Synthetic Ads Clicks Data: TVAE, CTGAN & LLMs

Team A: Hengyuan Liu, Hongze Lyu, Timothy Tu, Yuki Urata

Data Overview/Preprocessing

- **Original Data:** 7,675,517 rows, 35 features (train_data_ads with **1.56GB)**
- **Remove:** 17 features have exceed 100 unique values and (site_id) with same value
- **Current Data:** 7,675,517 rows, 17 features (train_ads_minimal with **322.8MB)**
- **Imbalance Issue:**
  - **Label 0 (No Clicks):** 7,556,381 (98.45%)
  - **Label 1 (Clicks):** 119,136 (1.55%)
    1. **Label 1 Data**: 119,136 rows, 17 features (train_ads_minimal_label1 with **5.2MB**), which we will use on training and synthesizing

Roadmap

- Use **TVAE** synthesize 119,136 Label 1 (Clicks)
- Use **CTGAN** synthesize 119,136 Label 1 (Clicks)
- Use **GReaT** synthesize 119,136 Label 1 (Clicks) by **Distil GPT2** and **GPT2**
- **Fidelity**
  - Data Density Distribution
  - JSD Score
  - Classifier-Based (Logistic)
- **Utility**
  - XGBoost (10-fold CV: Accuracy, F1 Score, Recall, Precision, ROC-AUC)
  - Check top 5 Feature Importance for each model (Gain Score)

Problem Statement

1. How were the each Synthesizers performance on Utility and Fidelity?
2. How the prediction change in ads click context?
3. Which is better?

CTGAN and GReaT

- **TVAE (Tabular Variational Autoencoder)**
  - Variational Autoencoder designed for tabular data with mixed types.
  - Learns a flexible latent space for high-fidelity synthetic sample generation.
- **CTGAN (Conditional Tabular GAN)**
  - Conditional GAN designed for tabular data with mixed types.
  - Captures complex dependencies to generate high-fidelity synthetic samples.
- **GReaT (Generating Realistic Relational & Tabular Data)**
  - Utilizes large language models by treating data as token sequences.
  - Captures complex relationships within the dataset.
  - LLM: Distil GPT2 and GPT2, or other pretrained models from Hugging Face.

Experiment Setup

TVAE Experiment

- 10 Epoch: Completes 10 times full passes through the label 1 data.
- Time taken: **15 mins** with Colab Free GPU.

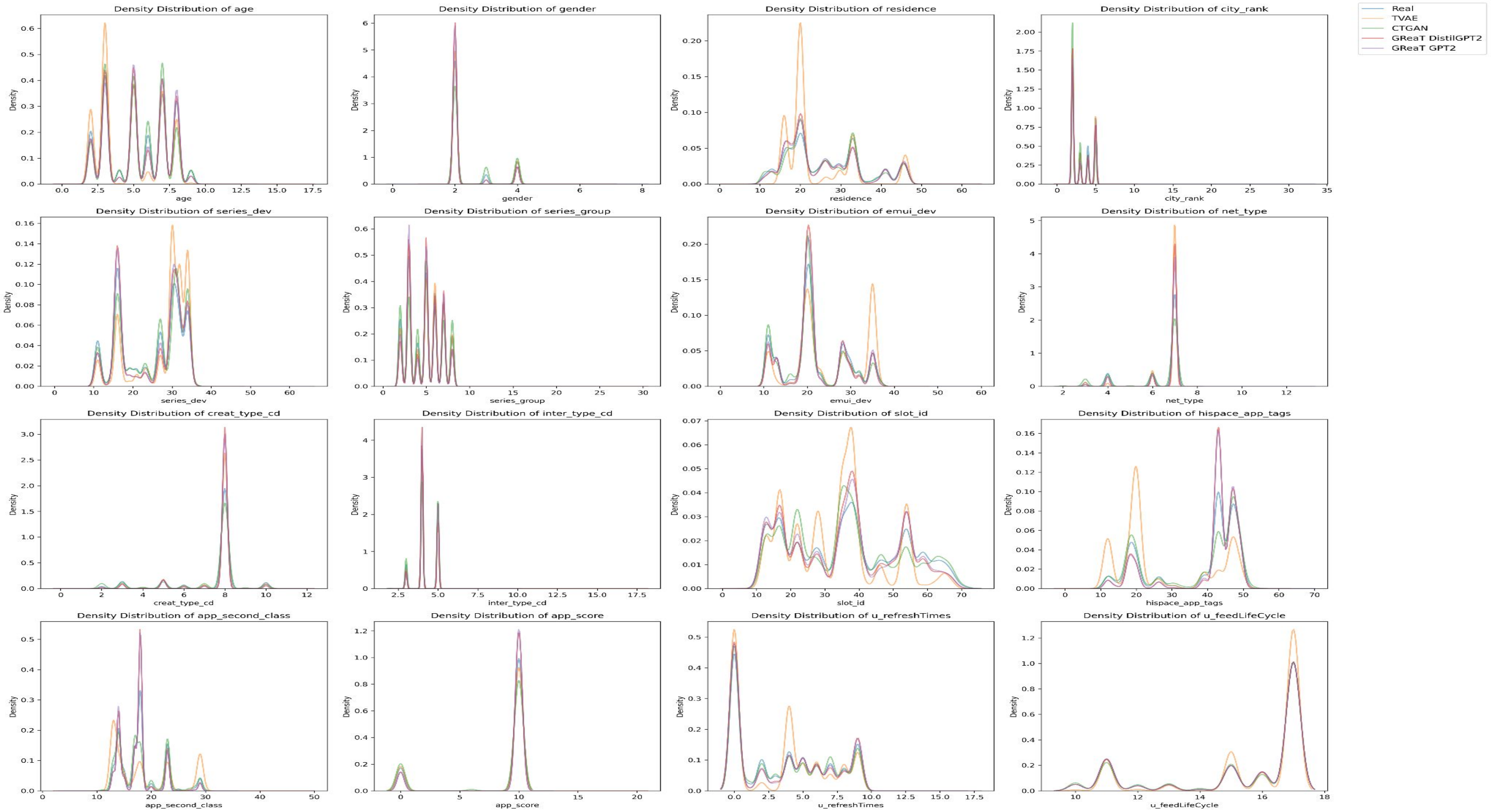
CTGAN Experiment

- 10 Epoch: Completes 10 times full passes through the label 1 data.
- Time taken: **9 mins** with Colab Free GPU.

GReaT Experiment (Distil GPT2 and GPT2)

- 2 Epoch and 64 Batch Size during training.
- Can not be directly generated 119,136 at once, so we use set 1000 batch size and generate about 119 times.
- Distil-GPT2 Total Time is **58 mins** with Colab pro L4 GPU (Model is 38 mins and synthesize is 20 mins).
- GPT2 Total Time is **80 mins** with Colab pro L4 GPU (Model is 60 mins and synthesize is 20 mins).

Fidelity - Density Distribution



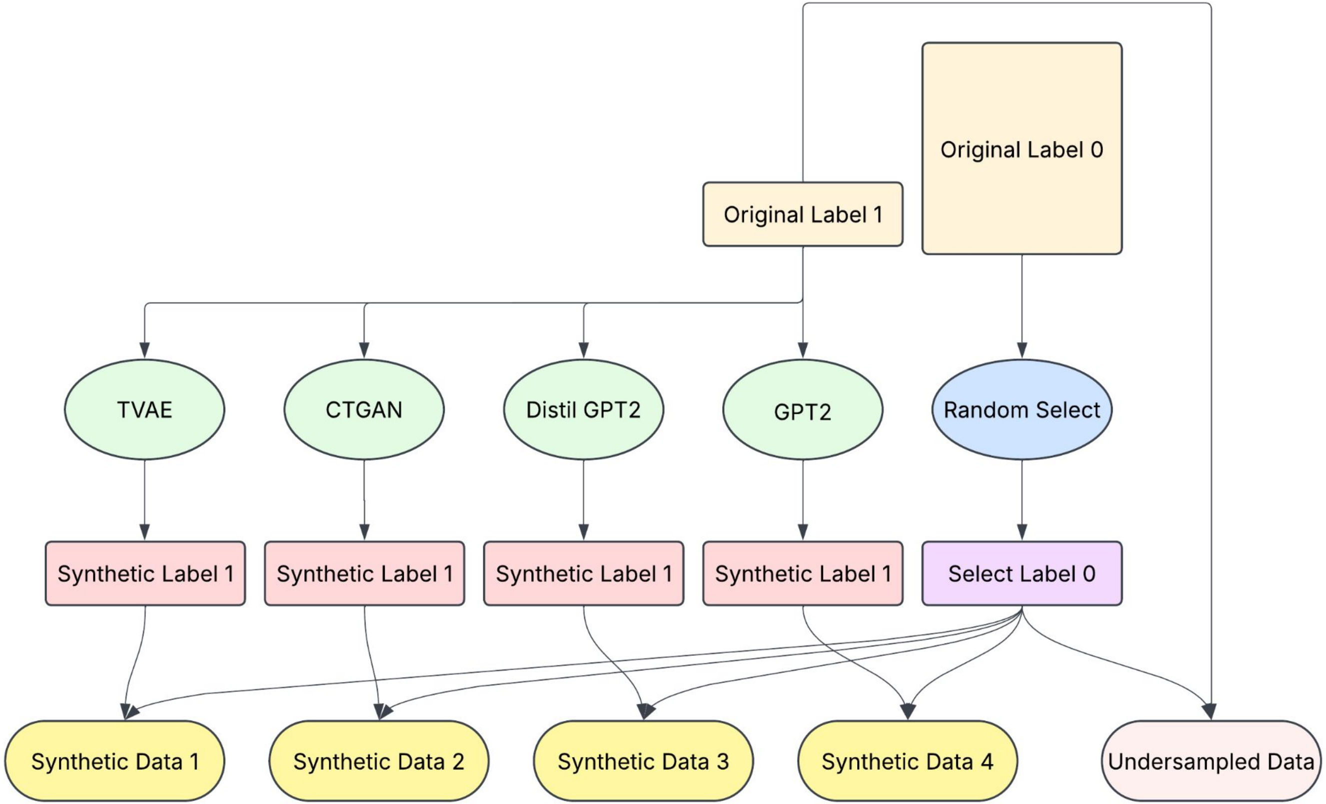
Fidelity - JSD and Classifier

- JSD: Jensen–Shannon divergence
- Classifier-Based (Logistic)

Logistic Regression	Classifier Accuracy
(Real vs. TVAE)	0.7392
(Real vs. CTGAN)	0.6047
(Real vs. DistilGPT2):	0.5952
(Real vs. GPT2)	0.5907

	Feature	CTGAN	TVAE	GReaT DistilGPT2	GReaT GPT2
0	age	0.077138	0.181051	0.068496	0.069658
1	app_score	0.070496	0.019761	0.046615	0.047919
2	app_second_class	0.152890	0.317083	0.122578	0.119174
3	city_rank	0.118697	0.055582	0.079165	0.059775
4	creat_type_cd	0.075454	0.071569	0.122139	0.108252
5	emui_dev	0.131999	0.235147	0.107088	0.089582
6	gender	0.087916	0.147340	0.087937	0.086044
7	hispace_app_tags	0.127757	0.383423	0.147048	0.140991
8	inter_type_cd	0.068092	0.038877	0.056481	0.049431
9	label	0.000000	0.000000	0.000000	0.000000
10	net_type	0.094496	0.147476	0.101385	0.078176
11	residence	0.103694	0.487610	0.096636	0.091456
12	series_dev	0.124746	0.225880	0.098572	0.091734
13	series_group	0.083806	0.031580	0.075091	0.076742
14	slot_id	0.136370	0.281559	0.123787	0.129316
15	u_feedLifeCycle	0.033410	0.116105	0.002241	0.002892
16	u_refreshTimes	0.043129	0.207437	0.069150	0.067120

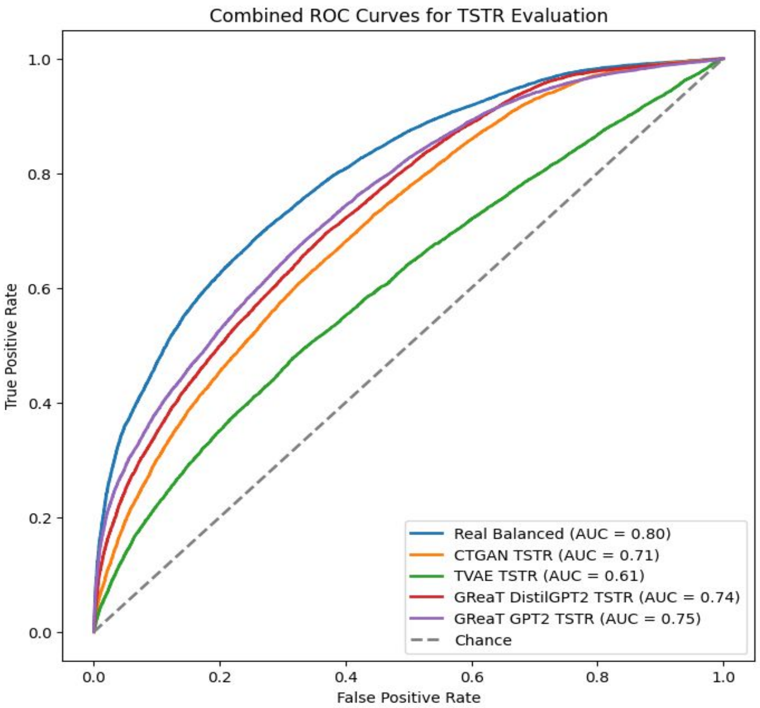
Data Training Process (TSTR)



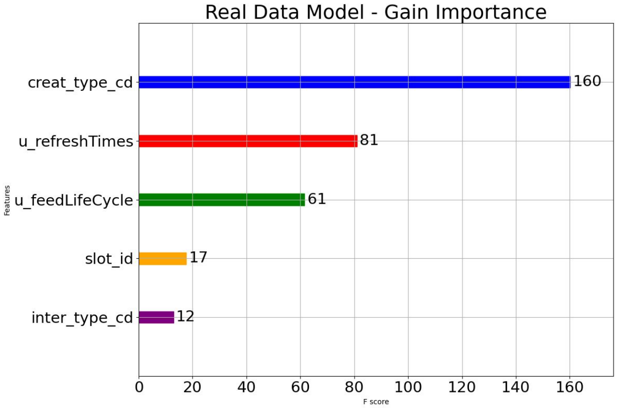
Utility - CTGAN vs. GReaT (10-Fold CV XGBoost)

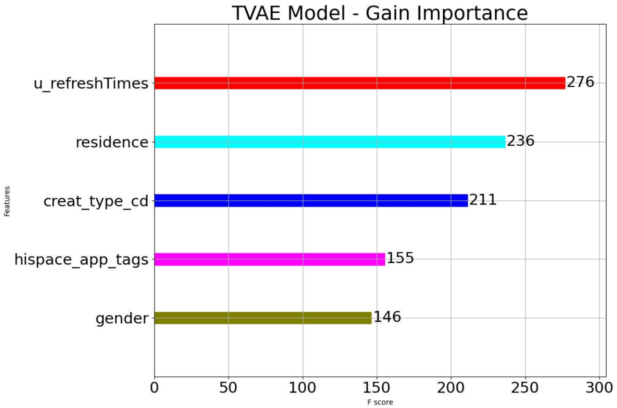
- TSTR (Train on Synthetic Data, Test on Real Data)

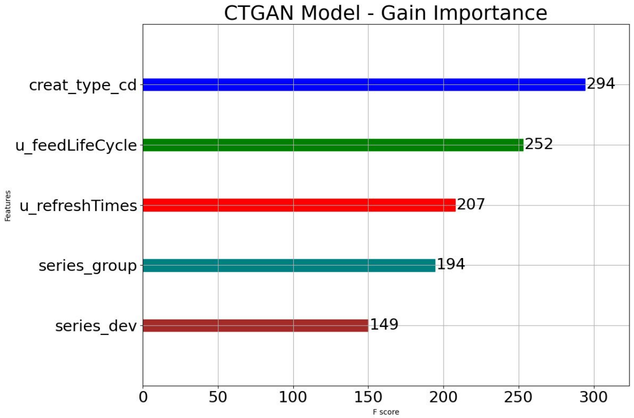
	Undersampled Data	TVAE	CTGAN	GReaT DistilGPT2	GReaT GPT2
Accuracy	**0.7149**	0.538	0.521	0.629	0.666
F1 Score	**0.710**	0.2040	0.094	0.499	0.614
AUC-ROC	**0.797**	0.612	0.710	0.739	0.752
Precision	**0.72**	0.742	0.874	0.771	0.727
Recall	**0.699**	0.118	0.050	0.368	0.531

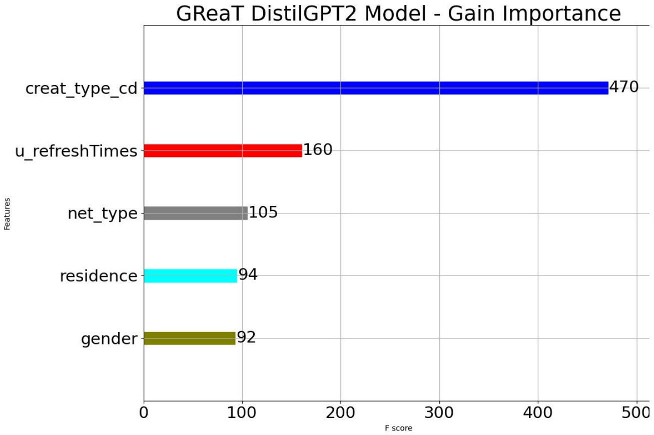


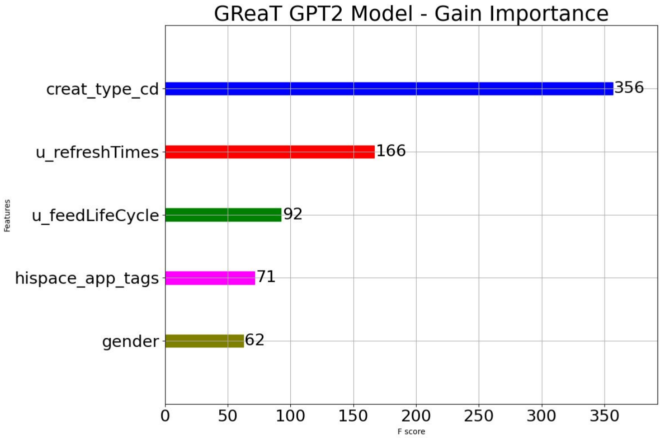
Top 5 Feature Importance - Gain Score











Conclusion & Limitations

- Fideility: All of the methods has similar distribution, JSD Score, and classifier accuracy, overall GPT models performed better than CTGAN.
- Utility: In XGboost, GReaT-GPT2 has the best performance.
- Top 3 feature importances from XGBoost are similar

- Try different LLM like Deepseek, Llama,...
- Try hyperparameter tuning like increase the epochs…
- Try different methods/models to evaluate the Fidelity, Utility and Privacy
- We tried REaLTabFormer to combine more data like feeds data and synthetic, but the computational was expensive

Thank You!

Q&A

References

https://github.com/kathrinse/be_great/tree/main
https://github.com/sdv-dev/CTGAN/tree/main
https://www.kaggle.com/datasets/xiaojiu1414/digix-global-ai-challenge/data
https://openreview.net/pdf?id=cEygmQNOeI
https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/tvaesynthesizer