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

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from sklearn.ensemble import RandomForestClassifier
```

```
df=pd.read_csv('/content/upi_transactions_2024.csv',engine='python',on_bad_lines='skip')
df
```

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status	sender_age_group	receiver_age_
0	TXN0000000001	2024-10-08 15:17:28	P2P	Entertainment	868	SUCCESS	26-35	
1	TXN0000000002	2024-04-11 06:56:00	P2M	Grocery	1011	SUCCESS	26-35	
2	TXN0000000003	2024-04-02 13:27:18	P2P	Grocery	477	SUCCESS	26-35	
3	TXN0000000004	2024-01-07 10:09:17	P2P	Fuel	2784	SUCCESS	26-35	
4	TXN0000000005	2024-01-23 19:04:23	P2P	Shopping	990	SUCCESS	26-35	
...	...	...	...	...	...	...	...	
332928	TXN0000249996	2024-11-08 22:41:43	Recharge	Food	373	SUCCESS	36-45	
332929	TXN0000249997	2024-12-15 02:58:03	P2P	Utilities	2025	SUCCESS	36-45	
332930	TXN0000249998	2024-11-27 16:33:25	P2P	Food	468	SUCCESS	26-35	
332931	TXN0000249999	2024-01-05 13:31:30	Recharge	Healthcare	284	SUCCESS	18-25	
332932	TXN0000250000	2024-01-17 15:23:07	P2P	Entertainment	531	SUCCESS	18-25	

332933 rows × 17 columns

```
df.shape
```

```
(332933, 17)
```

```
df.head()
```

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status	sender_age_group	receiver_age_group
0	TXN0000000001	2024-10-08 15:17:28	P2P	Entertainment	868	SUCCESS	26-35	18-25
1	TXN0000000002	2024-04-11 06:56:00	P2M	Grocery	1011	SUCCESS	26-35	26-35
2	TXN0000000003	2024-04-02 13:27:18	P2P	Grocery	477	SUCCESS	26-35	36-45
3	TXN0000000004	2024-01-07 10:09:17	P2P	Fuel	2784	SUCCESS	26-35	26-35
4	TXN0000000005	2024-01-23 19:04:23	P2P	Shopping	990	SUCCESS	26-35	18-25

```
df.tail()
```

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status	sender_age_group	receiver_age_
332928	TXN0000249996	2024-11-08 22:41:43	Recharge	Food	373	SUCCESS	36-45	
332929	TXN0000249997	2024-12-15 02:58:03	P2P	Utilities	2025	SUCCESS	36-45	
332930	TXN0000249998	2024-11-27 16:33:25	P2P	Food	468	SUCCESS	26-35	
332931	TXN0000249999	2024-01-05 13:31:30	Recharge	Healthcare	284	SUCCESS	18-25	
332932	TXN0000250000	2024-01-17 15:23:07	P2P	Entertainment	531	SUCCESS	18-25	

```
df.columns
```

```
Index(['transaction id', 'timestamp', 'transaction type', 'merchant_category',
       'amount (INR)', 'transaction_status', 'sender_age_group',
       'receiver_age_group', 'sender_state', 'sender_bank', 'receiver_bank',
       'device_type', 'network_type', 'fraud_flag', 'hour_of_day',
       'day_of_week', 'is_weekend'],
      dtype='object')
```

```
df.dtypes
```

	0
transaction id	object
timestamp	object
transaction type	object
merchant_category	object
amount (INR)	object
transaction_status	object
sender_age_group	object
receiver_age_group	object
sender_state	object
sender_bank	object
receiver_bank	object
device_type	object
network_type	object
fraud_flag	float64
hour_of_day	object
day_of_week	object
is_weekend	float64

dtype: object

```
df.isna().sum()
```

	0
transaction id	0
timestamp	0
transaction type	1
merchant_category	1
amount (INR)	1
transaction_status	1
sender_age_group	1
receiver_age_group	2
sender_state	2
sender_bank	3
receiver_bank	5
device_type	5
network_type	6
fraud_flag	6
hour_of_day	7
day_of_week	7
is_weekend	9

dtype: int64

```
df=df.dropna()
df
```

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status	sender_age_group	receiver_age_
0	TXN0000000001	2024-10-08 15:17:28	P2P	Entertainment	868	SUCCESS	26-35	
1	TXN0000000002	2024-04-11 06:56:00	P2M	Grocery	1011	SUCCESS	26-35	
2	TXN0000000003	2024-04-02 13:27:18	P2P	Grocery	477	SUCCESS	26-35	
3	TXN0000000004	2024-01-07 10:09:17	P2P	Fuel	2784	SUCCESS	26-35	
4	TXN0000000005	2024-01-23 19:04:23	P2P	Shopping	990	SUCCESS	26-35	
...	...	...	...	...	...	...	...	
332928	TXN0000249996	2024-11-08 22:41:43	Recharge	Food	373	SUCCESS	36-45	
332929	TXN0000249997	2024-12-15 02:58:03	P2P	Utilities	2025	SUCCESS	36-45	
332930	TXN0000249998	2024-11-27 16:33:25	P2P	Food	468	SUCCESS	26-35	
332931	TXN0000249999	2024-01-05 13:31:30	Recharge	Healthcare	284	SUCCESS	18-25	
332932	TXN0000250000	2024-01-17 15:23:07	P2P	Entertainment	531	SUCCESS	18-25	

332924 rows × 17 columns

```
from sklearn.preprocessing import LabelEncoder
lab = LabelEncoder()

for col in df.select_dtypes(include=['object']).columns:
    df[col] = lab.fit_transform(df[col])
df
```

```
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
x=df.drop('amount (INR)',axis=1)
y=df['amount (INR)']
df
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-83417693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
```

```
  df[col] = lab.fit_transform(df[col])
/tmp/ipython-input-834d7693.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df[col] = lab.fit_transform(df[col])
```

	transaction id	timestamp	transaction type	merchant_category	amount (INR)	transaction_status	sender_age_group	receiver_age_group
0	0	191652	2	1	9510	1	1	
1	1	68756	1	4	66	1	1	
2	2	62750	2	4	5782	1	1	
3	3	4131	2	3	3563	1	1	
4	4	15522	2	7	10299	1	1	
1	1	68756	1	4	66	1	1	
...	...	...	...	...	...	...	...	
332928	249986	213270	3	2	4628	1	2	
3	3	4131	2	3	3563	1	1	
332929	249987	237839	2	9	2647	1	2	
332930	249988	225749	2	2	5682	1	1	
332931	249989	2915	3	5	3625	1	0	
332932	249990	11236	2	1	6376	1	0	
332929	249987	237839	2	9	2647	1	2	
332930	249988	225749	2	2	5682	1	1	
332931	249989	2915	3	5	3625	1	0	

332924 rows × 17 columns

```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

332924 rows × 17 columns

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
print(X_train.shape)
print(y_train.shape)
```

```
(266339, 16)
(266339,)
```

```
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier(n_estimators=30,random_state=42)    #reduce trees
model.fit(X_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier(n_estimators=30, random_state=42)
```

```
from sklearn.metrics import accuracy_score
y_pred=model.predict(X_test)
print("Accuracy:",accuracy_score(y_test,y_pred))
```

```
Accuracy: 1.0
```

```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
cm=confusion_matrix(y_test,y_pred)
ConfusionMatrixDisplay(confusion_matrix=cm).plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7a96f16e64b0>
```