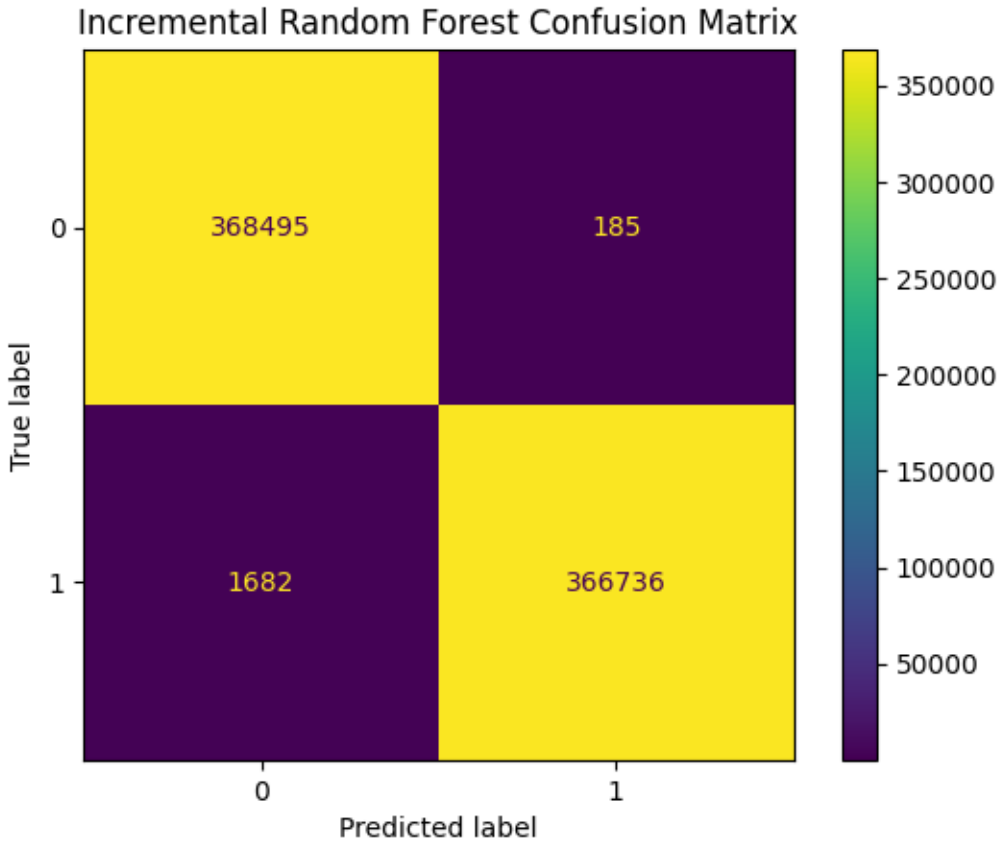
0.2.7 Entrenamiento Incremental de los Modelos

[ ]:
```
# Inicializar el modelo Random Forest de cuML
rf_model = cuRF(
    n_estimators=200,
    max_depth=10,
```

```
    min_samples_split=5,
    min_samples_leaf=2,
    max_features='sqrt',
    random_state=42
)

# Entrenamiento incremental manual por año
years = sorted(X_train['Transaction_year'].unique())
for year in years:
    # Filtrar los datos por año
    X_train_year = X_train[X_train['Transaction_year'] == year]
    y_train_year = y_train.loc[X_train_year.index]

    # Entrenar el modelo con los datos del año actual
    rf_model.fit(X_train_year.drop(columns=['Transaction_year']), y_train_year)

# Predicción y evaluación del modelo Random Forest
y_pred_rf_proba = rf_model.predict_proba(X_test.
 ↪drop(columns=['Transaction_year'])).to_numpy()[:, 1]
y_pred_rf = rf_model.predict(X_test.drop(columns=['Transaction_year']))

rf_roc_auc = roc_auc_score(y_test, y_pred_rf_proba)
rf_precision = precision_score(y_test, y_pred_rf)
rf_recall = recall_score(y_test, y_pred_rf)
rf_f1 = f1_score(y_test, y_pred_rf)
rf_accuracy = accuracy_score(y_test, y_pred_rf)

rf_metrics = {
    'ROC-AUC': rf_roc_auc,
    'Precision': rf_precision,
    'Recall': rf_recall,
    'F1 Score': rf_f1,
    'Accuracy': rf_accuracy
}

print("Incremental Random Forest Metrics:", rf_metrics)

# Matriz de Confusión
rf_cm = confusion_matrix(y_test, y_pred_rf)
ConfusionMatrixDisplay(rf_cm).plot()
plt.title("Incremental Random Forest Confusion Matrix")
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/cuml/internals/api_decorators.py:344:
UserWarning: For reproducible results in Random Forest Classifier or for almost
reproducible results in Random Forest Regressor, n_streams=1 is recommended. If
n_streams is > 1, results may vary due to stream/thread timing differences, even
```

```
when random_state is set
  return func(**kwargs)
/usr/local/lib/python3.10/dist-packages/cuml/internals/api_decorators.py:188:
UserWarning: To use pickling first train using float32 data to fit the estimator
  ret = func(*args, **kwargs)
/usr/local/lib/python3.10/dist-packages/cuml/internals/api_decorators.py:188:
UserWarning: To use pickling first train using float32 data to fit the estimator
  ret = func(*args, **kwargs)

Incremental Random Forest Metrics: {'ROC-AUC': 0.9999877081329369, 'Precision':
0.9994958042739446, 'Recall': 0.9954345336004212, 'F1 Score':
0.9974610349784249, 'Accuracy': 0.9974670939278087}
```



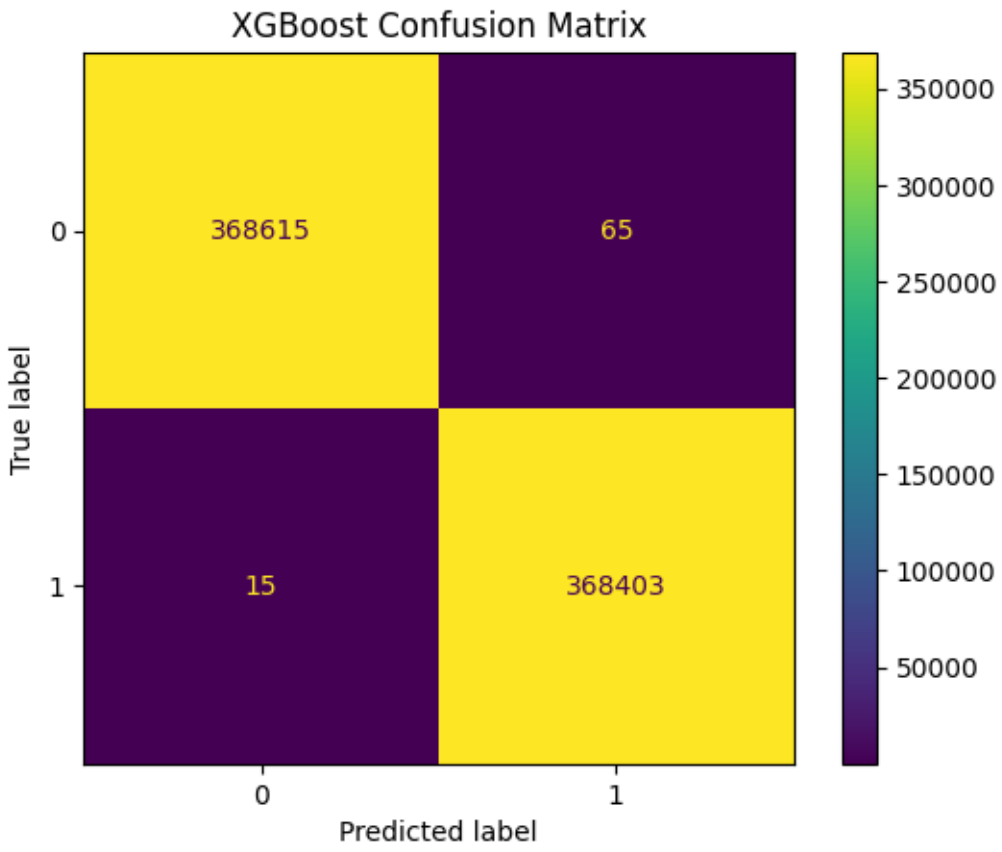
[ ]:
```python
# Inicializar los parámetros del modelo XGBoost
params = {
    'objective': 'binary:logistic',
    'tree_method': 'gpu_hist',
    'eval_metric': 'auc',
    'learning_rate': 0.05,
    'max_depth': 6,
    'min_child_weight': 3,
```

```
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'gamma': 0.1,
    'alpha': 0.1,
    'lambda': 1,
    'random_state': 42
}

# Inicializar el modelo XGBoost
xgb_model = None

# Entrenamiento incremental manual por año
years = sorted(X_train['Transaction_year'].unique())
for year in years:
    # Filtrar los datos por año
    X_train_year = X_train[X_train['Transaction_year'] == year]
    y_train_year = y_train.loc[X_train_year.index]

    dtrain = xgb.DMatrix(X_train_year.drop(columns=['Transaction_year']),␣
 ↪label=y_train_year)

    # Entrenar el modelo con los datos del año actual
    if xgb_model is None:
        xgb_model = xgb.train(params, dtrain, num_boost_round=100)
    else:
        xgb_model = xgb.train(params, dtrain, num_boost_round=100,␣
 ↪xgb_model=xgb_model)

# Predicción y evaluación del modelo XGBoost
dtest = xgb.DMatrix(X_test.drop(columns=['Transaction_year']))
y_pred_xgb_proba = xgb_model.predict(dtest)
y_pred_xgb = (y_pred_xgb_proba > 0.5).astype(int)

xgb_roc_auc = roc_auc_score(y_test, y_pred_xgb_proba)
xgb_precision = precision_score(y_test, y_pred_xgb)
xgb_recall = recall_score(y_test, y_pred_xgb)
xgb_f1 = f1_score(y_test, y_pred_xgb)
xgb_accuracy = accuracy_score(y_test, y_pred_xgb)

xgb_metrics = {
    'ROC-AUC': xgb_roc_auc,
    'Precision': xgb_precision,
    'Recall': xgb_recall,
    'F1 Score': xgb_f1,
    'Accuracy': xgb_accuracy
}
```

```
print("XGBoost Metrics:", xgb_metrics)

# Matriz de Confusión
xgb_cm = confusion_matrix(y_test, y_pred_xgb)
ConfusionMatrixDisplay(xgb_cm).plot()
plt.title("XGBoost Confusion Matrix")
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[07:47:14] WARNING: /workspace/src/common/error_msg.cc:27: The tree method
`gpu_hist` is deprecated since 2.0.0. To use GPU training, set the `device`
parameter to CUDA instead.

    E.g. tree_method = "hist", device = "cuda"

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[07:47:18] WARNING: /workspace/src/common/error_msg.cc:27: The tree method
`gpu_hist` is deprecated since 2.0.0. To use GPU training, set the `device`
parameter to CUDA instead.

    E.g. tree_method = "hist", device = "cuda"

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:160: UserWarning:
[07:47:20] WARNING: /workspace/src/common/error_msg.cc:27: The tree method
`gpu_hist` is deprecated since 2.0.0. To use GPU training, set the `device`
parameter to CUDA instead.

    E.g. tree_method = "hist", device = "cuda"

  warnings.warn(smsg, UserWarning)
```

```
XGBoost Metrics: {'ROC-AUC': 0.9999994098507341, 'Precision':
0.9998235939077478, 'Recall': 0.999959285376936, 'F1 Score': 0.9998914350387984,
'Accuracy': 0.9998914662636448}
```


XGBoost Confusion Matrix
368615
65
15
368403
0
1
True label
0
1
Predicted label
350000
300000
250000
200000
150000
100000
50000