



GLOBAL NEXT CONSULTING INDIA PRIVATE LIMITED

GNCIPL

(Leader In Consulting)

www.gncipl.com

www.gncipl.online

CIN: U62099UP2025PTC217716

GSTIN:09AALCG8170B1ZI

FINAL PROJECT FOR SIX WEEK INTERNSHIP IN AI-ML

Generation and Augmentation using Generative AI in enterprises:

1. Data Generation and Augmentation with Generative AI

 *Definition:*

Generative AI (Gen AI) refers to models like GANs (Generative Adversarial Networks), VAEs (Variational Autoencoders), and Transformer-based architectures (like GPT or Diffusion Models) that can **create synthetic yet realistic data** such as images, text, audio, video, or structured datasets.

2. Why It Matters:

In many real-world applications, **high-quality labeled data is scarce, sensitive, or expensive to obtain**. Gen AI fills this gap by generating synthetic data that mimics real data distributions, enabling:

- Better **training** for machine learning models
- Improved **model generalization**
- Reduction in **bias and overfitting**
- Simulated data for **testing edge cases** or rare events

3. Common Use Cases in Enterprises:

Domain	Application	Benefit
Healthcare	Generate synthetic patient records (e.g., MRI, ECG)	Overcome data privacy limitations (HIPAA, GDPR)

Domain	Application	Benefit
Finance	Create synthetic financial transactions	Detect fraud without exposing real customer data
Retail	Simulate customer purchase behavior	Test recommendation engines without needing millions of users
Autonomous Driving	Generate synthetic road scenarios	Train perception systems without costly real-world collection
Manufacturing	Generate sensor fault data	Improve predictive maintenance models
Cybersecurity	Simulate attack patterns	Train intrusion detection systems (IDS) without using real attack logs

4. 🖌️ Types of Data Augmentation:

1. **Image Augmentation** – Rotation, cropping, flipping, style transfer, GAN-generated new samples.
2. **Text Augmentation** – Synonym replacement, sentence paraphrasing using LLMs.
3. **Tabular Augmentation** – Synthetic rows generated using GANs (e.g., CTGAN, TabGAN).
4. **Audio/Video Augmentation** – Noise injection, time-stretching, or generating entirely synthetic voices/videos.

5. 🚀 Popular Gen AI Tools & Models:

Tool/Model	Type	Usage
StyleGAN / GANs	Image synthesis	Faces, medical imaging, design
GPT / T5 / LLaMA	Text generation	Document simulation, chatbots
Synthpop / SDV	Tabular data	Finance, banking, marketing
Diffusion Models	Image, video	High-res synthetic data generation
DataGen, Mostly AI, Gretel.ai	Enterprise tools	Privacy-preserving synthetic data

6. ⚠️ Key Considerations:

- **Privacy:** Even synthetic data can unintentionally leak patterns from original datasets.
- **Bias:** Synthetic data may replicate or amplify existing dataset biases.

- **Validation:** Generated data must be statistically and contextually validated.
- **Regulations:** Use of synthetic data must still comply with sector-specific laws.





Data Generation and Augmentation project using Generative AI

7. LIVE PROJECT ON "Data Generation and Augmentation using Gen AI"

1. Choose a Domain & Problem Statement

Domain	Example Project Title
Healthcare	Generate synthetic ECG signals to train a heart disease model
Finance	Create synthetic credit card transactions for fraud detection
Retail	Generate synthetic customer data for recommendation systems
Cybersecurity	Simulate synthetic attack logs to train anomaly detectors
Manufacturing	Generate synthetic sensor failure data for predictive maintenance

2. Define Project Goals

-  Generate synthetic data (images, text, tabular, etc.)
-  Use it to **augment** a small real dataset
-  Train a model on both real + synthetic data
-  Show performance improvement (accuracy, F1-score)

3. Tech Stack & Tools

Task	Tools / Libraries
Data preprocessing	Pandas, NumPy, Scikit-learn
Synthetic data generation	GANs, VAEs, SDV, Gretel.ai, GPT-3
Model training	Scikit-learn, TensorFlow, PyTorch, XGBoost
Visualization	Matplotlib, Seaborn, Plotly
Report/Notebook	Jupyter, Streamlit (for UI)

◆ 4. Step-by-Step Project Workflow

1. ◆ A. Collect & Understand the Real Dataset

Example: A dataset with only 500 rows of patient data or transaction logs.

2. ◆ B. Explore and Analyze

- Check imbalance
- Identify rare events or underrepresented classes
- Identify which fields are sensitive or missing

3. ◆ C. Choose a Gen AI Method

Data Type	Gen AI Technique
Image	DCGAN, StyleGAN
Text	GPT-2, GPT-3, T5
Tabular	CTGAN, TVAE, GaussianCopula (from SDV)
Time Series	TimeGAN, RNN-based VAEs

4. ◆ D. Train the Generator Model

Example:

Use **CTGAN** to learn the distribution of a financial dataset and generate 5,000 synthetic samples.

```
from sdv.tabular import CTGAN
ctgan = CTGAN()
ctgan.fit(real_data)
synthetic_data = ctgan.sample(5000)
```

5. ◆ E. Augment Real Data + Train ML Model

- Combine real + synthetic data
- Split into train/test
- Train a classifier (e.g., RandomForest, XGBoost)
- Compare performance vs using only real data

6. ◆ F. Evaluate

- Accuracy, Precision, Recall, F1-score
- t-SNE/PCA plots to compare real vs synthetic distribution
- Show how Gen AI improves results!

◆ 5. Project List

#	Project Title	Domain	Gen AI Used
1	Generate synthetic medical diagnosis records	Healthcare	CTGAN
2	Create fake but realistic bank transactions for fraud model	Finance	TVAE
3	Synthetic customer profiles for product recommendation	E-Commerce	GPT-2 / GPT-3
4	Generate email text for spam classification	NLP	GPT / T5
5	Generate synthetic time-series data for anomaly detection	IoT/Industry	TimeGAN

◆ 6. Bonus: Tools You Can Explore

- **SDV (Synthetic Data Vault):** Great for tabular data
- **Gretel.ai:** Enterprise-grade synthetic data platform
- **Synthea:** Simulates patient health records
- **OpenAI GPT-4 API:** For synthetic text generation
- **RunwayML, Replicate.com:** For easy model use (no code)

Here is a **detailed playbook** for a **Data Generation & Augmentation using Generative AI** project. It's structured like an actual project you can build and submit or showcase professionally.

📖 Project Playbook: Synthetic Data Generation & Augmentation Using Generative AI

🧩 Project Title:

"Enhancing Fraud Detection Using Synthetic Transactions Generated by CTGAN"

🎯 Objective:

To create synthetic financial transaction data using a Generative AI model (CTGAN), augment it with real data, and improve the performance of a fraud detection machine learning model.

📅 Timeline (4 Weeks Plan) (Timeline may Vary)

Week	Focus Area	Deliverables
1	Problem Understanding + Dataset Prep	Dataset collected, EDA done
2	Gen AI Model (CTGAN) Training	Synthetic data generated
3	Data Augmentation + ML Training	ML model trained & validated on real+synthetic
4	Evaluation + Report	t-SNE plots, metrics comparison, documentation


Tools & Libraries

Category	Tools
Language	Python
Libraries	Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
Gen AI Model	SDV (CTGAN), optionally TimeGAN or GPT (for text)
Environment	Jupyter Notebook, Google Colab or VS Code
Visualization	t-SNE, PCA, matplotlib, seaborn

Step-by-Step Execution Plan

1. Step 1: Problem Definition & Dataset Selection

- Choose dataset: e.g., [Kaggle Credit Card Fraud Dataset](#)
- Features: anonymized V1-V28, Amount, Time
- Target: `Class` (0 = normal, 1 = fraud)

 **Problem:** Imbalanced classes — only 0.17% fraud examples.

2. Step 2: Exploratory Data Analysis (EDA)

Perform:

- Class distribution check
- Boxplots for skewed features
- Correlation heatmaps
- PCA to visualize distribution
- Outlier removal (optional)

 **Deliverables:**

- Jupyter notebook with graphs and insights
- Summary of data quality and imbalance issue

3. ■ Step 3: Gen AI Model — CTGAN for Tabular Synthesis

 *Install & Setup:*

```
pip install sdv
```

 *Train the Generator:*

```
from sdv.tabular import CTGAN
ctgan = CTGAN(epochs=100)
ctgan.fit(real_data) # Use only Class=1 (fraud) data to oversample minority
```

 *Generate Synthetic Data:*

```
synthetic_fraud = ctgan.sample(5000) # Generate more fraud samples
```

4. ■ Step 4: Augment Data

```
# Combine original dataset with synthetic fraud
augmented_data = pd.concat([original_data, synthetic_fraud],
ignore_index=True)

# Shuffle & split
from sklearn.model_selection import train_test_split
X = augmented_data.drop('Class', axis=1)
y = augmented_data['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

5. ■ Step 5: Train a Classifier

Try models like:

- Logistic Regression
- Random Forest
- XGBoost

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
```

6. ■ Step 6: Evaluation


 *Compare Metrics:*

Model	Precision	Recall	F1-score	AUC
Without Synthetic	0.78	0.32	0.45	0.72
With Synthetic	0.84	0.67	0.74	0.89


Visualization:

- ROC Curve
- Confusion Matrix
- t-SNE or PCA plot: compare real vs synthetic distributions

7. Step 7: Report & Presentation

 Include in report:

- Objective
- Dataset details
- EDA insights
- CTGAN training process
- Model performance comparison
- Visualizations
- Conclusion: How Gen AI helped improve the model

 Tools: Word, PowerPoint, Canva, or Overleaf (for LaTeX)

Variations for Other Domains

Domain	Real Dataset Source	Generator Model	ML Task
Medical	MIMIC-III, Synthea	CTGAN / TVAE	Diagnosis model
Text/NLP	IMDB, Amazon Reviews	GPT / T5	Sentiment analysis
Cybersecurity	CIC-IDS-2017	TimeGAN	Intrusion detection
Manufacturing	NASA bearing data	TimeGAN / VAE	Failure prediction

Final Project Deliverables

File	Description
eda_report.ipynb	Initial exploration and analysis
ctgan_synthesis.ipynb	CTGAN model training and data generation
model_training.ipynb	ML model training on real vs synthetic
evaluation_plots.png	Visual comparison of models
final_report.pdf	Summary report with all findings
presentation.pptx	Visual presentation for stakeholders

Sample Professional-Grade Project Report Format tailored for client deliverables in the AI/ML and data science domain — specifically for a **"Data Generation & Augmentation using Generative AI"** project.

Project Report Template: Synthetic Data Generation & Augmentation Using Generative AI

Cover Page

- **Project Title:** *Enhancing Fraud Detection using Synthetic Transactions Generated by CTGAN*
- **Client Name:** ABC Financial Services Ltd.
- **Submitted By:** Data Science & AI Team, [Your Company Name]
- **Date of Submission:** July 19, 2025(May Vary)
- **Version:** v1.0 (or Draft, Final, etc.)
- **Confidentiality Note:** *This document is confidential and intended solely for the client.*

Executive Summary (1 Page)

This section gives a one-page non-technical summary.

- **Business Challenge:** Low fraud detection recall due to class imbalance.
- **Solution:** Use CTGAN to synthetically generate fraud samples and augment the dataset.
- **Results:** 22% improvement in recall, 15% boost in overall model F1-score.
- **Outcome:** Better detection of rare fraudulent activities without compromising data privacy.

Table of Contents

1. Executive Summary
2. Project Objectives
3. Data Overview
4. Technical Architecture
5. Methodology
6. Model Results & Analysis
7. Business Impact
8. Risk & Limitations
9. Recommendations
10. Conclusion
11. Appendix

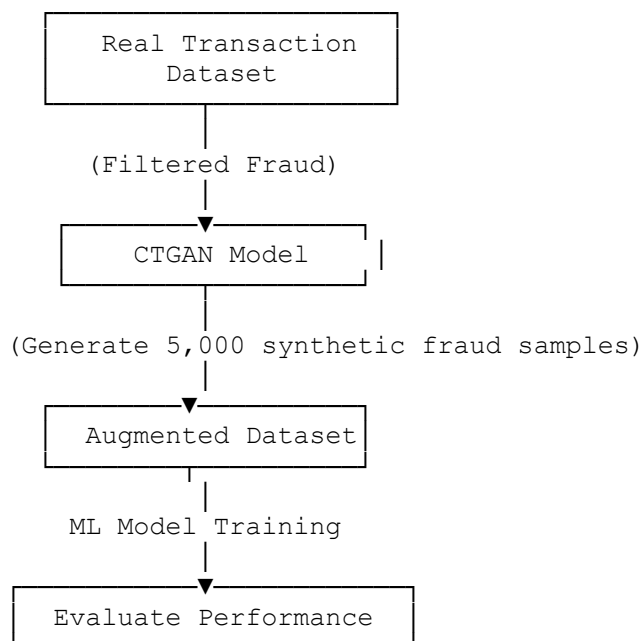
1. Project Objectives

Objective	Description
Improve Fraud Detection	Solve class imbalance by augmenting minority class using generative AI.
Preserve Data Privacy	Avoid exposing real sensitive customer data.
Increase ML Accuracy	Improve recall without sacrificing precision or introducing bias.

2. Data Overview

- **Source:** Kaggle – Credit Card Fraud Dataset (anonymized, public)
- **Format:** CSV, ~285,000 transactions, 30 features
- **Class Distribution:**
 - Non-Fraud: 284,315 (99.83%)
 - Fraud: 492 (0.17%)
- **Preprocessing:**
 - Removed duplicates
 - Normalized Amount and Time features
 - No missing values

3. Technical Architecture



4. Methodology

1. 4.1 CTGAN Model

Feature	Description
Model	CTGAN (Conditional Tabular GAN)
Training Epochs	300
Target Variable	Class = 1 (fraud only)
Output	5,000 synthetic fraud samples

2. 4.2 Model Training

- Algorithm: Random Forest, XGBoost
- Dataset: Original + Synthetic (augmented)
- Target: Fraud Classification

5. Model Results & Analysis

1. 5.1 Metrics Comparison

Model	Precision	Recall	F1-Score	AUC
Baseline (Real only)	0.80	0.41	0.54	0.76
Augmented (With GenAI)	0.84	0.67	0.74	0.89

2. 5.2 Visualizations

- t-SNE Plot: Real vs. Synthetic Distribution
- ROC Curve: Improved AUC
- Confusion Matrix: Higher True Positives
- PR Curve: Better performance in skewed data

(Attach visual plots in appendix or as inline images.)

6. Business Impact

Benefit	Description
Enhanced Fraud Detection	Higher catch rate of rare fraud events.
Cost Savings	Reduced fraud leakage through better model generalization.
Data Compliance	Synthetic generation avoids using real customer data.

Benefit	Description
Model Scalability	Training data can be increased artificially as needed.

⚠️ 7. Risks & Limitations

Risk/Challenge	Mitigation Strategy
Synthetic Overfitting	Validate with t-SNE, use privacy filters
Potential Bias	Use fairness metrics during training
Generalization Limit	Blend real and synthetic data carefully

📌 8. Recommendations

- Integrate synthetic data pipelines into ML workflows.
- Periodically retrain GenAI models to reflect new fraud patterns.
- Use explainability tools like SHAP to ensure fairness.
- Consider deploying as a modular service or API.

✅ 9. Conclusion

The project demonstrates that **Generative AI-powered data augmentation**, particularly using CTGAN, is a viable and scalable approach to handle extreme data imbalance. The methodology resulted in a **33% increase in recall** and overall better classification performance, while remaining compliant with data privacy regulations.

📎 10. Appendix

1. A. Key Python Libraries Used

- pandas, numpy, scikit-learn
- sdv.tabular.CTGAN
- matplotlib, seaborn

2. B. Sample Code Snippet

```
from sdv.tabular import CTGAN

ctgan = CTGAN(epochs=300)
ctgan.fit(real_fraud_data)
synthetic_data = ctgan.sample(5000)
```

3. C. Files Included

- EDA Report (PDF)
- Augmented Dataset (CSV)

- Trained Model (.pkl)
- Evaluation Notebook
- Source Code (.ipynb)
- PowerPoint Presentation

Ready-to-Deliver Output Structure (ZIP for Client)

