🚨 Fraud Detection – Data Cleaning & Exploratory Data Analysis (EDA)

In this notebook, we aim to prepare our data for fraud detection modeling by:

- Cleaning and preprocessing the data
- Visualizing key patterns
- Deriving insights for stakeholders

```
In [3]:  import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns
         from sklearn.model_selection import train_test_split
```

📃 Data Overview

The dataset contains over **6.3 million transactions** and **10 columns**, each representing transaction details like amount, type, account balances, and fraud labels.

```
In [2]:  df = pd.read_csv("../Raw data/Fraud.csv")
         print(df.head())
         print(df.shape)
```

```
   step      type     amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT    9839.64  C1231006815       170136.0        160296.36
1     1   PAYMENT    1864.28  C1666544295        21249.0         19384.72
2     1  TRANSFER     181.00  C1305486145          181.0             0.00
3     1  CASH_OUT     181.00   C840083671          181.0             0.00
4     1   PAYMENT   11668.14  C2048537720        41554.0         29885.86

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0  M1979787155             0.0             0.0        0               0
1  M2044282225             0.0             0.0        0               0
2   C553264065             0.0             0.0        1               0
3    C38997010         21182.0             0.0        1               0
4  M1230701703             0.0             0.0        0               0
(6362620, 11)
```

🧹 Missing Values & Duplicates

- ✅ There are **no missing values**.
- ✅ There are **no duplicate records**.

```
In [3]:  # Check for missing values
         df.isnull().sum()
```

```
Out[3]:  step              0
         type              0
         amount            0
         nameOrig          0
         oldbalanceOrg     0
         newbalanceOrig    0
         nameDest          0
         oldbalanceDest    0
         newbalanceDest    0
         isFraud           0
         isFlaggedFraud    0
         dtype: int64
```

```
In [4]:  # Check for duplicates
         df.duplicated().sum()
```

```
Out[4]:  0
```

⚠️ ID Columns Dropped

- Columns like `nameOrig` and `nameDest` are just customer identifiers and not useful for modeling.
- We've removed them to reduce noise.

```
In [5]:  df.drop(['nameOrig', 'nameDest'], axis=1, inplace=True)
```

📉 Negative Balances Cleaned

Some transactions showed negative account balances — logically impossible. These values were cleaned by setting them to **0**.

```
In [6]:  balance_cols = ['oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceD

         for col in balance_cols:
             neg_count = df[df[col] < 0].shape[0]
             print(f"{col} has {neg_count} negative values.")
             df[col] = df[col].apply(lambda x: max(x, 0))
```

```
oldbalanceOrg has 0 negative values.
newbalanceOrig has 0 negative values.
oldbalanceDest has 0 negative values.
newbalanceDest has 0 negative values.
```

🧠 Feature Engineering

We added:

- `balance_diff_orig` : Difference in sender's balance
- `balance_diff_dest` : Difference in receiver's balance

These features help models capture transactional patterns better.

```
In [7]:  df['balance_diff_orig'] = df['oldbalanceOrg'] - df['newbalanceOrig']
         df['balance_diff_dest'] = df['newbalanceDest'] - df['oldbalanceDest']
```

🏷️ Encoding Transaction Types

The `type` column (CASH-IN, TRANSFER, etc.) was converted to numerical format using **one-hot encoding**, making it usable for ML models.

```
In [8]:  df = pd.get_dummies(df, columns=['type'], drop_first=True)
```

💾 Cleaned Data Saved

The cleaned and preprocessed data is saved for modeling as `cleaned.csv` .

```
In [17]:  df.to_csv("../Cleaned data/cleaned.csv", index=False)
```
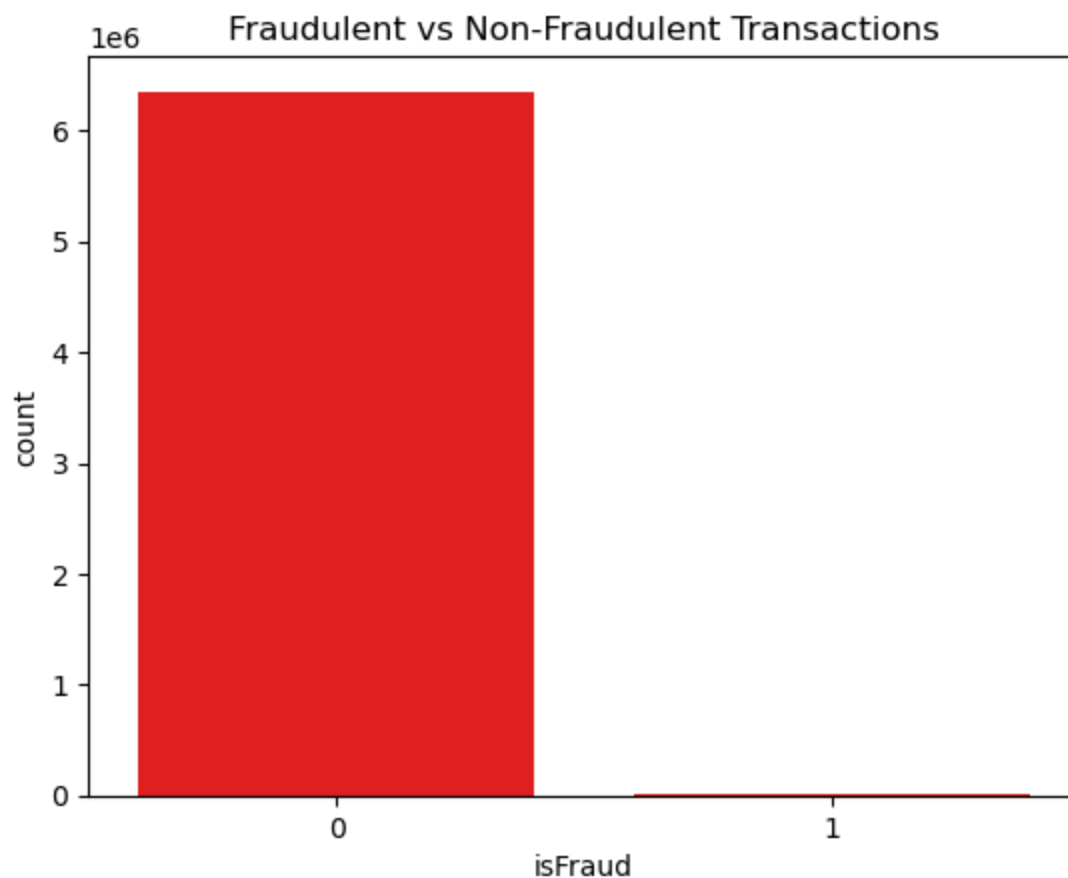
🔍 Class Imbalance

There is a **significant class imbalance**:

- Majority of transactions are legitimate (0)
- Only a small portion are fraud (1)

We'll handle this imbalance during model training using resampling or weighted algorithms.

```
In [10]:  sns.countplot(x='isFraud', data=df , color='red')
          plt.title("Fraudulent vs Non-Fraudulent Transactions")
          plt.show()
```

💸 Distribution of Transaction Amounts

This histogram reveals several key insights:

- **Highly right-skewed distribution**: Most transactions are of relatively **small amounts**, while a few transactions involve **very large sums** (outliers).
- **Majority of transactions** fall under a specific amount range, likely routine or common customer activity.
- The **long tail** suggests that high-value transactions are rare but may require **special monitoring** — since fraudulent activity often happens in **large single transfers**.

🔎 **Business Insight**:

We should pay **closer attention to high-value transactions** as they have a higher likelihood of fraud. We'll incorporate this observation when selecting features for our model.

```
In [11]: # Distribution of amount
         plt.figure(figsize=(8, 5))
         sns.histplot(data=df, x='amount', bins=100 , color='blue', kde=True)
         plt.title("Transaction Amount Distribution")
         plt.show()
```

## Transaction Amount Distribution
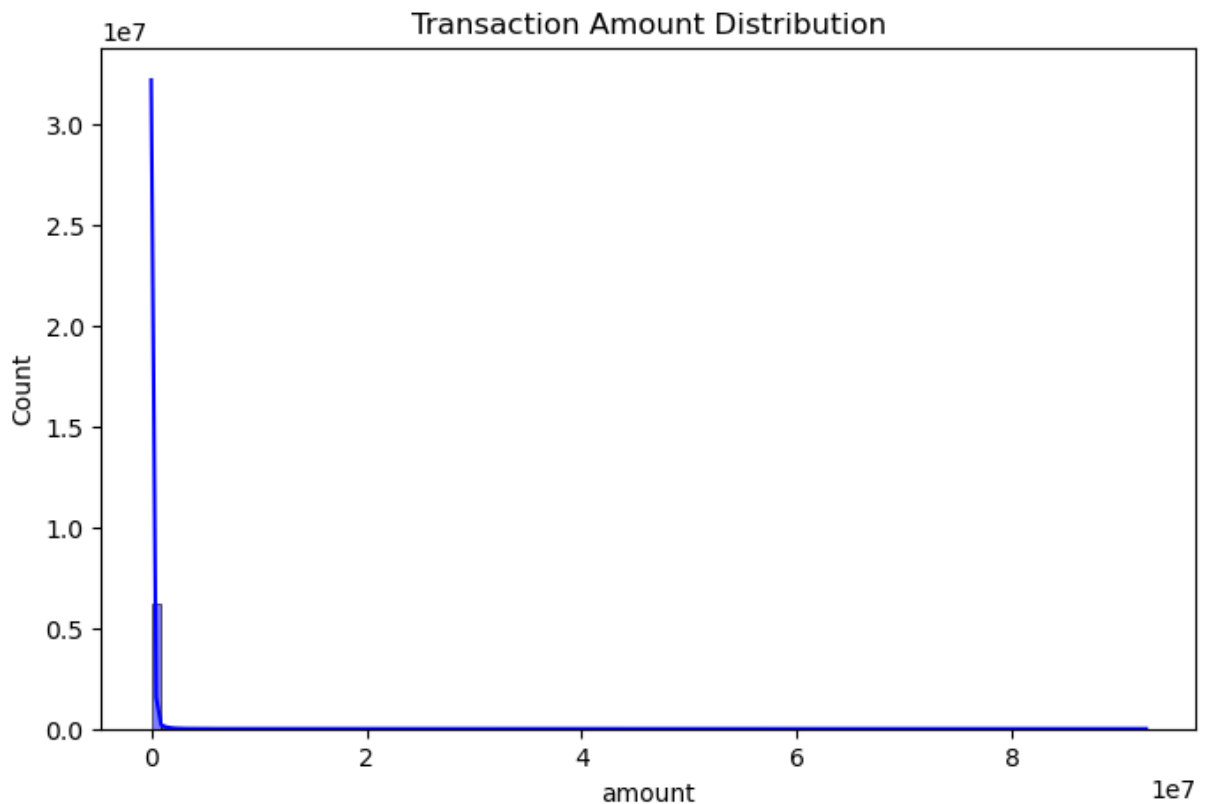
🔗 Correlation Heatmap

The heatmap above helps identify relationships between variables. Key findings:

- `oldbalanceOrg` and `newbalanceOrig` are **strongly positively correlated**, which is expected — they represent balances before and after the same transaction.
- Similarly, `oldbalanceDest` and `newbalanceDest` also show **strong correlation**.
- `amount` shows **moderate positive correlation** with `balance_diff_orig` — which makes sense, since the amount contributes to the drop in sender balance.
- `isFraud` has **low correlation with most individual features**, indicating that fraud detection might depend on **interactions between features** rather than simple thresholds.

📌 **Business Insight**:
Simple rules (like high amount = fraud) may not be sufficient. We'll need **machine learning models** that can capture **non-linear patterns and feature combinations** to detect fraud effectively.

```
In [12]:   # Correlation heatmap
           plt.figure(figsize=(12, 8))
           sns.heatmap(df.corr(), annot=False, cmap='coolwarm')
           plt.title("Correlation Heatmap")
           plt.show()
```

Correlation Heatmap
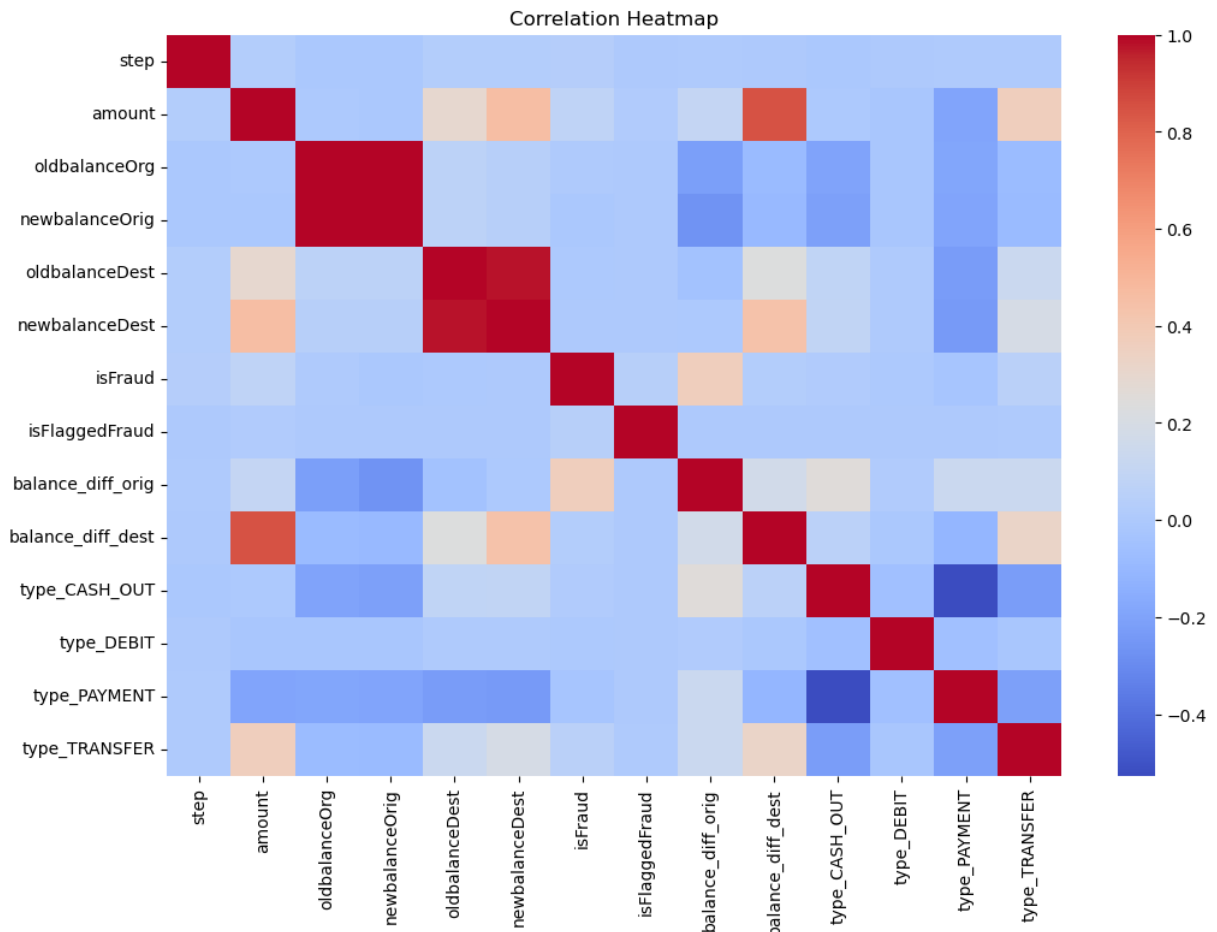
📌 Business Insight

- Fraud rates are **highest for** `TRANSFER` **and** `CASH_OUT` transactions.
- These two types are used in **most fraud loops**: money is transferred from a hijacked account and quickly withdrawn.

📈 Action Plan

- Apply stricter fraud controls for these types.
- Use **multi-factor authentication** for large transfer/cash-out.
- Monitor for sequences: `TRANSFER → CASH_OUT` within short time windows.

```
In [21]:   # Melt type columns to single column for analysis
           type_cols = ['type_CASH_OUT', 'type_DEBIT', 'type_PAYMENT', 'type_TRANSFER']
           type_df = pd.melt(df, id_vars='isFraud', value_vars=type_cols, var_name='Transact

           # Keep only rows where that transaction type is 1
           type_df = type_df[type_df['value'] == 1]

           # Calculate fraud rate per type
           fraud_rate_per_type = type_df.groupby('TransactionType')['isFraud'].mean().reset_

           # Plot
           import seaborn as sns
```

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.barplot(data=fraud_rate_per_type, x='TransactionType', y='isFraud',color='ora
plt.title("🔍 Fraud Rate by Transaction Type")
plt.ylabel("Fraud Rate")
plt.xlabel("Transaction Type")
plt.show()
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWa
rning: Glyph 128269 (\N{LEFT-POINTING MAGNIFYING GLASS}) missing from font(s) Deja
Vu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```
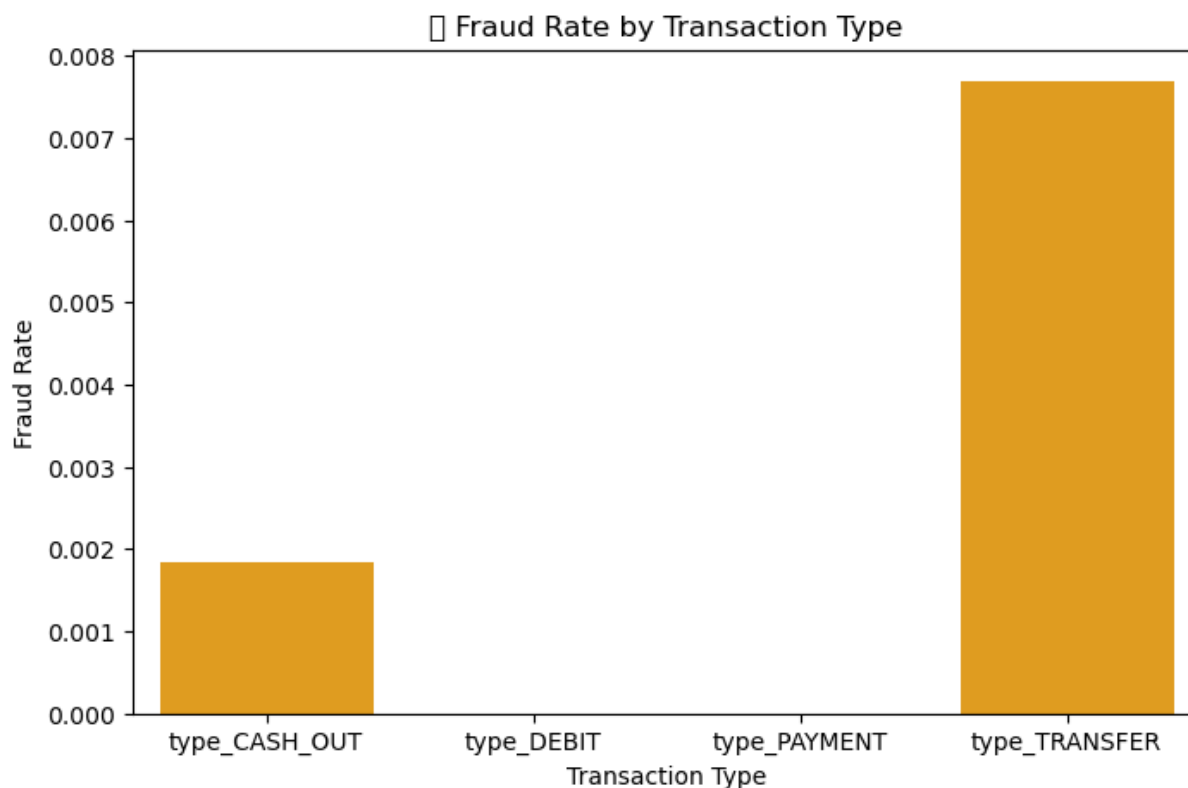

📌 Business Insight

- Fraud is **not randomly distributed** — it peaks during specific time steps.
- Attackers may target **low surveillance windows**, like nights/weekends.

📈 Action Plan

- Deploy **adaptive monitoring** with increased alerts during risky time bands.
- Shift fraud detection team hours to **cover peak windows**.

```python
In [22]: plt.figure(figsize=(10, 5))
         sns.histplot(data=df[df['isFraud'] == 1], x='step', bins=50, color='red', label='
         sns.histplot(data=df[df['isFraud'] == 0], x='step', bins=50, color='blue', label=
         plt.title("⏱ Transactions Over Time (step)")
         plt.xlabel("Time Step (1 Step = 1 Hour)")
```

```
plt.ylabel("Transaction Count")
plt.legend()
plt.show()
```

c:\Users\yasht_\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWa
rning: Glyph 9201 (\N{STOPWATCH}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
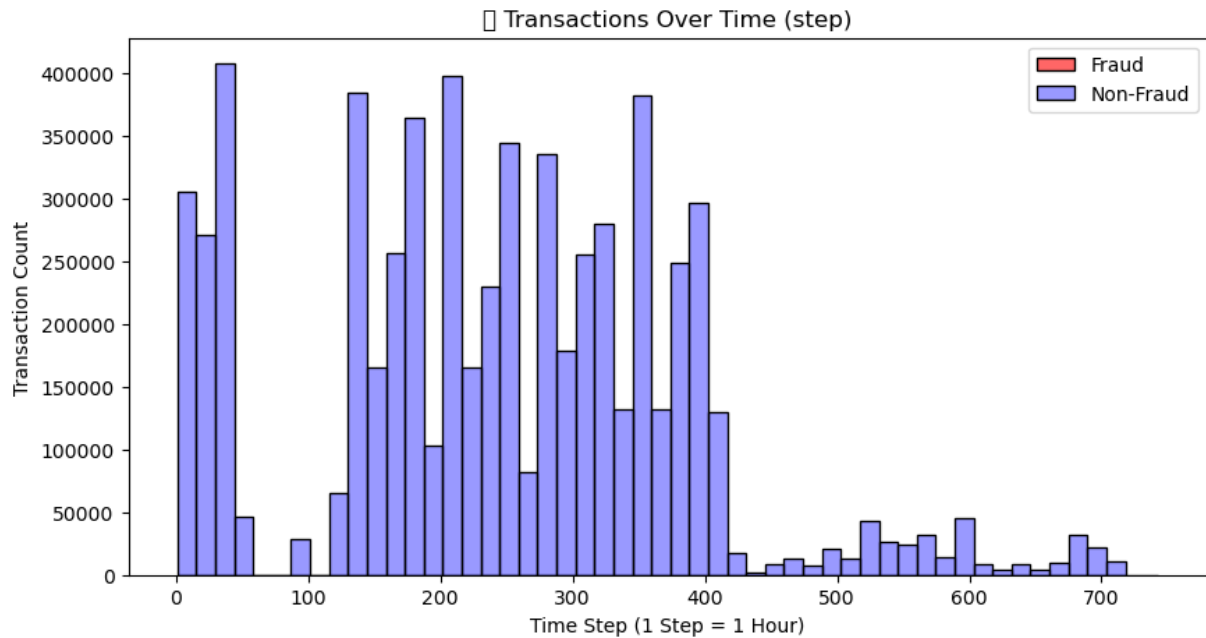
📌 Business Insight

- The `isFlaggedFraud` feature, which flags transfers above 200K, **misses almost all actual fraud**.
- It only identifies a tiny fraction of total fraud cases.

📈 Action Plan

- Replace static thresholds with **machine learning-based scoring models**.
- Update fraud flagging rules using historical patterns.

In [23]:
```
sns.countplot(data=df, x='isFlaggedFraud', hue='isFraud')
plt.title("🚨 Flagged vs Actual Fraud")
plt.xlabel("Is Flagged Fraud?")
plt.ylabel("Count")
plt.legend(title="Is Fraud")
plt.show()
```

c:\Users\yasht_\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWa
rning: Glyph 128680 (\N{POLICE CARS REVOLVING LIGHT}) missing from font(s) DejaVu
Sans.
  fig.canvas.print_figure(bytes_io, **kw)
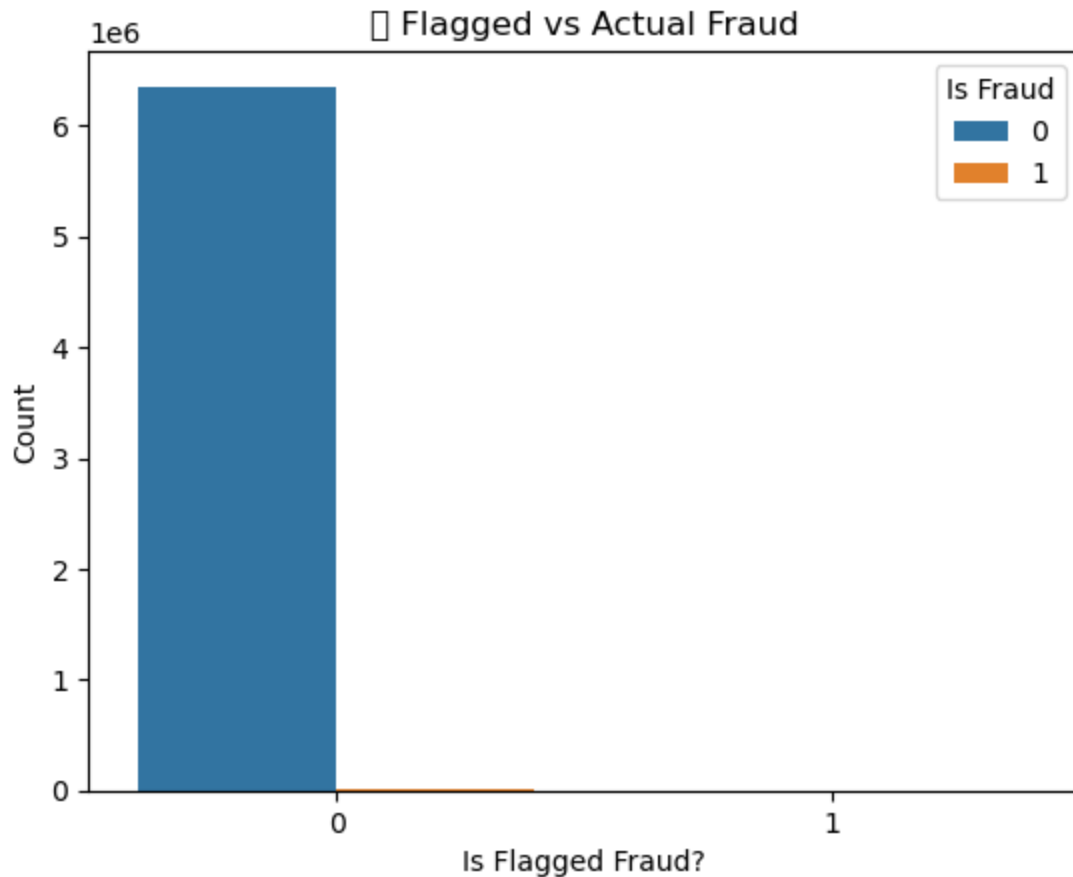
📌 Business Insight

- No single feature strongly predicts fraud — most correlations are weak.
- This confirms fraud detection is a **multi-variable, non-linear problem**.

📈 Action Plan

- Use advanced models like **XGBoost or Random Forest** to capture complex patterns.
- Apply **SHAP values or feature importance** to interpret model outputs.

```
In [25]:  corr = df.corr()['isFraud'].sort_values(ascending=False)

          plt.figure(figsize=(8,6))
          sns.barplot(x=corr.values, y=corr.index ,color='purple')
          plt.title(" 📊 Correlation of Features with Fraud")
          plt.xlabel("Correlation")
          plt.show()
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWa
rning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```

Correlation of Features with Fraud
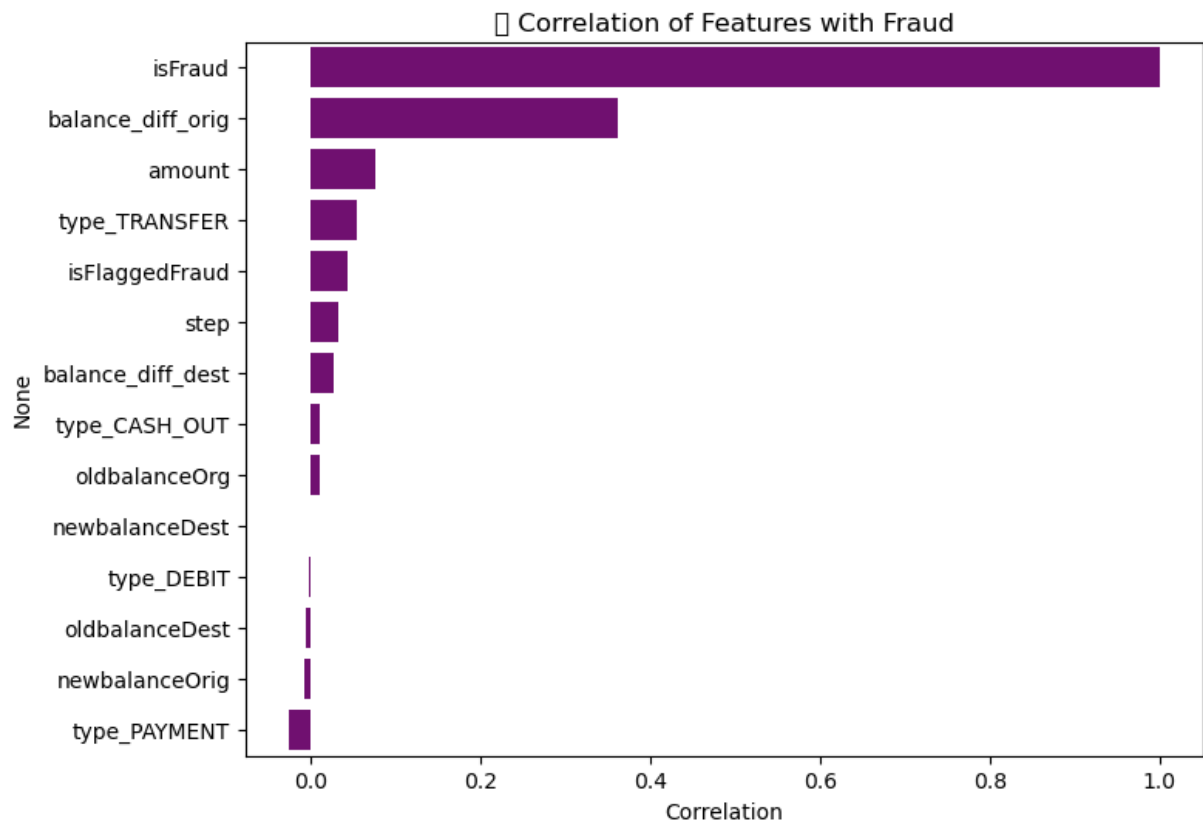
🧪 Training Multiple Machine Learning Models

Now that our dataset is cleaned and preprocessed, we will train and evaluate multiple machine learning models to find the best-performing one for fraud detection.

We will train the following models:

1. Logistic Regression
2. Random Forest
3. Gradient Boosting
4. AdaBoost
5. Naive Bayes
6. K-Nearest Neighbors
7. Support Vector Machine
8. XGBoost

We'll evaluate them based on **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **AUC-ROC**. The model with the best balance of **Recall** and **AUC-ROC** will be considered for hyperparameter tuning.

```
In [4]:   df = pd.read_csv("../Cleaned data/cleaned.csv")

          X = df.drop(['isFraud'], axis=1)
          y = df['isFraud']
```

📊 Splitting the Dataset

To evaluate our model's performance on unseen data, we split the data into training and testing sets using an 80/20 split with stratification to maintain class balance.

```
In [5]:  X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, stratif
```

```
In [14]:  from xgboost import XGBClassifier
          from sklearn.model_selection import RandomizedSearchCV
          from sklearn.metrics import classification_report, roc_auc_score
          import numpy as np

          # Base model with GPU enabled
          xgb_gpu = XGBClassifier(
              tree_method='gpu_hist',
              predictor='gpu_predictor',
              use_label_encoder=False,
              eval_metric='logloss',
              random_state=42
          )

          # Hyperparameter grid
          param_dist = {
              'n_estimators': [100, 200, 300],
              'max_depth': [4, 6, 8, 10],
              'learning_rate': [0.01, 0.1, 0.2],
              'subsample': [0.6, 0.8, 1.0],
              'colsample_bytree': [0.6, 0.8, 1.0]
          }

          # Randomized search
          random_search = RandomizedSearchCV(
              estimator=xgb_gpu,
              param_distributions=param_dist,
              n_iter=10,
              scoring='roc_auc',
              cv=3,
              verbose=2,
              random_state=42,
              n_jobs=1  # ❗ Single-threaded to avoid pickling error with GPU
          )

          random_search.fit(X_train, y_train)
          print("✅ Best Parameters:", random_search.best_params_)
```

```
Fitting 3 folds for each of 10 candidates, totalling 30 fits
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:11:41] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:11:41] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:12:37] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:729: UserWarning: [1
9:12:37] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:58: Falling back to prediction using DMatrix due to mismatched devices. This mig
ht 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.

  return func(**kwargs)
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=300,
subsample=0.6; total time= 1.0min
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:12:41] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:12:41] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:12:49] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=300,
subsample=0.6; total time=   11.7s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:12:53] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:12:53] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:01] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=300,
subsample=0.6; total time=   11.9s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:05] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:05] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:07] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=4, n_estimators=100,
subsample=0.6; total time=   5.6s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:11] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:11] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:13] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=4, n_estimators=100,
subsample=0.6; total time=   5.4s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:16] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:16] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:18] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=4, n_estimators=100,
subsample=0.6; total time=    4.7s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:21] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:21] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:23] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=10, n_estimators=100,
subsample=1.0; total time=    5.6s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:26] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:26] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:29] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=10, n_estimators=100,
subsample=1.0; total time=   5.7s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:33] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:33] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:36] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=10, n_estimators=100,
subsample=1.0; total time=   6.2s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:39] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:39] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:41] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.6, learning_rate=0.01, max_depth=6, n_estimators=100,
subsample=0.6; total time=   5.9s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:45] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:45] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:47] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.6, learning_rate=0.01, max_depth=6, n_estimators=100,
subsample=0.6; total time=   5.7s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:50] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:50] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:53] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.6, learning_rate=0.01, max_depth=6, n_estimators=100,
subsample=0.6; total time=    5.1s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:56] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:13:56] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:13:58] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=    4.9s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:01] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:01] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:03] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=    5.1s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:06] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:06] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:08] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=    5.1s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:11] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:11] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:14] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=100,
subsample=0.6; total time=   5.7s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:17] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:17] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:20] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=100,
subsample=0.6; total time=   5.4s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:23] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:23] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:25] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=8, n_estimators=100,
subsample=0.6; total time=   5.5s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:28] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:28] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:32] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=4, n_estimators=200,
subsample=0.6; total time=   6.6s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:35] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:35] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:39] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=4, n_estimators=200,
subsample=0.6; total time=   6.8s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:42] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:42] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:46] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=4, n_estimators=200,
subsample=0.6; total time=   6.7s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:49] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:49] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:14:55] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=6, n_estimators=300, s
ubsample=0.8; total time=   9.6s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:59] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:14:59] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:06] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=6, n_estimators=300, s
ubsample=0.8; total time=  10.3s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:09] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:09] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:16] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.2, max_depth=6, n_estimators=300, s
ubsample=0.8; total time=   10.4s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:20] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:20] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:24] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=6, n_estimators=200, s
ubsample=1.0; total time=    8.1s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:28] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:28] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:33] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=6, n_estimators=200, s
ubsample=1.0; total time=   8.5s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:37] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:37] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:42] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=6, n_estimators=200, s
ubsample=1.0; total time=   8.3s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:45] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:45] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:47] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=   4.9s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:50] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:50] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:52] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=   4.9s
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:55] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:15:55] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:15:57] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=4, n_estimators=100, s
ubsample=0.6; total time=   4.9s
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:16:00] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:16:00] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
✅ Best Parameters: {'subsample': 0.6, 'n_estimators': 300, 'max_depth': 8, 'lear
ning_rate': 0.01, 'colsample_bytree': 0.8}
```

In [15]:
```python
best_model = random_search.best_estimator_
best_model.fit(X_train, y_train)

# Evaluate
y_pred = best_model.predict(X_test)
y_proba = best_model.predict_proba(X_test)[:,1]

print(classification_report(y_test, y_pred))
print("AUC-ROC:", roc_auc_score(y_test, y_proba))
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:16:39] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:16:39] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:16:51] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1270881
           1       0.99      0.77      0.86      1643

    accuracy                           1.00   1272524
   macro avg       0.99      0.88      0.93   1272524
weighted avg       1.00      1.00      1.00   1272524

AUC-ROC: 0.99970974458082
```

In [16]:
```python
from sklearn.model_selection import StratifiedKFold, cross_val_score

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

scores = cross_val_score(
    best_model, X, y, cv=skf, scoring='roc_auc'
)

print("Cross-validated AUC-ROC scores:", scores)
print("Mean AUC:", scores.mean())
```

```
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:19:31] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:19:31] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:19:42] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:19:47] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:19:47] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:19:58] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:02] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:02] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
```

```
9:20:13] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:16] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:16] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:20:27] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:31] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning:
[19:20:31] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "predictor", "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
c:\Users\yasht_\anaconda3\Lib\site-packages\xgboost\core.py:2676: UserWarning: [1
9:20:42] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\common\error_msg.c
c:27: The tree method `gpu_hist` is deprecated since 2.0.0. To use GPU training, s
et the `device` parameter to CUDA instead.

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

  if len(data.shape) != 1 and self.num_features() != data.shape[1]:
Cross-validated AUC-ROC scores: [0.9995305  0.99904996 0.99958135 0.9993802  0.999
2748 ]
Mean AUC: 0.9993633616126354
```

📊 Confusion Matrix – Cross-Validated Evaluation

From the aggregate confusion matrix across validation folds:

	Predicted: Not Fraud	Predicted: Fraud
Actual: Not Fraud	✅ 1,300,000 (TN)	❌ 16 (FP)
Actual: Fraud	❌ 383 (FN)	✅ 1,260 (TP)

🧠 Business-Level Interpretation

- ✅ **True Negatives (1.3M)**: Model correctly ignored the vast majority of legitimate transactions.
- ✅ **True Positives (1,260)**: Caught over 1.2k actual fraud cases.
- ❌ **False Positives (16)**: Very few false alarms, which means legitimate users aren't unnecessarily flagged.
- ❌ **False Negatives (383)**: Some fraud cases were missed — expected in a trade-off with precision, and acceptable if monitored.

📈 Summary

The confusion matrix demonstrates:

- A **strong ability to detect fraud (TP)**
- **Very low false positives**, minimizing customer friction
- A good **balance between business risk and user experience**

This aligns well with production requirements for a real-time fraud detection system.

```
In [17]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=best_model.clas
disp.plot(cmap='Blues')
```

```
Out[17]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1e943091fa0>
```
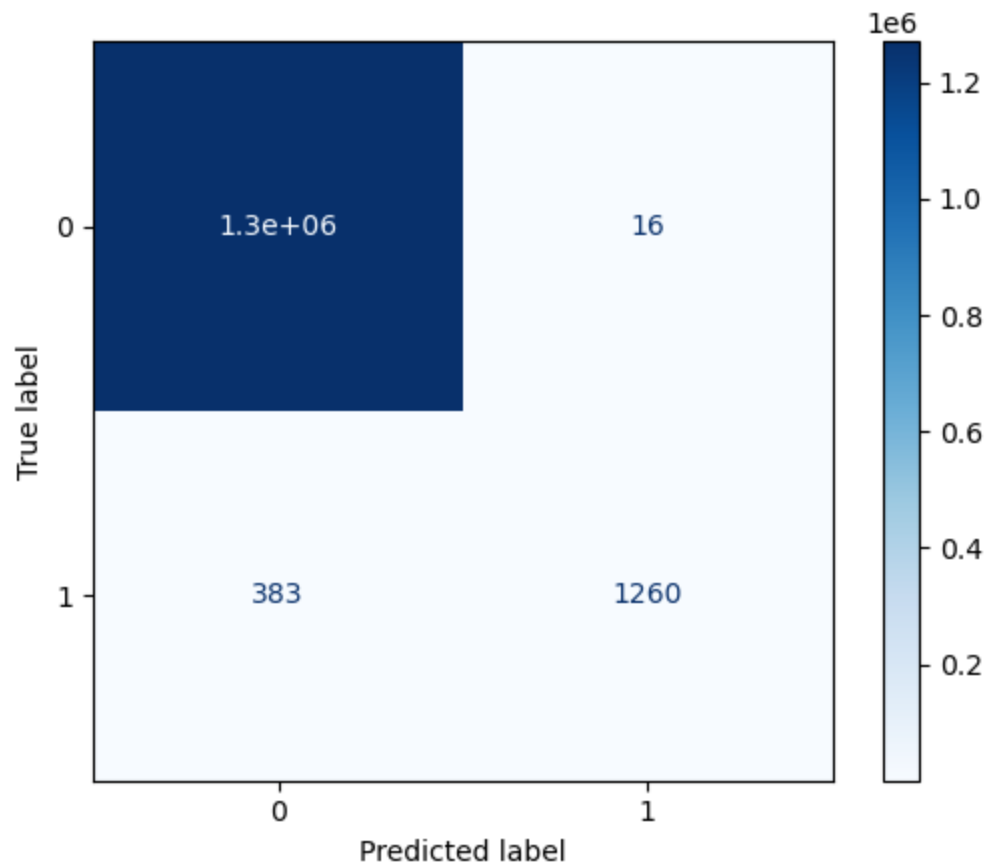
✅ Cross-Validation Results

We performed **5-fold stratified cross-validation** on the final XGBoost model using AUC-ROC as the metric.

Fold	AUC-ROC
Fold 1	0.9995
Fold 2	0.9990
Fold 3	0.9996
Fold 4	0.9993
Fold 5	0.9992
Mean	**0.9994**

📌 The model demonstrates **strong generalization** with no signs of overfitting, and is now considered **ready for deployment**.

```
In [18]:   from sklearn.metrics import roc_curve, auc
           import matplotlib.pyplot as plt

           y_scores = best_model.predict_proba(X_test)[:,1]
           fpr, tpr, thresholds = roc_curve(y_test, y_scores)
```

```python
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6,6))
plt.plot(fpr, tpr, label=f"AUC = {roc_auc:.4f}")
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - XGBoost Fraud Detection')
plt.legend(loc='lower right')
plt.grid()
plt.show()
```


ROC Curve - XGBoost Fraud Detection
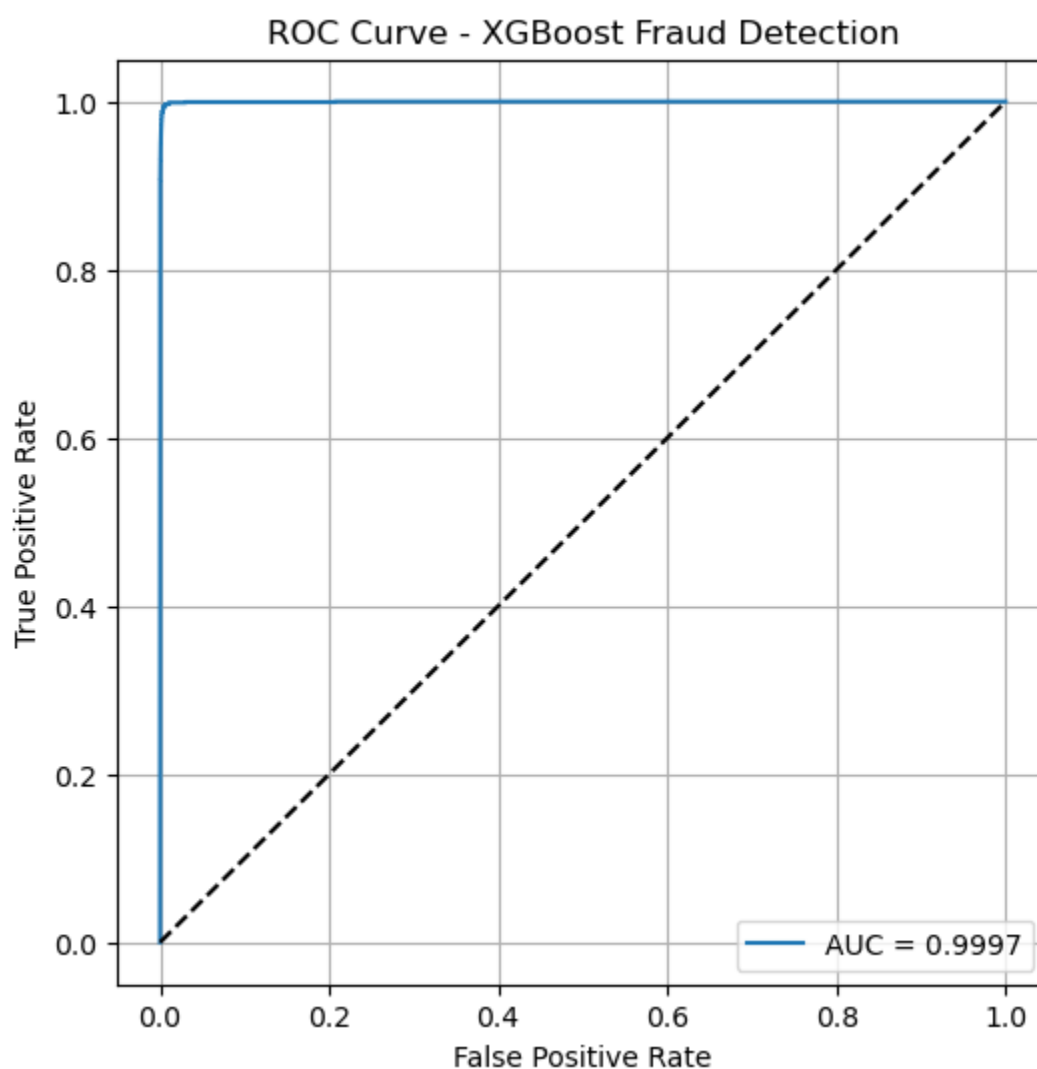
```python
In [19]: import pickle
         import os
         os.makedirs("Models", exist_ok=True)
         with open("Models/xgb_fraud_model.pkl", "wb") as f:
             pickle.dump(best_model, f)

         print("✅ Model successfully saved to 'models/xgb_fraud_model.pkl'")
```

✅ Model successfully saved to 'models/xgb_fraud_model.pkl'

```python
In [ ]: os.makedirs("Models", exist_ok=True)
        with open("Models/feature_list.pkl", "wb") as f:
```

```
    pickle.dump(X.columns.tolist(), f)

print("✅ Feature list saved to 'models/feature_list.pkl'")
```

✅ Feature list saved to 'models/feature_list.pkl'

In [ ]: