Saama Interview Problem

Name:	Alexander Minch
Company:	Saama
Project Title:	Synthetic Thyroid Data Generation
Supervisor:	Tamil Arasan

1 Problem Statement

Given a small real Electronic Health Record (EHR) table (demographics, vitals, labs) and you can choose any dataset from the UC Irvine Machine Learning Data Repository. Select any tabular-data synthesis framework or library, fit a generative model to the real data, and sample 1,000 synthetic patient records. For evaluation, compute column-wise statistical tests and a privacy metric such as PCA-based distance between real and synthetic data or record-matching risk.

Deliverable: A script/notebook that shows your framework choice, model training code, sampling code, and evaluation results. Write also brief commentary on synthesis quality and privacy trade-offs.

2 Solution

2.1 Synthetic Data Generation

Ultimately, the framework I chose was the Synthetic Data Vault (sdv) as it includes a variety of synthesizers, useful to me for testing. In choosing a dataset, I looked for a relatively clean and preprocessed set with few null values so that the generated data would avoid unwanted noise. The set that fit these parameters was a Thyroid Disease Dataset from the Garavan Institute. The synthesizer I chose was the Gaussian Copula, as it works well with continuous numerical data, and is standard for small datasets like the one I chose. Below is the code fragment for modeling and fitting the synthesizer.

```
import pandas as pd
from sdv.metadata import SingleTableMetadata
from sdv.single_table import GaussianCopulaSynthesizer

ds = pd.read_csv("new-thyroid.data", header=None)

ds.columns = [
    "class",
    "T3_uptake",
    "thyroxin",
    "triiodothyronine",
    "basal_TSH",
    "TSH_response"
]

metadata = SingleTableMetadata()
metadata.detect_from_dataframe(data=ds)
metadata.update_column(column_name='class', sdtype='categorical')
synthesizer = GaussianCopulaSynthesizer(metadata)
synthesizer.fit(ds)
synthetic_data = synthesizer.sample(num_rows=1000)
```

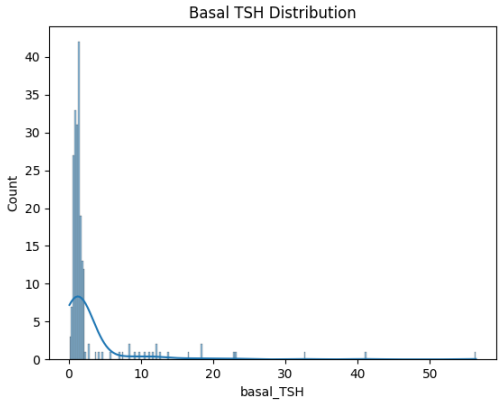
2.2 Evaluation Results

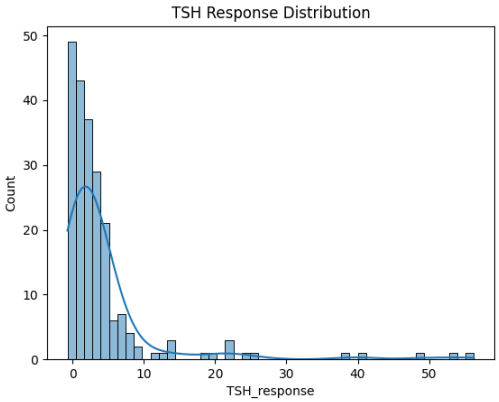
The performance of the Gaussian Copula on the dataset was relatively successful. In SDV's own data quality test, the generated set scored 86.14% on column shapes score and 84.58% on column pair trends, for an average of 85.36%. To more specifically analyze the results, I ran my own tests on the columns using the χ^2 test

for categorical columns and KS (Kolmogorov–Smirnov) for continuous. The synthesis quality of each column, based on those statistical similarity tests between the real and synthetic data, is summarized below:

Column	Test	p-value
class	χ^2	0.7565
T3_uptake	KS	0.3200
thyroxin	KS	0.1502
triiodothyronine	KS	0.0102
basal_TSH	KS	0.0000
TSH_response	KS	0.0000

This shows more clearly the underperformance of the GaussianCopula synthesizer, especially for the `basal_TSH` and `TSH_response` columns. The Gaussian model performs better when data follows a normal distribution, however after graph analysis both columns show positive skew. Possibly a log transformation of the columns to make their distributions less extreme would have contributed to better synthesis quality.





For the privacy assessment tests, I used PCA and Euclidean distance to calculate the average nearest neighbor from a synthetic point to a real data point. The test yielded an average of 0.4166 units synth-to-real. As this is a smaller dataset, this smaller number is largely expected. So for comparison, I did a simple NN on the real data set itself, which returned 0.2803 real-to-real. Thus the synth-to-real distance is 48.6% higher, a healthy margin that is unlikely to result in the leakage of personal information. This result supports a favorable privacy–utility trade-off, the synthetic data achieves fair similarity to the original dataset without replicating individual records too closely. This is a key requirement for responsibly sharing or analyzing synthetic data, especially in sensitive domains like healthcare.