Source code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Load the CSV
df = pd.read_csv('/content/ds_guarding_500.csv')

# Encode categorical columns
label_cols = ['country', 'merchant_type']
le = LabelEncoder()
for col in label_cols:
    df[col] = le.fit_transform(df[col])

# Define features and label
X = df.drop(['transaction_id', 'is_fraud'], axis=1)
y = df['is_fraud']

# Split data
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


# Train model

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

 Evaluate

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))


avg_by_hour_fraud = df.groupby(['hour', 'is_fraud'])['amount'].mean().unstack()


# Line plot

plt.figure(figsize=(8, 4))

plt.plot(avg_by_hour_fraud.index, avg_by_hour_fraud[0], label='Not Fraud',
marker='o')

plt.plot(avg_by_hour_fraud.index, avg_by_hour_fraud[1], label='Fraud',
marker='x')

plt.title('Avg. Transaction Amount by Hour (Fraud vs Non-Fraud)')

plt.xlabel('Hour of Day')

plt.ylabel('Amount')

plt.grid(True)

plt.xticks(range(0, 24))

plt.legend()

plt.tight_layout()
```

```
plt.show()


plt.figure(figsize=(8, 5))
sns.kdeplot(df[df['is_fraud'] == 0]['amount'], label='Not Fraud', fill=True, color='green')
sns.kdeplot(df[df['is_fraud'] == 1]['amount'], label='Fraud', fill=True, color='red')
plt.title('KDE Plot of Transaction Amount by Fraud Label')
plt.xlabel('Transaction Amount')
plt.ylabel('Density')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()


plt.figure(figsize=(8, 4))
plt.hist(df['amount'], bins=30, color='skyblue', edgecolor='black')
plt.title('Histogram of Transaction Amounts')
plt.xlabel('Amount')
plt.ylabel('Frequency')
plt.grid(True)
plt.tight_layout()
plt.show()


fraud_by_country = df.groupby('country')['is_fraud'].mean().sort_values()
```

```
plt.figure(figsize=(10, 5))
sns.barplot(x=fraud_by_country.index, y=fraud_by_country.values)
plt.title('Fraud Rate by Country')
plt.ylabel('Fraud Rate')
plt.xlabel('Country')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# First, encode categorical columns temporarily for plotting
df_encoded = df.copy()
from sklearn.preprocessing import LabelEncoder
df_encoded['country'] = LabelEncoder().fit_transform(df['country'])
df_encoded['merchant_type'] =
LabelEncoder().fit_transform(df['merchant_type'])

# Use a few numeric columns
sns.pairplot(df_encoded[['amount', 'hour', 'card_present', 'country', 'is_fraud']],
hue='is_fraud')
plt.suptitle("Pair Plot", y=1.02)
plt.show()

plt.figure(figsize=(8, 6))
corr = df_encoded.corr()
```

```
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()

plt.figure(figsize=(6, 5))
sns.boxplot(x='is_fraud', y='amount', data=df)
plt.title('Box Plot of Transaction Amounts by Fraud Status')
plt.xlabel('Is Fraud')
plt.ylabel('Amount')
plt.grid(True)
plt.tight_layout()
plt.show()
```