Project Summary

Batch details	PGP in Data Science with Specialization in GenAI
Team members	Shreya Bhat Bento M J Fernandes Shristi Agrawal Chinmayee Y Venkatesh Pattar
Domain of Project	Generative AI
Proposed project title	Generative AI for test Data synthesis
Group Number	6
Team Leader	Shreya Bhat
Mentor Name	Vaibhav Kulkarni

Date:

Signature of the Mentor

Signature of the Team Leader

Table of Contents

List of Figures

1. Industry Review

1.1. Industry Review – Current Practices & Background Research

Synthetic data generation is increasingly used across industries where privacy, regulatory constraints, or data scarcity hinder development and testing.
Current industry practices include:

- **Rule-based test data generators** used in QA teams (limited diversity).
- **Manual data masking & anonymization**, which often destroys important patterns.
- **Sampling and duplication**, which fails to represent edge cases and rare scenarios.
- **Vendor tools** like Tonic.ai, MostlyAI, Gretel.ai providing generative synthetic data for enterprise testing.
- **Diffusion Models & CTGAN** emerging as standard for high-quality tabular synthetic data.

Industry need is shifting toward generative AI–driven synthetic data to ensure:

- Privacy (GDPR/DPDP Act compliance)
- Wider test-coverage
- Simulating rare-production scenarios
- No dependency on sensitive customer data

1.2. Literature Survey – Publications & Ongoing Research

Key research demonstrates the maturity and importance of generative synthetic data:

- **SDV (Synthetic Data Vault) – MIT:** Pioneered frameworks like **CTGAN** (Conditional Tabular GAN) and **TVAE** (Tabular Variational Autoencoder) for high-fidelity tabular data synthesis. These models are standard for capturing mixed-type data dependencies.
- **TimeGAN:** An innovation by Jinsung Yoon et al., specifically for generating synthetic time-series data.
- **SynthEval (2024):** A critical framework designed to measure synthetic data quality against both **utility** and **privacy** standards.
- **Gartner (2024):** Predicts synthetic data will overshadow real data in AI training and testing.

Applications include:

- Test data generation
- ML model training
- Bias analysis
- Performance benchmarking
- Privacy-preserving AI research

2. Dataset and Domain

2.1. Data Dictionary (Census Income Dataset)

The project utilizes the **Census Income Dataset** for synthesizing structured, mixed-type tabular data.

Variable	Description	Type
age	Age of individual	Numeric
workclass	Type of employment	Categorical
fnlwgt	Sampling weight	Numeric
education	Highest education level	Categorical
education-num	Numerical representation of education	Numeric
marital-status	Marital background	Categorical
occupation	Type of job	Categorical
relationship	Relation to family	Categorical
race	Race categories	Categorical
sex	Gender	Categorical
capital-gain	Capital gains	Numeric
capital-loss	Capital losses	Numeric
hours-per-week	Weekly working hours	Numeric
native-country	Country of birth	Categorical
income	Income bracket (<=50K or >50K)	Target

2.2. Variable Categorization

- **Numeric variables (6):**
  age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week

- **Categorical variables (9):**
  workclass, education, marital-status, occupation, relationship, race, sex, native-country, income

2.3. Pre-processing Data Analysis

- **Missing / Null Values:** The current analysis shows zero null values across all major columns, indicating clean input data.
- **Redundancy:** The columns education and education-num are highly redundant/duplicates, as one is a numerical representation of the other. The column education will be dropped or replaced.
- **Feature Disregard:** fnlwgt (sampling weight) is typically not required for generative modeling and may be treated as optional or excluded to simplify the latent space.

2.4. Project Justification

Project Statement

To develop a generative AI system that learns the statistical characteristics of the Census Income dataset and produces synthetic tabular data for testing enterprise systems without violating privacy.

Complexity Involved

- Understanding high-cardinality categorical variables
- Modeling multi-modal numeric distributions
- Avoiding privacy leakage
- Preserving correlations between features (e.g., education vs income)
- Validating utility using statistical tests

Project Outcome

- **Commercial Value:** Secure test-data pipelines for BFSI, healthcare, telecom, retail.
- **Academic Value:** Benchmark for synthetic data quality, privacy analysis.
- **Social Value:** Promotes privacy-preserving AI and reduces data misuse.

3. Data Exploration (EDA)

3.1. Relationship Between Variables

I. Distribution of Numeric Variables

- **High Skewness:** capital-gain and capital-loss are extremely skewed toward zero. This confirms the need for a **Log-Transformation** to stabilize their distribution, as suggested in the treatment plan.
- **Multi-Modality:** The education-num distribution clearly shows multiple distinct peaks, reflecting the discrete nature of education levels (e.g., high school, bachelor's, master's). The Generative AI model (CTGAN) must be capable of accurately modeling these **multi-modal distributions**.
- **High Kurtosis:** hours-per-week is highly peaked around the standard full-time work week (40 hours). The model must replicate this peak accurately while also generating realistic, rarer values (outliers up to 99 hours).

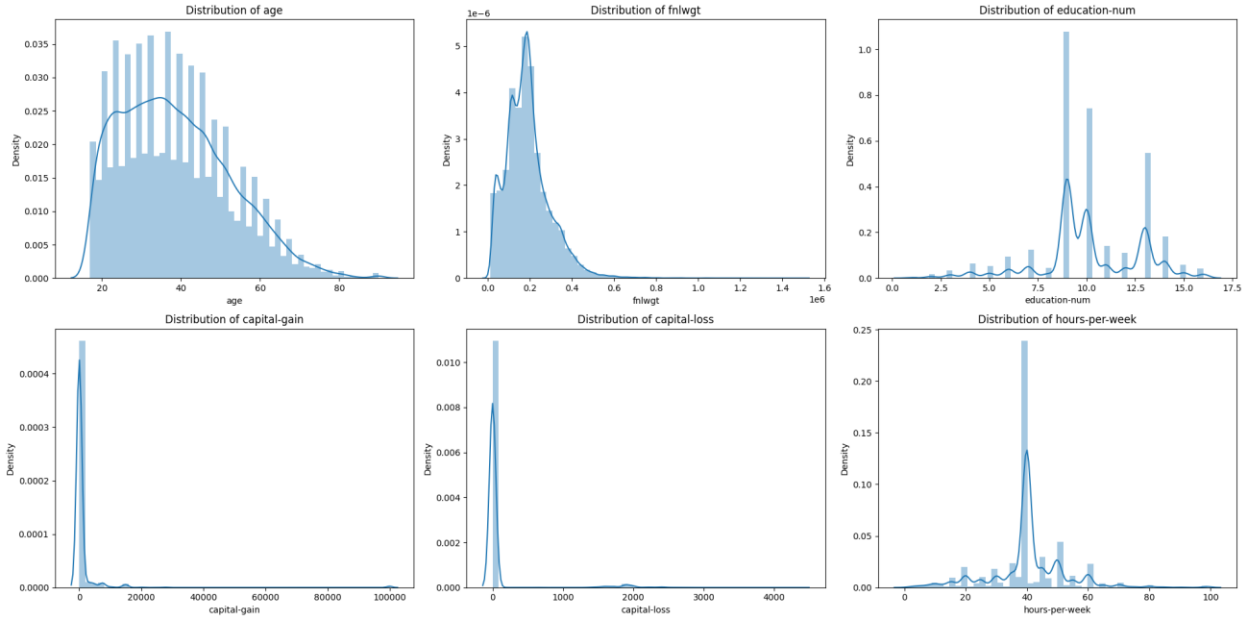

Figure 3.1.1. Distribution Plot of all Numerical variables

II. Distribution of categorical features

- **High-Cardinality Issue:** native-country has a large number of unique categories, with the United States dominating. This requires Feature Engineering (combining rare categories into an "Other" category) to improve the CTGAN's ability to model the distribution and prevent mode collapse for rare values.
- **Dominant Categories:** Features like workclass (Private) and race (White) have heavily dominant categories. The model must accurately represent these dominant modes while still generating realistic data for smaller categories.

- **Class Imbalance**: The income countplot visually confirms the severe imbalance: ~76% are <=50K$ and ~24% are $>50K$. CTGAN must handle this imbalance during training.

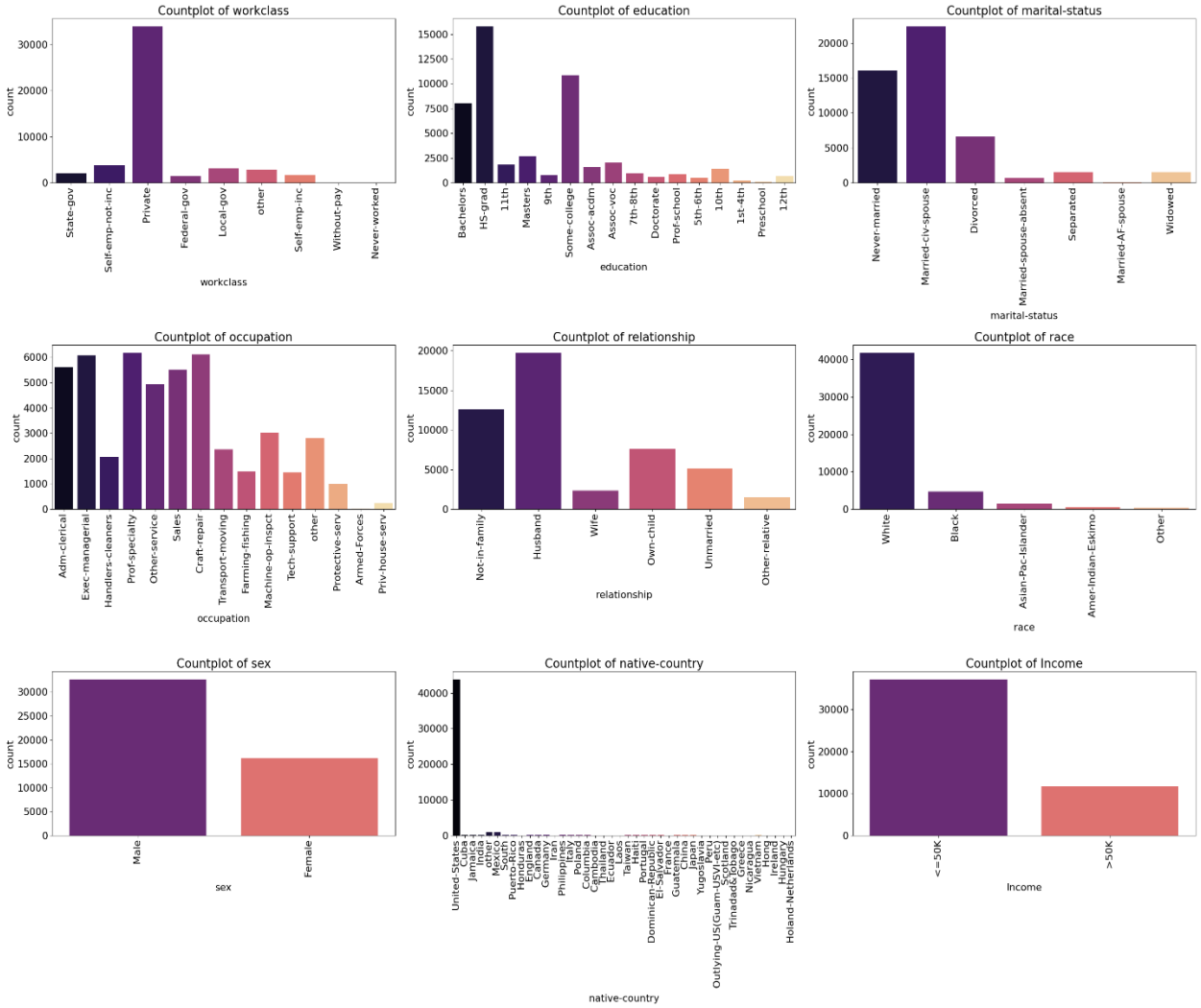

Figure 3.1.2. Distribution Plot of all Categorical variables

III. Conditional distribution of Numeric variables with Income

- **Clear Feature-Target Dependence:** There are strong visual differences between the two income classes (<=50K$ and >50K$) across several features. For example, individuals earning >50K$ have a significantly higher mean age, education-num, capital-gain/loss, and hours-per-week compared to the low-income group.
- **Crucial Correlation Preservation:** This plot directly visualizes the need for the Generative AI model to preserve the correlation between features and the target variable11. For instance, a synthetic record with a high education-num must have a statistically higher chance of having a >50K$ income.
- **Class Imbalance Context:** While the imbalance is best seen in a countplot of Income alone, this plot demonstrates that the model must be able to generate synthetic data that accurately reflects the conditional distribution of features given the income class.

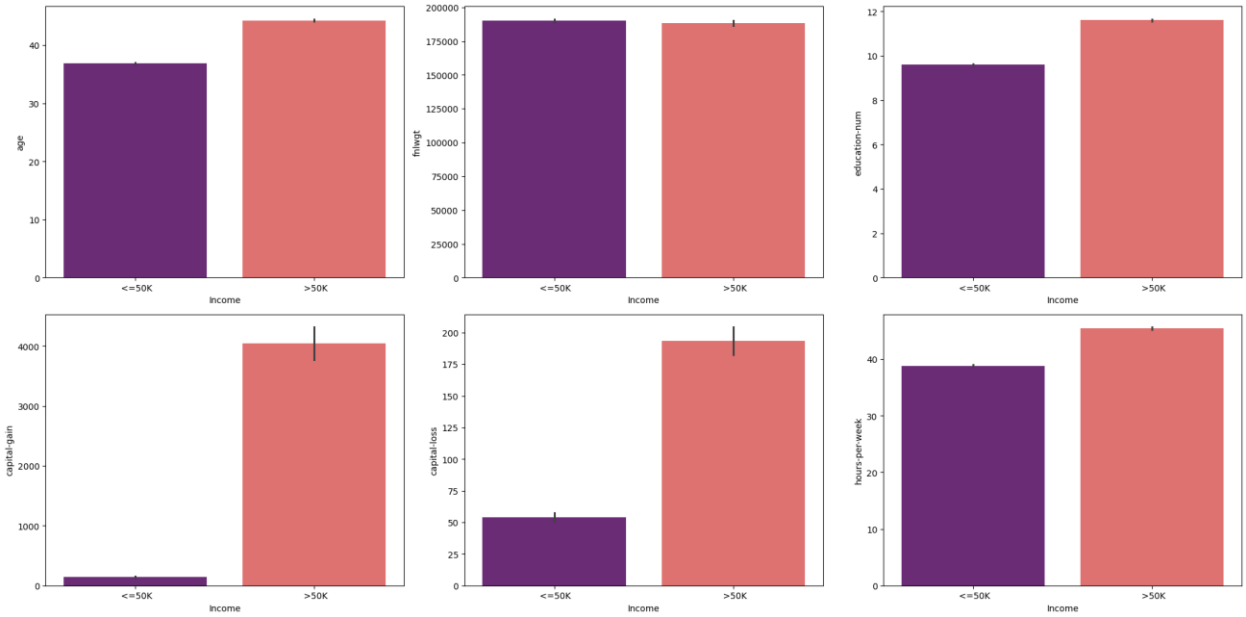

Figure 3.1.3.a Bar Plot Showing Numerical variables v/s Income

- **Mean/Median Shift:** The median age, education-num, and hours-per-week are notably higher for the >50K$ group compared to the <=50K$ group. The CTGAN must capture this shift in distribution when generating high-income records.
- **Outlier Confirmation:** The boxplots confirm that capital-gain, capital-loss, and fnlwgt contain extreme outliers.
- **Data Sparsity:** For capital-gain and capital-loss, the <=50K$ boxplots are highly compressed, reinforcing the treatment plan to use Log-transformation to better model these skewed features.

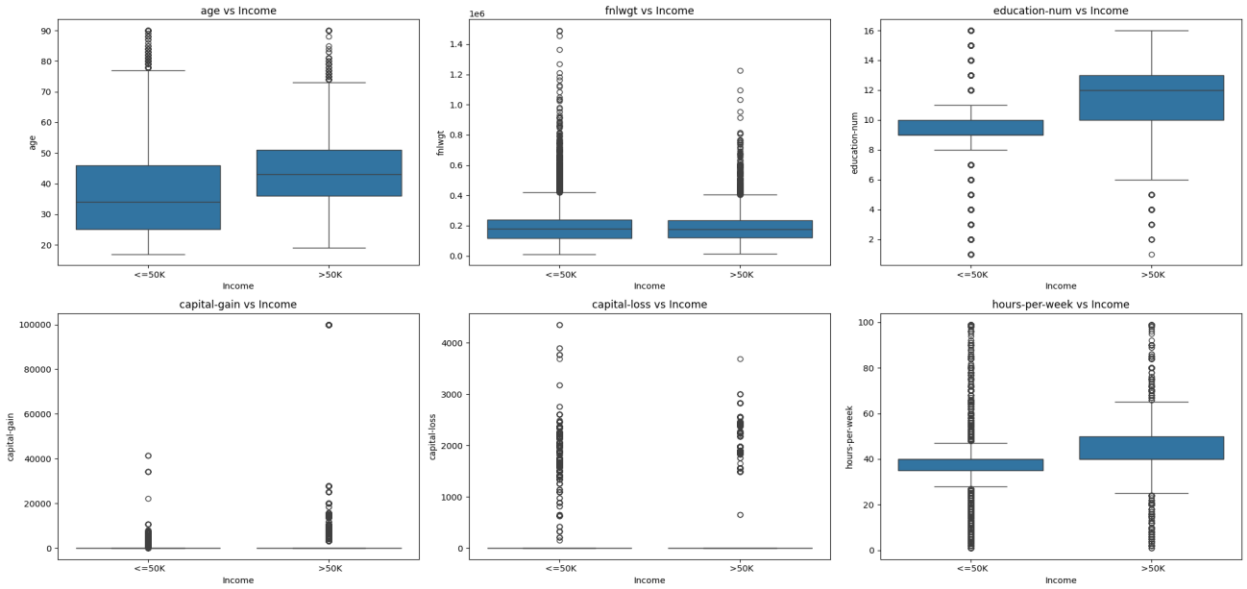

Figure 3.1.3.b Box Plot Showing Numerical variable v/s Income

IV. Conditional Distribution of all Categorical columns with Income

- **Strong Predictive Power:** Features like education (Bachelors, Masters, Prof-school) and relationship (Husband) show a disproportionately high count in the >50K$ category compared to their overall frequency. The synthetic data must preserve these conditional probabilities.
- **Clear Separation:** sex shows that while males are more numerous in both categories, a much higher proportion of males achieve >50K$ compared to females. The generative model must learn this gender-income correlation.
- **Sampling Bias**: Features like native-country have very few >50K$ records outside of the US. The model needs adequate training data to synthesize these rare high-income instances correctly.

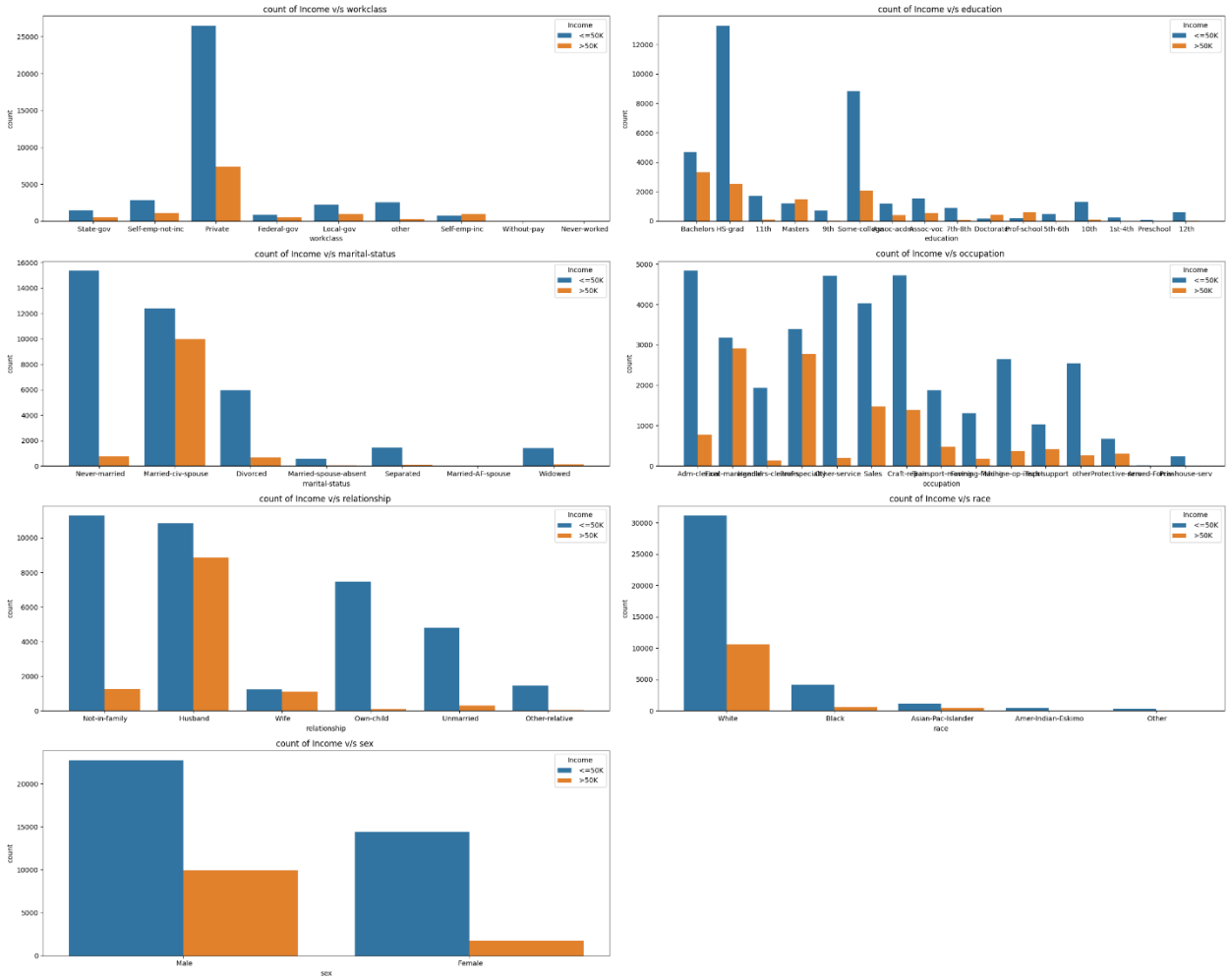

Figure 3.1.4. Bar Plot Showing Categorical columns v/s income

3.2. Check for Multi-collinearity

The perfect collinearity between education and education-num has been noted. For standard predictive models, a VIF (Variance Inflation Factor) check is recommended. For generative modeling, one of the redundant columns will be removed during pre-processing.

Figure 3.2.1 Heatmap of Numerical variables

The correlation heatmap (Figure 3.2.1) confirms that most numeric variables have a very weak linear correlation (values near 0).

- Redundancy: The perfect collinearity between education and education-num is noted. The column education will be removed during pre-processing.
- CTGAN Requirement: Since linear correlation is weak, the Generative AI model must capture complex, non-linear dependencies.

3.3. Outlier Detection & Treatment

Outlier Presence: Extreme spikes are observed in capital-gain and capital-loss (highly skewed). Outliers are also present in hours-per-week (e.g., 99 hours).

Visualization:

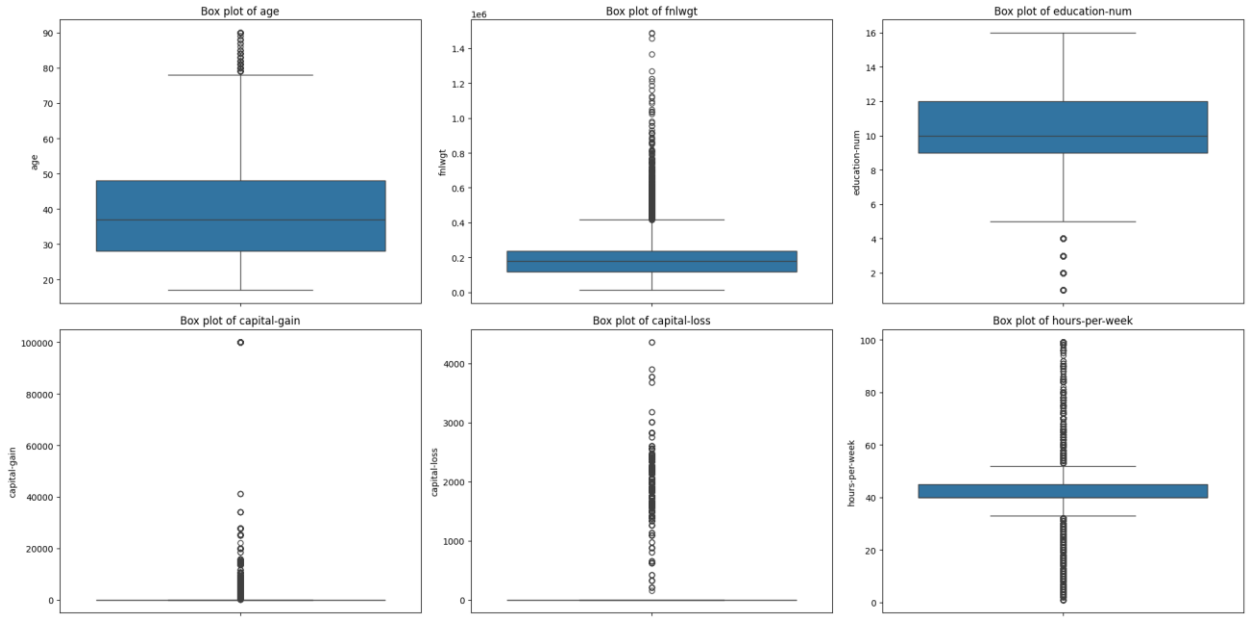

Figure 3.3.1 Box Plot of Numerical variable for outliers

Treatment: Log-transformation is planned for capital-gain/loss to normalize the highly skewed distribution. **Winsorization** is an option for hours-per-week if necessary.

3.6. Class Imbalance

The target variable income has a significant imbalance (~$76% vs ~$24%).

- **Generative AI Treatment: CTGAN** is known to automatically handle class imbalance during its training process, making explicit oversampling (like SMOTE) unnecessary for the generation task itself.

4. Feature Engineering

Step	Transformation/Status	Notes
Transformation	Log-transform for capital-gain, capital-loss. Combine rare categories in native-country.	Essential for models to handle skewed distributions.
Scaling	Required for numeric variables before modeling.	CTGAN handles raw data, but normalization generally improves training stability.
Feature Selection	NOT APPLICABLE	All features must be preserved to maintain the original data structure and correlations.

5. Methodology to be Followed (Phase 2)

5.1. Model Selection and Justification

- Selected Model: CTGAN (Conditional Tabular GAN).
- Justification: CTGAN is specifically designed for mixed-type tabular data. It uses a conditional generation process to handle the diverse distributions (numeric and categorical) and complex correlations present in the Census Income dataset.

5.2. Detailed Pre-processing Pipeline

The pipeline transforms the mixed-type data into a network-ready format:

1. **Redundancy Handling:** Drop the education column to eliminate perfect collinearity with education-num. fnlwgt may be optionally excluded.
2. **Categorical Encoding:**
    - **Standard OHE:** Applied to standard categorical features (workclass, sex, etc.).
    - **High-Cardinality Treatment:** The native-country rare categories will be **combined** into an "Other" category to prevent training instability.
    - **Target Encoding:** The binary income column is converted to 0/1.
3. **Numerical Transformation:**
    - **Logarithmic Transformation:** Apply $log(x+1)$ to the highly skewed capital-gain/loss to normalize the distribution.
    - **Scaling:** All final numeric variables will be normalized (using MinMaxScaler) to ensure balanced contribution to the GAN's loss function.

5.3. Training Architecture

The project uses the **CTGAN** architecture, a specialized Generative Adversarial Network designed for mixed-type tabular data.

- **Generator:** Learns to map random noise and a conditional vector to synthetic data. Its goal is to generate samples indistinguishable from the real data.
- **Discriminator:** Acts as a binary classifier, learning to distinguish between real and synthetic data.
- **Goal:** Training continues until the Generator successfully fools the Discriminator, indicating that the synthetic data is statistically similar to the real data.

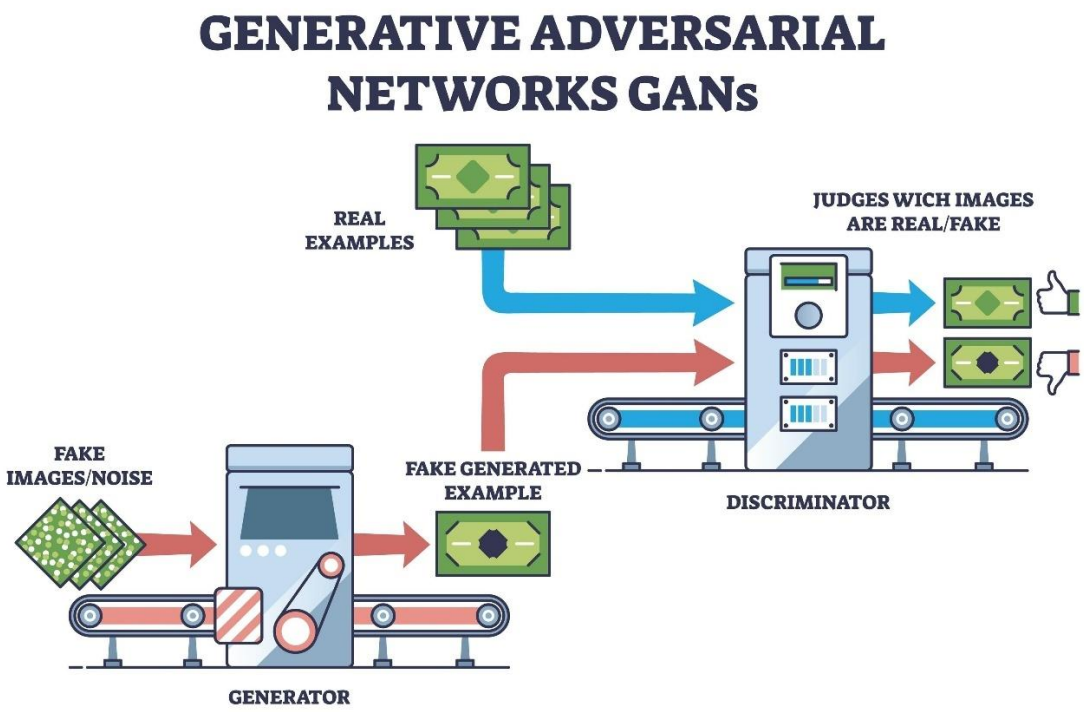

Figure 5.3.1 GAN-Generative Adversarial Network Training Architecture

5.4. Utility and Privacy Evaluation Metrics

Validation will be performed using standard metrics:

- **Data Utility:**
    - **TSTR (Train on Synthetic, Test on Real) Score:** Measures the predictive performance of a downstream model (e.g., XGBoost) trained on synthetic data but tested on real data.
    - **SDMetrics / KS-Test:** Compares the marginal and pair-wise distributions between the real and synthetic datasets.
- **Data Privacy:**
    - **Distance to Closest Record:** Quantifies the privacy risk by assessing the likelihood of membership inference.

Appendix

i.	AI	Artificial Intelligence
ii.	API	Application Programming Interface
iii.	CPS	Current Population Survey
iv.	CSV	Comma-Separated Values
v.	CTGAN	Conditional Tabular GAN
vi.	EDA	Exploratory Data Analysis
vii.	GAN	Generative Adversarial Network
viii.	GDPR	General Data Protection Regulation
ix.	GPT	Generative Pretrained Transformer
x.	HIPAA	Health Insurance Portability and Accountability Act
xi.	JSON	JavaScript Object Notation
xii.	KL	Kullback–Leibler Divergence
xiii.	LL.M / LLM	Large Language Model
xiv.	ML	Machine Learning
xv.	SAP	Systems, Applications & Products in Data Processing
xvi.	SDV	Synthetic Data Vault
xvii.	SDMetrics	Synthetic Data Metrics Framework
xviii.	KS-Test	Kolmogorov–Smirnov Test
xix.	TSTR	Train on Synthetic, Test on Real
xx.	TRTS	Train on Real, Test on Synthetic
xxi.	TVAE	Tabular Variational Autoencoder
xxii.	UI	User Interface

****