Fraud detection

Python code

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,
confusion_matrix, accuracy_score
from sklearn.preprocessing import StandardScaler

# Step 1: Load Dataset
df = pd.read_csv('creditcard.csv')  # Ensure the file is in the
same directory

# Step 2: Data Preprocessing
print("Original dataset shape:", df.shape)
print("Fraud cases:", df['Class'].sum())

# Step 3: Handle imbalance by undersampling
fraud = df[df['Class'] == 1]
non_fraud = df[df['Class'] == 0].sample(n=len(fraud),
random_state=42)
balanced_df = pd.concat([fraud, non_fraud]).sample(frac=1,
random_state=42)
```

```
# Step 4: Prepare features and labels
X = balanced_df.drop(['Class', 'Time'], axis=1)
y = balanced_df['Class']

# Step 5: Normalize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 6: Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.3, random_state=42)

# Step 7: Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
# Step 8: Evaluate the model
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test,
y_pred))

# Step 9: Predict new transaction
def predict_transaction(input_features):
    input_scaled = scaler.transform([input_features])
    pred = model.predict(input_scaled)[0]
    return "Fraudulent" if pred == 1 else "Legitimate"
```