```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import SMOTE
import joblib

# Step 1: Load and Preprocess the Data
data = pd.read_csv('/content/creditcard.csv', on_bad_lines='skip')

# Check for missing values
print("Missing values per column:\n", data.isnull().sum())

# Separate features and target variable
X = data.drop(columns=['Class'])  # Assuming 'Class' is the target column
y = data['Class']

# Normalize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 2: Handle Class Imbalance using SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)

# Step 3: Split the Data into Training and Testing Sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.3, random_state=42)

# Step 4: Train the Model using Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 5: Evaluate the Model
y_pred = model.predict(X_test)

# Calculate metrics
accuracy = accuracy_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Print the results
print(f'Accuracy: {accuracy}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
print('Confusion Matrix:')
print(conf_matrix)

# Step 6: Save the Model
joblib.dump(model, 'credit_card_fraud_model.pkl')

# Step 7: Load and Use the Model (if needed)
# loaded_model = joblib.load('credit_card_fraud_model.pkl')
# new_predictions = loaded_model.predict(X_test)
```

```
<ipython-input-3-6c1da513be7a>:11: DtypeWarning: Columns (2,3,4,6,8,9,10,11,13,14,15,17,18,19,20,21,24,25) have mixed types. Specif
  data = pd.read_csv('/content/creditcard.csv', on_bad_lines='skip')
Missing values per column:
 Time        0
V1          0
V2          0
V3          0
V4          0
V5          0
V6          1
V7          3
V8          8
V9          9
V10        12
V11        14
V12        20
V13        26
V14        32
V15        46
V16        51
V17        59
V18        63
V19        68
V20        79
V21        86
V22        99
V23       109
V24       118
V25       129
V26       141
V27       151
V28       170
Amount    177
Class     186
dtype: int64
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-3-6c1da513be7a> in <cell line: 22>()
     20 # Normalize the data
     21 scaler = StandardScaler()
---> 22 X_scaled = scaler.fit_transform(X)
     23 
     24 # Step 2: Handle Class Imbalance using SMOTE
```

8 frames

```
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in __array__(self, dtype)
   2082     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
   2083         values = self._values
-> 2084         arr = np.asarray(values, dtype=dtype)
   2085         if (
   2086             astype_is_view(values.dtype, arr.dtype)

ValueError: could not convert string to float: '-0.7336.90479989626219'
```

Next steps: Explain error