POST 50- GEN AI

GENERATIVE AI
FOR ALL

Using Generative AI for

Synthetic Fraud Data Generation

DINESHLAL

Introduction

- Fraud detection is a cornerstone of financial security. However, training effective machine learning models for this task is often hampered by the scarcity of high-quality, labeled fraud data.
- Fraudulent activities are, fortunately, relatively infrequent, leading to imbalanced datasets that pose challenges for developing robust models.
- One promising solution is leveraging Generative AI models, such as OpenAI's GPT,

Introduction

- To create synthetic fraud data and supplement existing datasets. This approach can help address data limitations and improve model generalization.
- This article explores how to use OpenAI GPT and Python to generate synthetic fraud data. We will also discuss how this data can enhance fraud detection models by tackling data scarcity and improving their performance.

Why Use Synthetic Fraud Data?

- **Overcome Data Scarcity:** Real fraud data is often limited due to privacy concerns, regulatory restrictions (e.g., GDPR, CCPA), and the inherent rarity of fraudulent events. Generative AI can create diverse, realistic fraud scenarios, effectively augmenting limited real-world data. Imagine trying to train a model to detect credit card fraud.
- You might have very few examples of actual fraudulent transactions, making it difficult for the model to learn the subtle patterns that distinguish them from legitimate purchases. Synthetic data can fill this gap.

Why Use Synthetic Fraud Data?

- **Balance Class Distribution:** Fraud datasets are frequently imbalanced, with a disproportionately small number of fraudulent cases compared to legitimate transactions.
- This imbalance can bias machine learning models, causing them to perform poorly in detecting actual fraud. Synthetic fraud transactions can help balance the dataset, leading to improved model performance.

Why Use Synthetic Fraud Data?

- **Improve Model Robustness:** Training models on more diverse data, including synthetic examples, enhances their ability to generalize and detect a wider range of fraud patterns. This makes the fraud detection system more robust.
- **Mitigate Data Privacy Concerns:** Synthetic data eliminates the risks associated with using sensitive real-world data. It allows organizations to train and test their models without compromising customer privacy or violating data protection regulations, while still maintaining the statistical properties of real data.

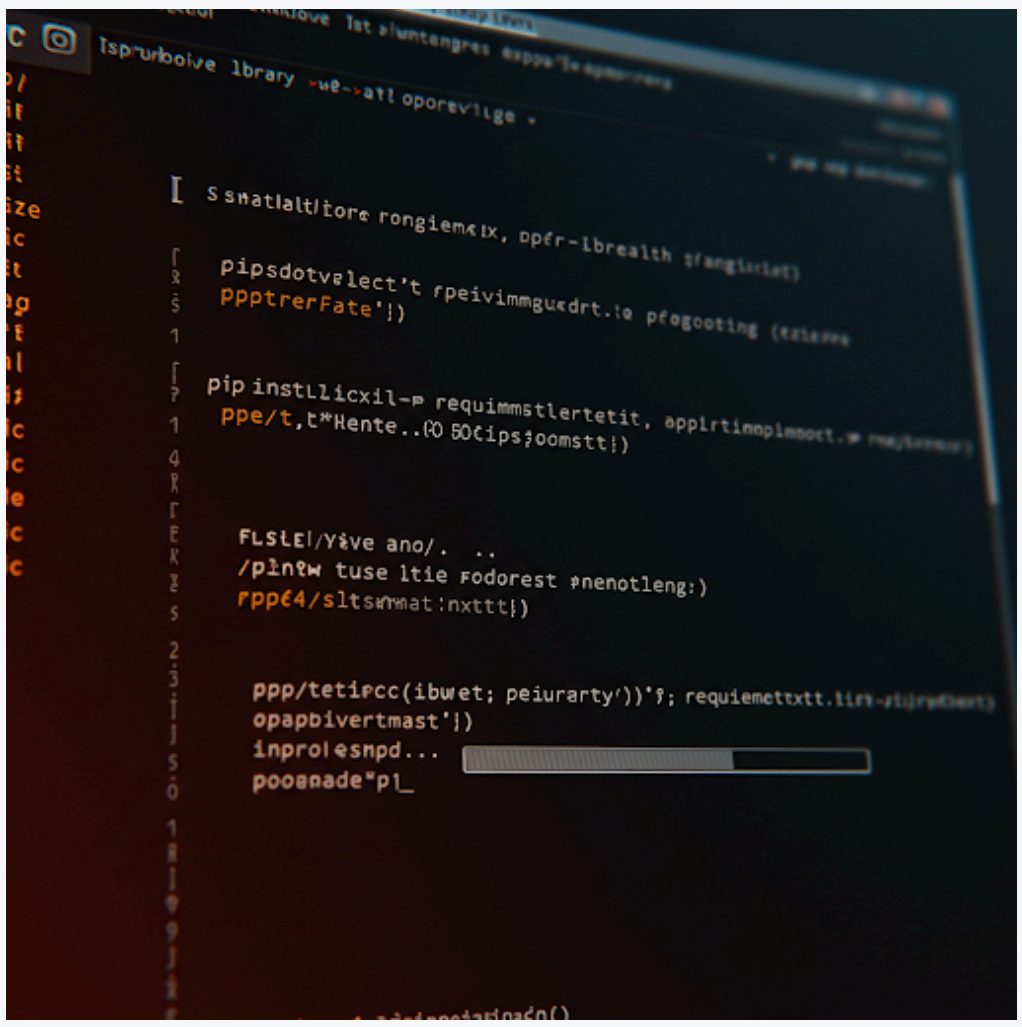
Generating Synthetic Fraud Data with OpenAI GPT

Step 1: Install Required Libraries

Ensure you have the necessary libraries installed in your Python environment:

Code:

pip install openai pandas



DINESHLAL

Generating Synthetic Fraud Data with OpenAI GPT

Step 2: Set Up OpenAI API Key

You'll need an OpenAI API key to access GPT. Set it up in your Python script:

Code:

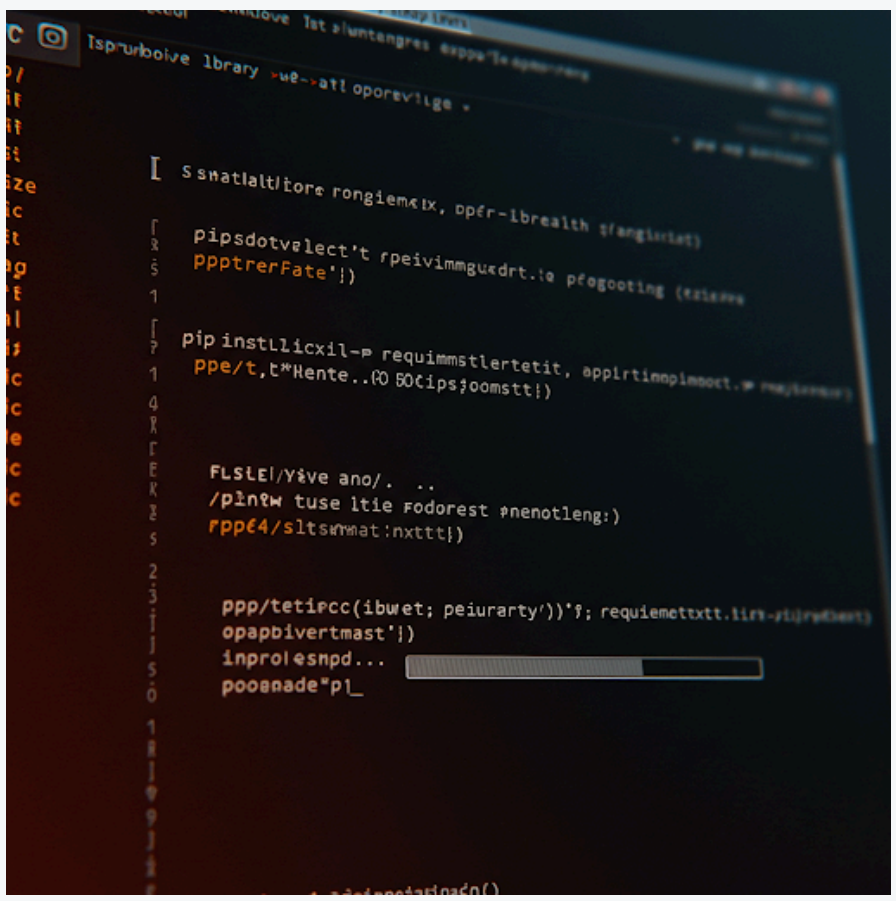
```
import openai
import pandas as pd
import json

# Set your OpenAI API Key
openai.api_key = "your_openai_api_key" # Replace with your actual key
```

Generating Synthetic Fraud Data with OpenAI GPT

Step 3: Define a Prompt for Generating Fraudulent Transactions

Craft a prompt that describes typical fraud transaction patterns. This prompt will guide GPT in generating realistic synthetic data.

Code in Next Page:



Generating Synthetic Fraud Data with OpenAI GPT

```
def generate_fraud_data(n_samples=10):
  prompt = (
      f"Generate {n_samples} synthetic fraudulent financial transactions in JSON format. "
          "Each transaction should include transaction_id, amount, transaction_type (e.g., online, in-store), location (e.g., city, country), timestamp, and fraud_flag (True/False)."
      "Consider various fraud scenarios, such as: "
      "- Large, unusual purchases"
      "- Multiple transactions in a short period"
      "- Transactions from unfamiliar locations"
       "- Use realistic but varied data for amounts, locations, and transaction types."
  )

  response = openai.ChatCompletion.create(
    model="gpt-4",  # or gpt-3.5-turbo
        messages=[{"role": "system", "content": "You are a data generator specializing in financial transactions."},
            {"role": "user", "content": prompt}]
  )

  try:
      fraud_data = json.loads(response['choices'][0]['message']['content'])
      return pd.DataFrame(fraud_data)
  except json.JSONDecodeError as e:
      print(f"Error decoding JSON: {e}")
          print(response['choices'][0]['message']['content']) # Print the raw response for debugging
      return pd.DataFrame() # Return empty DataFrame in case of error
```

Generating Synthetic Fraud Data with OpenAI GPT

Step 4: Generate and Save the Data

Generate the synthetic fraud transactions and save them to a CSV file:

Python Code Generation:

```
fraud_df = generate_fraud_data(1000) # Generate 1000 synthetic fraud transactions
if not fraud_df.empty: fraud_df.to_csv("synthetic_fraud_data.csv", index=False) print(fraud_df.head())
```

Enhancing Fraud Detection Models with Synthetic Data

Synthetic data can be integrated into fraud detection models in several ways:

- **Data Augmentation:** Combine synthetic data with real-world fraud data to increase the size and diversity of the training dataset.
- **Balanced Training:** Use synthetic data to balance the class distribution in the training data, improving the model's ability to detect fraud.

Enhancing Fraud Detection Models with Synthetic Data

Synthetic data can be integrated into fraud detection models in several ways:

- **Stress Testing:** Subject fraud detection systems to a wide range of synthetic fraud scenarios to evaluate their performance and identify potential weaknesses.

Example Integration with a Machine Learning Pipeline:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Assume real_df contains real transaction data
if not fraud_df.empty: # Check if the synthetic data generation was successful
    combined_df = pd.concat([real_df, fraud_df]) # Combine real and synthetic data
    X = combined_df.drop(columns=["fraud_flag"]) # Features
 y = combined_df["fraud_flag"] # Target variable

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

 clf = RandomForestClassifier()
 clf.fit(X_train, y_train)
 y_pred = clf.predict(X_test)
 print("Accuracy:", accuracy_score(y_test, y_pred))
else:
   print("Synthetic data generation failed. Cannot train the model.")
```

Conclusion

- Generating synthetic fraud data using OpenAI GPT provides a powerful solution to the challenges of limited real-world fraud data.
- By leveraging Generative AI, financial institutions and data scientists can develop more robust and resilient fraud detection models while adhering to data privacy regulations.
- As AI models continue to evolve, synthetic data generation will become an increasingly important tool for fraud prevention and security analytics. This approach empowers students and early-career professionals to contribute effectively to the fight against financial fraud.

THANK YOU

- Special thanks to Gemini and ChatGPT for all the help on content
- Follow along for more informative articles in Generative AI space

DINESHLAL