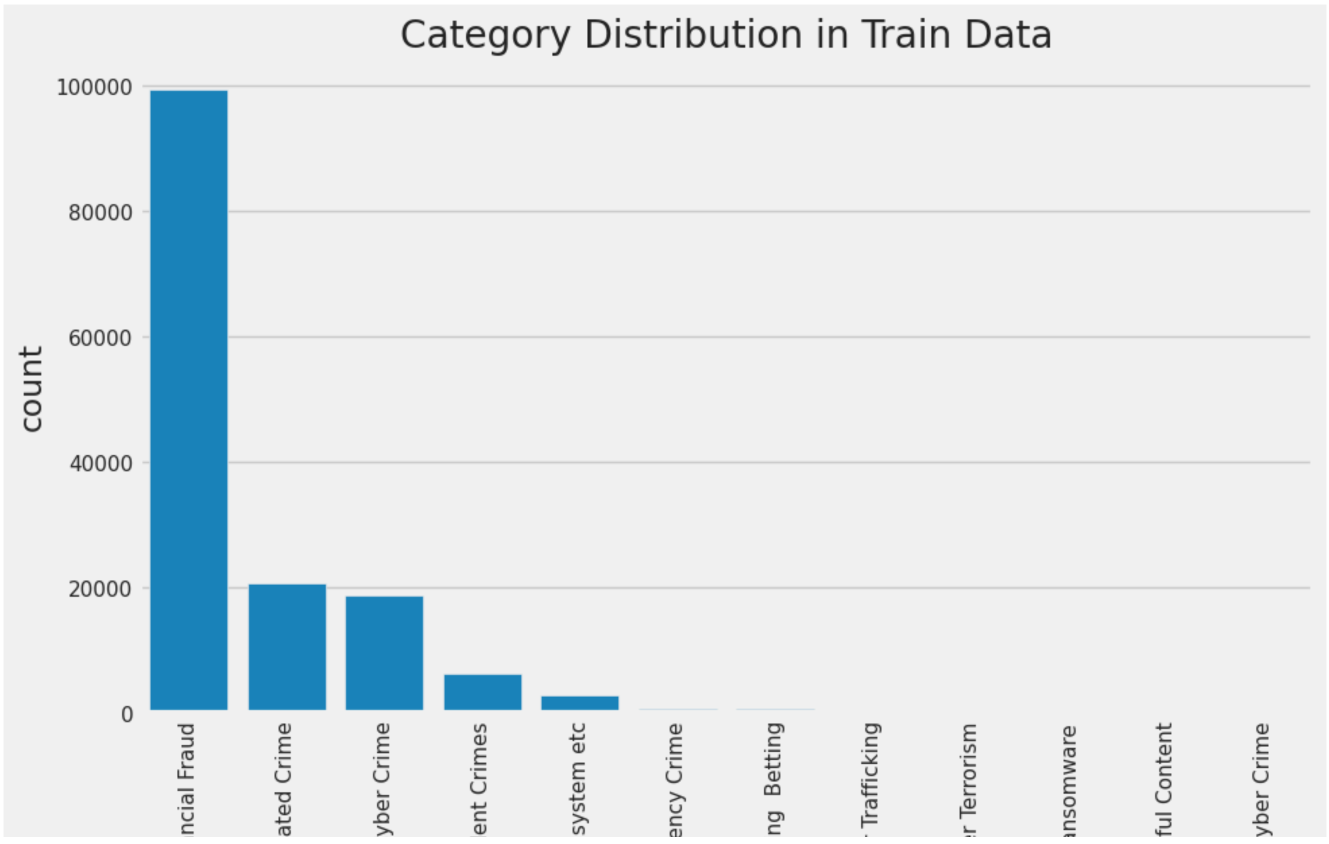
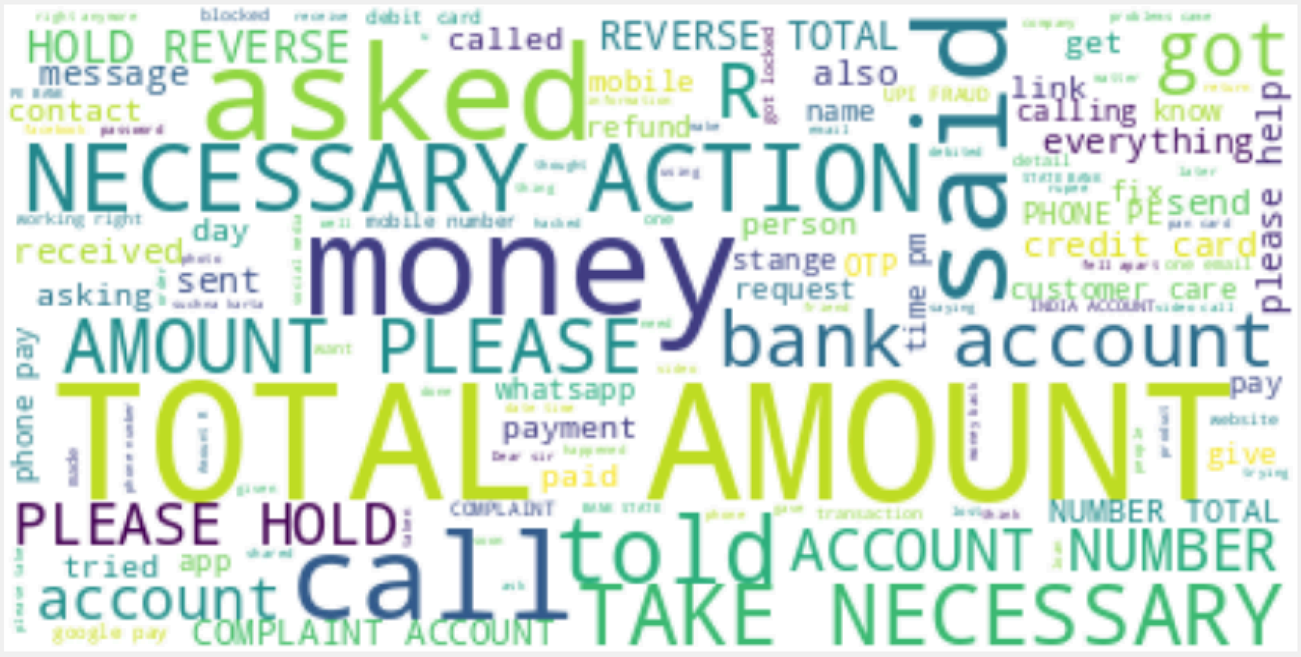
```
# Import necessary libraries
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
import os

# Download NLTK stopwords
nltk.download('stopwords')

# Set up matplotlib and seaborn styles
sns.set(style="whitegrid")
plt.style.use("fivethirtyeight")

# File paths for train and test datasets
train_file_path = '/content/train.csv'
test_file_path = '/content/test.csv'

# Function to load and clean CSV data
def load_and_clean_csv(file_path):
    try:
        # Try loading the CSV with default settings
        data = pd.read_csv(file_path)
    except pd.errors.ParserError:
        print(f"ParserError encountered in {file_path}. Attempting to clean...")
        cleaned_rows = []
        with open(file_path, 'r') as file:
            import csv
            reader = csv.reader(file)
            for row in reader:
                # Ensure rows have the expected number of columns
                if len(row) == 3:  # Adjust the number to match your dataset's column count
                    cleaned_rows.append(row)
        data = pd.DataFrame(cleaned_rows[1:], columns=cleaned_rows[0])  # Use the first row as headers
    except Exception as e:
        print(f"Error loading {file_path}: {e}")
        return None

    # Ensure data types are consistent and handle missing values
    data = data.replace(r'^\s*$', np.nan, regex=True)  # Replace empty strings with NaN
    data = data.dropna()  # Drop rows with missing values
    return data

# Load and clean the datasets
train_data = load_and_clean_csv(train_file_path)
test_data = load_and_clean_csv(test_file_path)

# Check if datasets are loaded properly
if train_data is not None:
    print("Train dataset loaded successfully.")
    print(train_data.head())
else:
    print("Failed to load train dataset.")

if test_data is not None:
    print("Test dataset loaded successfully.")
    print(test_data.head())
else:
    print("Failed to load test dataset.")

# Data exploration and visualization
if train_data is not None:
    print("Train dataset info:")
    print(train_data.info())

    # Bar plot for category distribution
    plt.figure(figsize=(10, 6))
    sns.countplot(data=train_data, x='category', order=train_data['category'].value_counts().index)
    plt.title('Category Distribution in Train Data')
    plt.xticks(rotation=90)
    plt.show()

    # Generate a Word Cloud
    if 'crimeaditionalinfo' in train_data.columns:  # Replace with the actual text column
        text = ' '.join(train_data['crimeaditionalinfo'])
        wordcloud = WordCloud(stopwords=set(stopwords.words('english')),
                              background_color='white', max_words=200).generate(text)
        plt.figure(figsize=(10, 7))
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.axis('off')
        plt.title("Word Cloud for Train Dataset")
        plt.show()

# Text Classification
if train_data is not None:
    # Combine all text for vectorization
    train_texts = train_data['crimeaditionalinfo']
    train_labels = train_data['category']

    # Train-test split
    X_train, X_val, y_train, y_val = train_test_split(train_texts, train_labels, test_size=0.2, random_state

    # Text vectorization using CountVectorizer
    vectorizer = CountVectorizer(stop_words='english', max_features=10000)
    X_train_vec = vectorizer.fit_transform(X_train)
    X_val_vec = vectorizer.transform(X_val)

    # Train a Naive Bayes model
    model = MultinomialNB()
    model.fit(X_train_vec, y_train)

    # Predictions and evaluation
    y_pred = model.predict(X_val_vec)

    # Classification report
    print("Classification Report:")
    report = classification_report(y_val, y_pred)
    print(report)

    # Save classification report to a CSV file
    report_dict = classification_report(y_val, y_pred, output_dict=True)
    report_df = pd.DataFrame(report_dict).transpose()
    report_df.to_csv('/content/evaluation_report.csv', index=True)
    print("Evaluation report saved as 'evaluation_report.csv'.")

    # Confusion Matrix
    cm = confusion_matrix(y_val, y_pred)
    plt.figure(figsize=(10, 7))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=model.classes_, yticklabels=model.classes_
    plt.title("Confusion Matrix")
    plt.xlabel("Predicted Labels")
    plt.ylabel("True Labels")
    plt.show()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
ParserError encountered in /content/train.csv. Attempting to clean...
Train dataset loaded successfully.
                                category                       sub_category  \
0  Online and Social Media Related Crime  Cyber Bullying  Stalking  Sexting
1                 Online Financial Fraud                  Fraud CallVishing
2               Online Gambling  Betting           Online Gambling  Betting
3  Online and Social Media Related Crime                   Online Job Fraud
4                 Online Financial Fraud                  Fraud CallVishing

                                  crimeaditionalinfo
0  I had continue received random calls and abusi...
1  The above fraudster is continuously messaging ...
2  He is acting like a police and demanding for m...
3  In apna Job I have applied for job interview f...
4  I received a call from lady stating that she w...
Test dataset loaded successfully.
                         category                          sub_category  \
1          Online Financial Fraud  DebitCredit Card FraudSim Swap Fraud
2  Cyber Attack/ Dependent Crimes                         SQL Injection
3          Online Financial Fraud                     Fraud CallVishing
4           Any Other Cyber Crime                                 Other
5          Online Financial Fraud        Internet Banking Related Fraud

                                  crimeaditionalinfo
1          KOTAK MAHINDRA BANK FRAUD\r\nFRAUD AMOUNT
2  The issue actually started when I got this ema...
3  I am amit kumar from karwi chitrakoot I am tot...
4  I have ordered  saree and  blouse from rinki s...
5  My salary of amount  has to be credited to my ...
Train dataset info:
<class 'pandas.core.frame.DataFrame'>
Index: 150555 entries, 0 to 164097
Data columns (total 3 columns):
 #   Column              Non-Null Count   Dtype 
---  ------              --------------   ----- 
 0   category            150555 non-null  object
 1   sub_category        150555 non-null  object
 2   crimeaditionalinfo  150555 non-null  object
dtypes: object(3)
memory usage: 4.6+ MB
None
```



Word Cloud for Train Dataset



```
Classification Report:
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
                                                precision    recall  f1-score   support

                          Any Other Cyber Crime       0.33      0.42      0.37      3736
                           Cryptocurrency Crime       0.20      0.83      0.32       154
                     Cyber Attack/ Dependent Crimes       1.00      1.00      1.00      1316
                                Cyber Terrorism       0.25      0.19      0.22        57
Hacking  Damage to computercomputer system etc       0.18      0.61      0.28       556
                         Online Cyber Trafficking       0.08      0.04      0.05        50
                           Online Financial Fraud       0.92      0.78      0.84     20019
                          Online Gambling  Betting       0.16      0.30      0.21       164
           Online and Social Media Related Crime       0.62      0.64      0.63      4039
                                     Ransomware       0.42      0.26      0.32        19
                         Report Unlawful Content       0.00      0.00      0.00         1

                                       accuracy                           0.72     30111
                                      macro avg       0.38      0.46      0.39     30111
                                   weighted avg       0.79      0.72      0.74     30111

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```