STATS 414 Final Project

Generative Marketing Data Using Cross-Domain Information from Ads and News Feeds

Dan Knight, Sean Mulherin, Justin Huang

March 21, 2025

Abstract

The business of advertisement is a crucial aspect of the market, and this is clearly reflected in the magnitude of its financial footprint. This research sets out to employ modern statistical models to generate synthetic data pertaining to user interaction with online advertisements. Behavior characteristics of online users are utilized in conjunction with online feed characteristics in an effort to provide a holistic modeling approach. Data is provided by the 2022 DIGIX Global AI Challenge, where data from two previously separate sets are merged to describe the user and the user's online experience. One of the most problematic challenges of these data is the imbalance that exists within the binary response variable, *label*. Generative models such as SMOTE and Artificial Neural Networks are leveraged to synthesize data in the minority class. Synthetic data generation models are compared based on evaluation metrics as as fidelity, utility, and privacy.

Keywords: Neural network, SMOTE, oversampling, Gaussian mixture model, support vector machine, fidelity, utility, privacy

Contents

1 Introduction 2

1.1 Background 2

1.2 Motivation 2

1.3 Objective 2

2 Data 3

2.1 Description 3

2.2 Wrangling 3

2.3 Feature Engineering 4

2.4 Exploratory Data Analysis 5

3 Methods 7

3.1 Synthetic Data Generation 7

3.2 SMOTE 7

3.3 Generative Neural Network 8

4 Results 8

4.1 Fidelity, Utility, and Privacy 8

4.2 Visualization 9

4.3 Precision-Recall Curve 9

5 Discussion 10

5.1 Neural Networks 10

5.1.1 Autoregressive Data Generation 10

5.1.2 Generative Modeling 11

5.1.3 Multi-Model Approach 12

5.1.4 Unified Deep Learning Approach 15

5.2 Key Findings 15

5.3 Future Work 16

1 Introduction

1.1 Background

This paper discusses modern synthetic data generation techniques. The context within which such models will be tested is the prediction of advertisement click through rate (CTR). To model advertisement, or “ad”, performance, we consider the user’s interaction with the ad as an indication of success. Click through rate is a widely used metric to evaluate the efficacy of online advertisements. Ads are an integral aspect of modern business, as they act as the predominant avenue through which businesses interact with the general public. They provide opportunities for businesses to communicate with potential customers about the products and services they provide. Not only do businesses benefit from ad campaigns, but so too does the general public.

People in need of products and services rely on advertisements to learn about the market. Ads are didactic by nature in that they educate the audience about the products/services available. Ads inform the public about products as well as the current market conditions pertaining to that product. In this system, the public is informed about the cost of products while businesses compete to vie for people’s attention. This competition is exhibited through frequent advertisements administered by online platforms such as company websites, blogs, social media platforms, video games, news outlets, television, and more. Advertisements can be found on nearly all websites.

1.2 Motivation

The financial footprint of the ad market is astounding. In 2023, Alphabet, formally known as Google, generated approximately $240 billion in revenue from online ads (Bowman, 2024). This accounts for nearly 77% of the total annual revenue for Alphabet, which is currently the fifth largest publicly traded company in the world in terms of market cap (“Largest Companies by Marketcap”, 2025). A recent article published by the Financial Times stated that, for the first time ever, “global ad revenue is projected to exceed 1 trillion dollars”, with most of the money coming from leading tech companies, such as Apple, Meta, Alphabet, and Amazon (Thomas, Dec. 2024). The business of advertisement and marketing is a crucial aspect of the market, and this is clearly reflected in the magnitude of its financial footprint. Improving the statistical modeling capabilities of online marketing campaigns will aid in the overall business market.

1.3 Objective

This research sets out to employ modern statistical models to generate synthetic data pertaining to user interaction with online advertisements. Here, behavior characteristics of online users are utilized in conjunction with online feed characteristics in an effort to provide a holistic modeling approach, where data from two previously separate sets are merged to describe the user and the user's online experience.

Four generation methods are compared: KNN-SMOTE, SVM-SMOTE, GMM-SMOTE, and Artificial Neural Networks. To compare synthetic data generation models, we evaluate models based on their fidelity, utility, and privacy (see Section 4). In the forthcoming analysis, we build the aforementioned models with the goal of generating data for the minority class in such a way as to achieve high fidelity, utility, and privacy. Ultimately, increasing the representation of the minority class will aid in predicting CTR, revealing intriguing insights into digital marketing campaigns as well as synthetic data generation techniques.

2 Data

2.1 Description

Data have been provided by the 2022 DIGIX Global AI Challenge (Huawei, 2025). Provided are two data sets: ads and feeds. The ads data set predominantly relates to user demographics (i.e., age, gender, city, device name, device size, etc.). The feeds data set predominantly relates to user news feed (i.e., number of likes/dislikes, advertisement categories, phone price, etc.). As such, we consider the ads data set as the target domain while the feeds data set is the source domain. In total, there are 12,248,648 observations with 94,068 unique users. After merging the two data sets based on user ID, there exist 44 features.

2.2 Wrangling

One of the most problematic challenges of these data is the imbalance that exists within the response variable, *label.* Naturally, users click on advertisements sparingly. This imbalance is portrayed in Figure 1.

This strong imbalance is problematic because of the accuracy paradox, which describes the scenario in which a model learns to predict solely the majority response class because, in doing so, the model achieves high accuracy. Looking at both training and testing data, only 4.1% of users interact with advertisements. Hence, if they model is incentivized to prioritize accuracy, then it will achieve a high accuracy of 95.9%. Despite achieving a high accuracy, the model would lack useful predictive power, indicated by low precision.

Apart from the three aforementioned SMOTE-based oversampling methods for generating synthetic data, another method used to resolve the imbalance of the response variable is an initial strategic sampling technique wherein train/test/validation samples are randomly selected while ensuring equal proportions between the minority and majority classes. This practice of decreasing the number of samples in the majority class is referred to as *undersampling.* Cross-validating sets are formed to provide separate yet comparable subsets

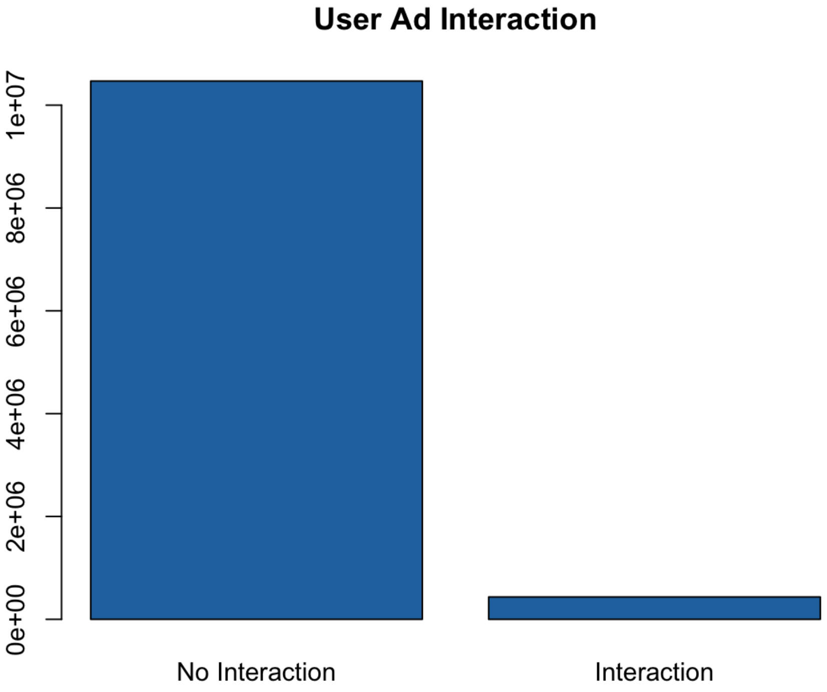


Figure 1: Imbalance in ad interaction, with about 98.5% of observations falling in the negative category.

of the data with which unbiased evaluation metrics can be calculated. In forming these subsets, we purposefully select random samples such that there is equal representation of either factor in the response class. Recall, *label* is our binary response variable with each factor representing whether or not a user interacts with an advertisement. The majority class is randomly undersampled to ensure equal representation with the minority class. The practice of undersampling is a popular and useful technique, but we will focus on data generation techniques.

2.3 Feature Engineering

Many features were carefully engineered in an effort to enhance the abilities of the original data. As you'll notice, many of the features are merely aggregates of a multitude of user interactions and advertisement categories. We engineered the following features:

- **Total Clicks**: total number of clicks performed by each individual user
- **Total Impressions**: total number of times an ad is exposed to the user
- **CTR**: The proportion of clicks performed by the user out of the number of impressions

- **Average Refresh Times**: the mean number the user refreshes their feed
- **Total Dislikes**: count of dislike/down votes from the user
- **Total Upvotes**: count of likes/up votes from the user
- **Unique News Categories**: splits the news categories by unique identifier
- **Most Common Category**: Find the mode of the unique news categories
- **Category Diversity**: the quantified diversity rating of category interest

2.4 Exploratory Data Analysis

Many initial plots are displayed to portray overall trends that are either insightful or problematic. First, we look at the correlation among the 44 features, shown in Figure 2 below.

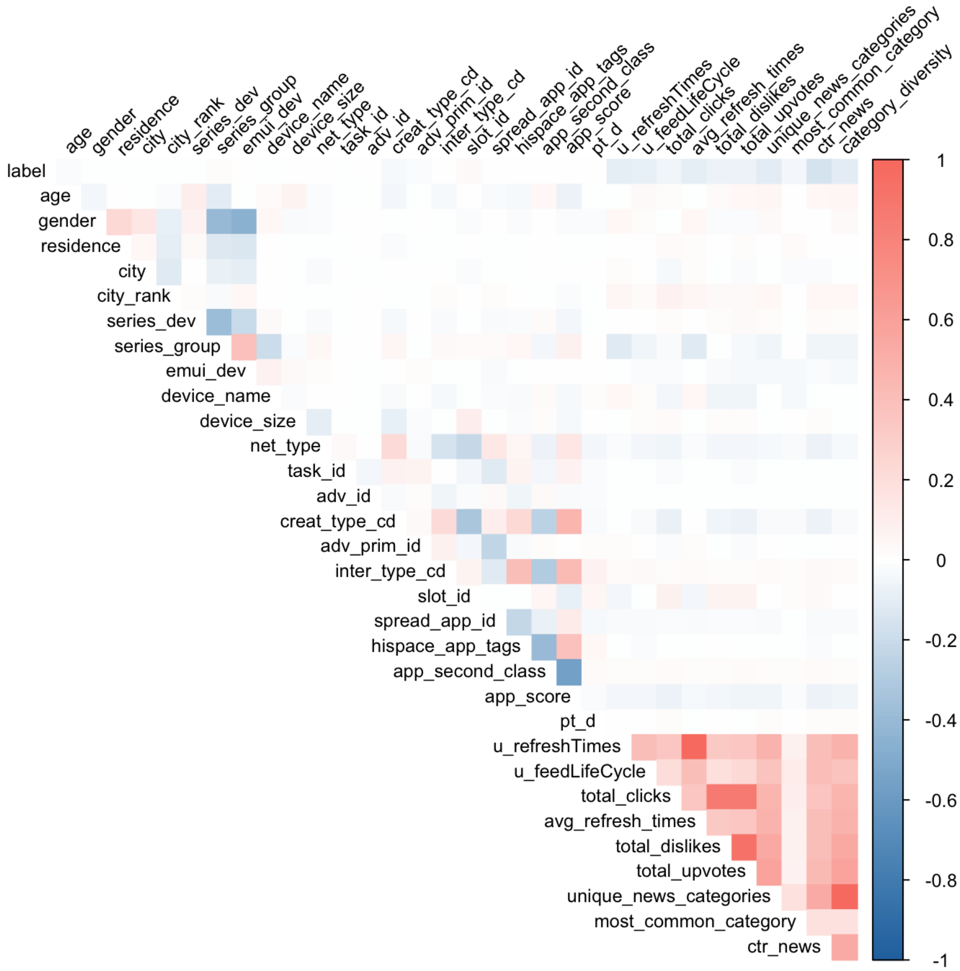
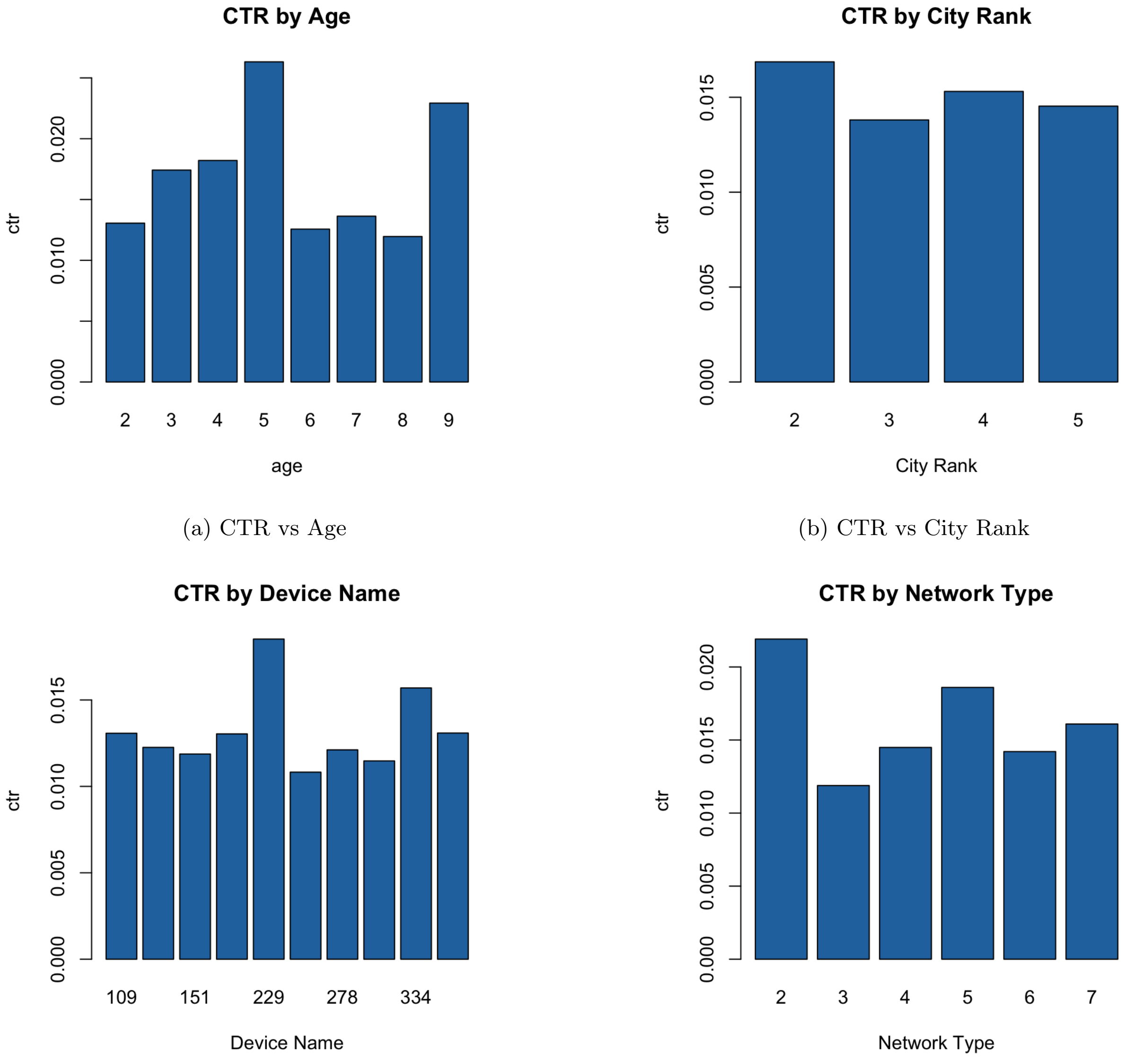


Figure 2: Correlation Plot

As shown, many of the factors that share high correlation coefficients are those that have been engineered to aggregate their associated factor. For example, *total_clicks* has an extremely high correlation with *total_upvotes* and *total_dislikes*, the variables from which *total_clicks* is calculated. Focusing on the response variable, *label*, we see meager correlation with most features. Interestingly, *label* is negatively correlated with *u_refreshTimes*, *u_feedLifeCycle*, *total_clicks*, *unique_news_categories*, and *ctr_news*.



(a) CTR vs Age

(b) CTR vs City Rank

(c) CTR vs Device Name

(d) CTR vs Network Type

Figure 3: Click Through Rate, as Compared to User Features

3 Methods

A myriad of advanced statistical methods are employed in the present research. Given the high dimensionality of the data and the diverse classes of features, modern modeling techniques that specialize in big data are the focus. Two methods investigated are the synthetic minority oversampling technique and artificial neural networks. Brief descriptions of each are provided below, after a brief introduction to synthetic data generation.

3.1 Synthetic Data Generation

Various methods are investigated to address the accuracy paradox, as introduced in Section 1.3. Synthetic data generation (SDG) describes the statistically rigorous practice of creating simulated data from models fitted to real data. With the recent explosion of big data, fueled by the groundbreaking discoveries in large language models such as ChatGPT and Gemini, SDG is experiencing rapid growth. According to Gartner, the use of synthetic data has surpassed the use of real data used in such models (Eastwood, 2023).

3.2 SMOTE

One of the main methods investigated in the present research is the Synthetic Minority Oversampling Technique (SMOTE) (Chawala, October 2020). In the most basic case of SMOTE, data are generated by selecting an observation in the minority class, fitting a K-nearest neighbors (KNN) to identify nearby observations belonging to the minority class, and finally choosing a random point between the original point and a KNN-classified point as your newly generated datum. This method works well for low-dimensional data, but here, with over forty factors, KNN is not powerful enough to correctly identify observations in the minority class. This scenario is often referred to as the *curse of dimensionality.* From this, it seems natural to explore more complex classification methods than KNN. Here, we offer two extensions of SMOTE.

First, by design, classification methods such as model-based clustering are better suited for high dimensional data. Specifically, Gaussian Mixture Models (GMM) offer a more rigorous classification method, as they are able to capture more complex shapes. GMM takes a Bayesian approach to estimate parameters and ultimately uses likelihood to classify data. This offers a more nuanced fit compared to KNN, which merely provides a deterministic approach where points are classified based on Euclidean proximity instead of distribution likelihood. One of the main drawbacks of GMM is the assumption of normality. As the name suggests, GMM assumes the underlying distribution to be Gaussian which may not necessarily to be true.

Second, supervised learning methods such as Support Vector Machines (SVM) provide an extremely flexible model to classify high dimensional data. SVMs identify classes by maximizing decision boundaries between classes. In higher dimensions, decision boundaries are in the form of hyperplanes which are governed by some predetermined kernel function.

3.3 Generative Neural Network

4 Results

4.1 Fidelity, Utility, and Privacy

Fidelity describes the compatibility between synthetic data and real data (Tao, 2025). As synthetic data ($\mathbb{Q}$) become more similar to real data ($\mathbb{P}$), fidelity increases. Utility describes how useful the synthetic data are for performing downstream tasks. Utility measures the accuracy of the model when compared to its fit on synthetic data versus real data. Here, a logistic regression model is used to evaluate utility. To achieve high utility, the accuracy of the model fitted to synthetic data should be similar to the accuracy of the model fitted to real data. Traditional evaluation metrics used to evaluate fidelity and utility include: propensity score, precision, recall, and propensity-recall density (PRD).

Propensity score is calculated by first training a logistic regression model to distinguish between $\mathbb{Q}$ and $\mathbb{P}$, then calculating the area under the curve (AUC) of the receiver operating characteristic curve (ROC). Hence, the propensity score is equal to the AUC ROC. An ideal score is 0.5, indicating that the model resorts to random guessing when making a distinction between the real and synthetic data. There are marginal differences between SMOTE models hovering around 0.62, with SMOTE SVM offering the best performance of 0.615 (see Table 1). This score indicates that the model is able to decipher the real data from the synthetic data with moderate accuracy.

Precision measures the proportion of synthetic point that have a real point nearby. High precision indicates similar distributions between $\mathbb{Q}$ and $\mathbb{P}$ Alternatively, recall measures the proportion of real points that have at least one synthetic point nearby. High recall indicates that all points in $\mathbb{Q}$ have a synthetic counterpart in $\mathbb{P}$. For both precision and recall, the threshold by which "nearby" is defined by implementing KNN on the real data. Precision-Recall Density (PRD) is defined as the harmonic mean of precision and recall, acting as a single summarizing metric to measure futility and utility. As shown in Table 1, SMOTE SVM offers the highest PRD score of 0.988.

Privacy describes the amount of real information available from the synthetic data (Tao, 2025). Not surprisingly, privacy is inversely proportional to fidelity. In this context, privacy refers to the anonymity of the online ad users. The more privacy a user is willing to sacrifice, the more relevant ads become, making the model more useful. One common metric to evaluation privacy is the Nearest-Neighbor Distance Ratio

(NNDR). To obtain the NNDR, k-nearest neighbors is implemented to find the first two nearest neighbors and their distances. Then, NNDR is calculated as the ratio between the distance to the nearest neighbor and the second-nearest neighbor. Higher NNDR values indicate higher privacy, indicating that the distances between points in $\mathbb{Q}$ and $\mathbb{P}$ are large and so difficult to leak. As shown, SMOTE GMM offers the highest privacy, with an NNDR score of 0.426 (see Table 1). This supports our previous claim about the inverse relationship between fidelity and privacy, as SMOTE GMM also has the lowest PRD score of 0.849.

Model	**Propensity Score**	**Precision**	**Recall**	**PRD**	**Privacy Risk (NNDR)**
SMOTE KNN	0.63	0.945	1.0	0.972	0.317
SMOTE SVM	0.615	0.977	1.0	0.988	0.207
SMOTE GMM	0.617	0.738	1.0	0.849	0.426

Table 1: Evaluation Metrics

4.2 Visualization

To visualize our data, we employ Kernel Density Estimation (KDE) to compare the synthetic data with the original dataset. Through our analysis, we identified several columns that are not relevant to the visualization process, such as login IDs. Instead, we focus on visualizing attributes that are most pertinent to our study: age, device size, app score, total impressions, average refresh times, total dislikes, total upvotes, and the number of unique news categories.

From our analysis, we observe that the density distribution of the synthetic data closely mirrors that of the original dataset, with both exhibiting peaks at similar locations. Among our models, K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) generate synthetic data distributions that are extremely close to the original data. Notably, the SVM-generated data is more concentrated around the median, which can be attributed to SVM's superior performance in terms of precision and recall.

4.3 Precision-Recall Curve

The Precision-Recall Curve (PRC) provides a graphical illustration of the tradeoff between precision and recall for our classification models. This visualization is particularly useful in evaluating how well synthetic data approximates the original dataset.

To construct the PRC, we use KNN to determine the distance of each synthetic data point from the nearest original data point, establishing a set of threshold values. By iterating through these thresholds, we compute precision as the fraction of synthetic points that fall within the threshold distance and recall as the fraction of original data points that are within the threshold distance of a synthetic point.

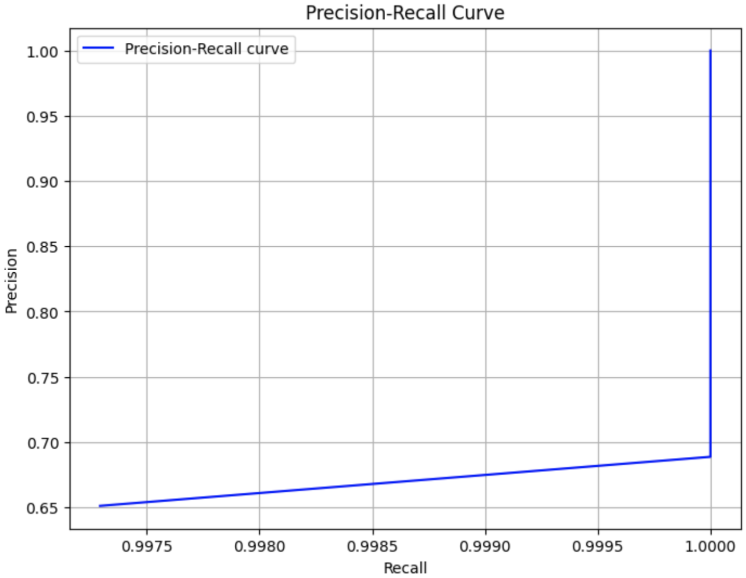


Figure 4: Precision-Recall Curve (PRC)

5 Discussion

5.1 Neural Networks

Neural networks, a class of machine learning models inspired by the structure and function of the human brain, have become a powerful tool in data analysis and pattern recognition. By learning from vast amounts of data, these models can identify intricate relationships and make predictions with remarkable accuracy. At the most basic level, neural networks describe a statistical model in which parameters work together in layers of neurons, weights, and activations, but their ability to model complex, non-linear patterns makes them particularly useful for tasks where traditional methods fall short such as generating new observations from complex datasets. Large neural network models have become standard throughout the field of generative AI and language modeling through the Transformer model architecture (Vaswani et al., 2017), which makes models such as ChatGPT possible (Brown et al., 2020). Here, we will detail our attempts to improve on SMOTE's data generation by leveraging neural network models from first principles.

5.1.1 Autoregressive Data Generation

Our concept is centered around the idea of *autoregression*, where the current value is predicted based on past values. Rather than using past observations like LLMs do, we use previously generated features. Our model predicts feature values one-by-one, feeding the new features back into the model input to predict the next feature until a full observation is created. First, p_1 will be generated randomly, independently of the others. Then, this generated feature is used as the input to generate p_2, essentially predicting the second feature given the previously generated feature $p_2|p_1 = x_1$. Then, this generated p_2 is passed (along with p_1) to predict $p_3|p_1 = x_1, p_2 = x_2$, and so on. Finally, the last p-th feature p_p is predicted as $p_p|p_1 = x_1, p_2 = x_2, p_3 = x_3, \ldots, p_{p-1} = x_{p-1}$. Essentially, this final prediction is the same as a more traditional predictive model which uses all other features to predict a single outcome (for example, predicting click-through rates using the other variables).

This approach is highly dependent on the order in which features are generated. The earlier features should be easier to predict and fit simpler, more straightforward distributions. The later, more complex variables will then have more information to better predict non-standard distributions and complex relationships with the previously generated variables. For example, using a combination of common sense, domain knowledge, exploratory data analysis, and experimentation, we began with the *city group*, *gender* and *age* of the user, then later moved onto the more complex *device name* variable. Essentially, this approach asks ”Which device is a male user between the ages of 26-30 from a tier-1 city most likely to use?”

5.1.2 Generative Modeling

To generate a full marketing observation, the model must handle both categorical and continuous variables. In a generative context, categorical variables are simple to generate by predicting a probability for each of the m possible classes using a softmax function. New observations can be generated by sampling from this predicted multinomial distribution. To train the model, cross-entropy loss was minimized to refine the predicted probability distributions.

Continuous variables, on the other hand, are less straightforward. While it is typical for neural networks to directly predict the value of a continuous variable (similar to a regression model), this deterministic approach will not work for generative models - the same inputs will always result in the same generated variables. Instead, the model must predict a distribution like in the categorical case. Specifically, the model must predict the parameters of a distribution (for example μ and σ for normal, or λ for exponential). Here, we used mostly the normal distribution, and sometimes log-normal when positive values were required - not far from a quadratic approximation approach. To train these parameter predictions, a negative log-likelihood function was minimized to predict the most likely pair of parameters given the other, previously predicted values. (Log-likelihood is chosen to simplify the calculation with many observations.)

The optimization functions used were:

Normal

$$-\log \mathcal{L}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \propto \frac{1}{n}\sum_{i=1}^{n} \log \sigma_i + \left(\frac{\log y_i - \mu_i}{\sigma_i}\right)^2$$

Log-Normal

$$-\log \mathcal{L}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \propto \frac{1}{n}\sum_{i=1}^{n} \log y_i + \log \sigma_i + \left(\frac{\log y_i - \mu_i}{\sigma_i}\right)^2$$

(Functions were simplified by removing scaling factors independent of μ, σ, or y which would not contribute to optimization. The mean loss of all observations is used to account for differences in sample size during training/testing.)

5.1.3 Multi-Model Approach

For simplicity, we began by training separate neural networks to predict each feature. This allowed models to be as simple or complex as needed given the requirements of each individual feature prediction problem. The simplest models were small, convolutional neural networks (CNNs). As demands increased, there were a variety of methods to increase model flexibility. First, dimensionality was increased with more hidden layers with larger dimensionality d_{model}. Concepts from the Transformer architecture (Vaswani et al., 2017) were then borrowed. Hidden layers were expanded using *feed-forward* layers, further increasing dimensionality. As input size grew, a *multi-head attention block* was added to allow information to be more flexibly communicated between previously generated features in the context. As in the Transformer, *residual connections* were added between sublayers to facilitate the flow of information through the models and stabilize the training process.

	$\mathbf{d}_{\text{model}}$	**Hidden Layers**	**Feed-Forward**	**Attention**
City Rank	1	0	No	No
Gender	4	0	No	No
Age	12	6	Yes	No
City	48	6	Yes	No
Unique News Categories	48	6	Yes	No

Table 2: Model architecture for the first 5 variables. (Note that models with 0 hidden layers are essentially logistic regression models.)

Generating data using information from actual observations was successful. The generated data closely followed the distributions of held-out test data for both categorical and continuous variables. This showed that if the data coming into each individual model was high-fidelity, the output also would be.

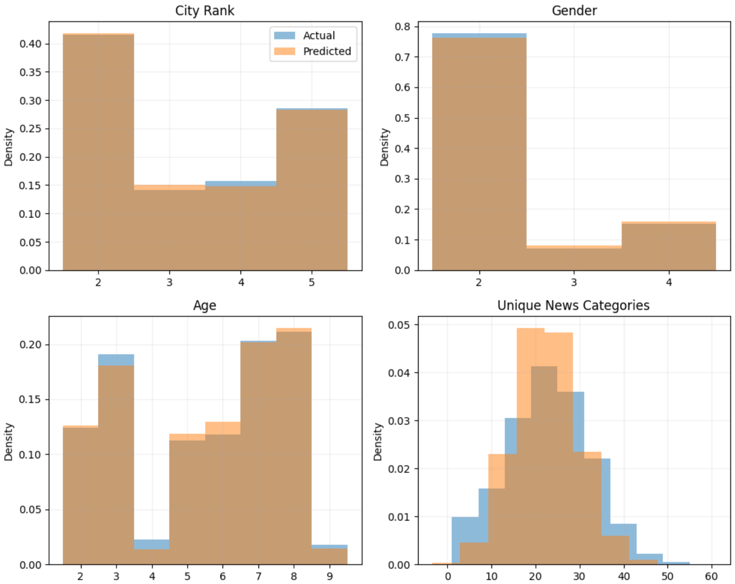


Figure 5: Comparison of test data vs. predicted distributions for *city rank, gender, age, and unique news categories.*

However, evaluating fully autoregressively generated data using standard fidelity, utility, and privacy metrics (propensity score, precision, recall, PRD, privacy risk, etc.) found disappointing results. In early generation steps p_1 and $p_2|p_1 = x_1$, the multi-model neural network approach outperformed all SMOTE variations, but performance quickly degraded across all evaluation metrics once more complex features were added.

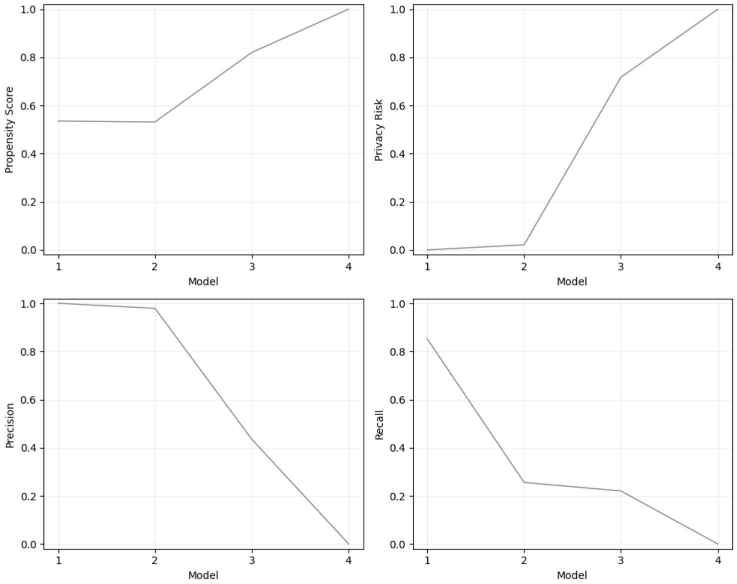


Figure 6: Multi-model performance measured after generating new observations of feature p_k. Therefore, model 1 contains p_1, model 2 contains p_1, p_2, and so on.

This was particularly evident when generating specific city predictions. The model performed well with training data, generating data which closely followed the many complexities of the true distribution. However, when using generated data as input, the model's predictions were largely uniform across the 100+ cities. This is poor generalization is likely caused by overfitting to the true data during training. It is possible that training on a blend of generated and actual observations will help the model generalize better.

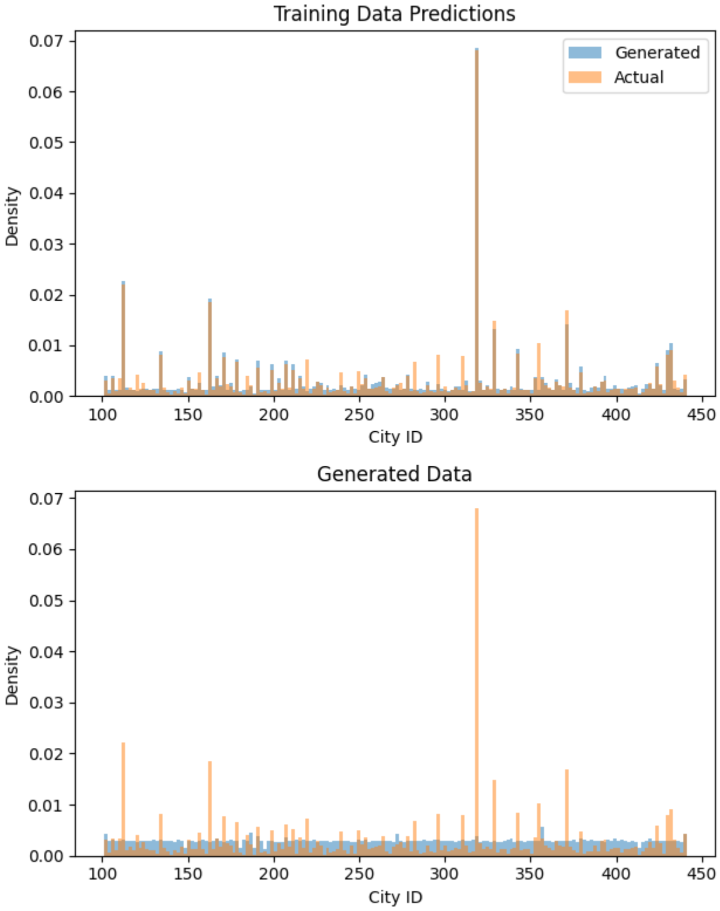


Figure 7: Comparison of *city ID* prediction using fully generated input vs. using training input.

These results could be improved by experimenting with the order of feature generation, and refining the design of each individual model through systematic hyperparameter tuning. However, both are prohibitive from a time and computation standpoint. Changes to feature order would require retraining multiple models, as the effects would propagate to the inputs of other models. Incorporate other design improvements such as dropout, batch/layer normalization, etc. would also require retraining multiple models.

5.1.4 Unified Deep Learning Approach

The multi-model approach was promising as a proof-of-concept for autoregressive data generation. Each feature has a discrete model, each of which defines a manageable sub-problem, and can be trained separately. However, there is a clear tradeoff with scalability and flexibility. These models are highly dependent, and the handcrafted nature of this system of models makes it highly specific to a single problem. This approach likely would not be able to make use of modern deep learning concepts such as *fine tuning*.

As an alternative, we propose a supervised single-model Transformer approach, where the context for each prediction will be the known (previously generated) features. Rather than the traditional LLM approach where previous observations inform the next prediction, each observation would be generated independently, with previous features informing the next features. Therefore, the set of valid context length $t \in \{0, 1, \ldots, p\}$ where p is the total number of features in the target dataset. Multi-headed attention blocks would handle differences in context size, coordinating information across the provided input features. In each pass, the model would generate a full observation with all features present. During generation, only the next feature would be used from each forward pass. This value would be appended to the input context to create more informed predictions at each step. In this way, there is no conceptual difference from the multi-model approach.

In practice, the model will need to train on all input lengths $t \in \{0, 1, 2, \ldots, p\}$. Each training loop will contain inputs of all sizes before back propagation, effectively expanding the training dataset from size n to size np. The model's dimensionality will need to be sufficiently large to learn the many complex relationships between variables while simultaneously supporting missing information with variable context lengths t. For reference, the *GPT-3 Small* model contains 125 million parameters with a model dimension $d_{\text{model}} = 768$ and 12 layers (Brown et al., 2020). With sufficient computing power and time, we expect this model to improve on the performance of both the multi-model neural network approach and the various SMOTE approaches.

5.2 Key Findings

Our study on SMOTE revealed that while our best-performing model, Support Vector Machine (SVM), enhances fidelity and utility, it does so at the expense of privacy. Moreover, SMOTE places greater emphasis on minority class data, and while it may be beneficial for imbalanced datasets, can introduce excessive noise. This results in lower precision and higher recall, potentially impacting the quality of synthetic data, especially when applied to balanced datasets.

The autoregressive neural network presented significant challenges, requiring substantial computational resources and extended processing time. Given our current limitations—relying on local machines and basic

Google Colab versions—scaling our approach remains a constraint. Additionally, running Precision-Recall Curve (PRC) evaluations was computationally intensive due to its nested loop structure, bring up the total runtime to approximately one hour per model.

5.3 Future Work

To enhance our methodology, we propose integrating diffusion models to mitigate noise and variance through a forward and reverse diffusion process. These models have already demonstrated effectiveness in healthcare applications for protecting patient privacy in tabular data. Implementing a similar approach in ad revenue analysis could provide insights into user privacy protection and address recent concerns regarding social media confidentiality.

For support vector machines, we plan to investigate alternative kernel functions through empirical testing to optimize performance. Moreover, in the case of Gaussian Mixture Models (GMMs), our current assumption of normality may lead to suboptimal results. Investigating transformations to improve the normality of skewed features is sure to provide additional insights.

For autoregressive neural networks, a key challenge is the inefficiency of using multiple models separately. A promising solution is to adopt a single unified model with global masked self-attention, which would enable efficient long-range dependency modeling while maintaining causality, potentially improving overall model effectiveness. By refining these approaches, we aim to enhance the trade-offs between privacy, fidelity, and computational efficiency, contributing to more robust synthetic data generation techniques.

References

Bowman, J. (2024). How much does google make in ad revenue? *The Motley Fool*. https://www.nasdaq.com/articles/how-much-does-google-make-ad-revenue

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://doi.org/10.48550/ARXIV.2005.14165

Chawala, N. (October 2020). Overcoming class imbalance using smote techniques. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

Eastwood, B. (2023). What is synthetic data — and how can it help you competitively? *MIT Management Sloan School*.

Huawei. (2025). Digix global innovation challenge. https://developer.huawei.com/consumer/en/activity/digixActivity/digixdetail/201655283879815928

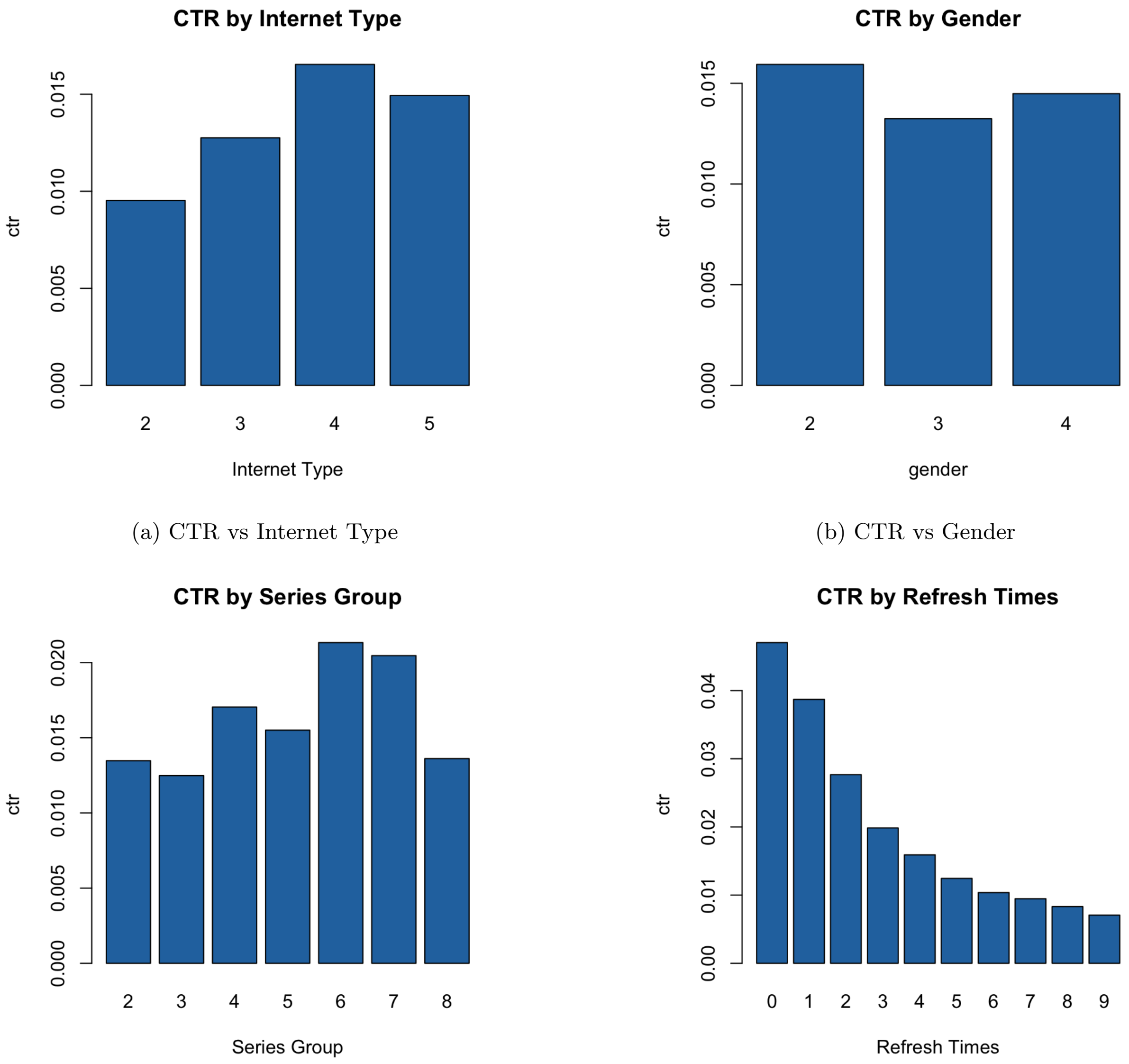
Largest companies by marketcap. (2025). https://companiesmarketcap.com/

Tao, L. (2025). *Introduction to the evaluation of generative data - fidelity, utility and privacy* (tech. rep.). Department of Statistics and Data Science, UCLA.

Thomas, D. (Dec. 2024). Advertising revenues set to hit $1tninMarketDominatedbyTechnologyCompanies$.. *Financial Times*. https://www.ft.com/content/e9d9befb-d5fd-438e-89d3-47f894c56736

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. https://doi.org/10.48550/ARXIV.1706.03762

Appendix



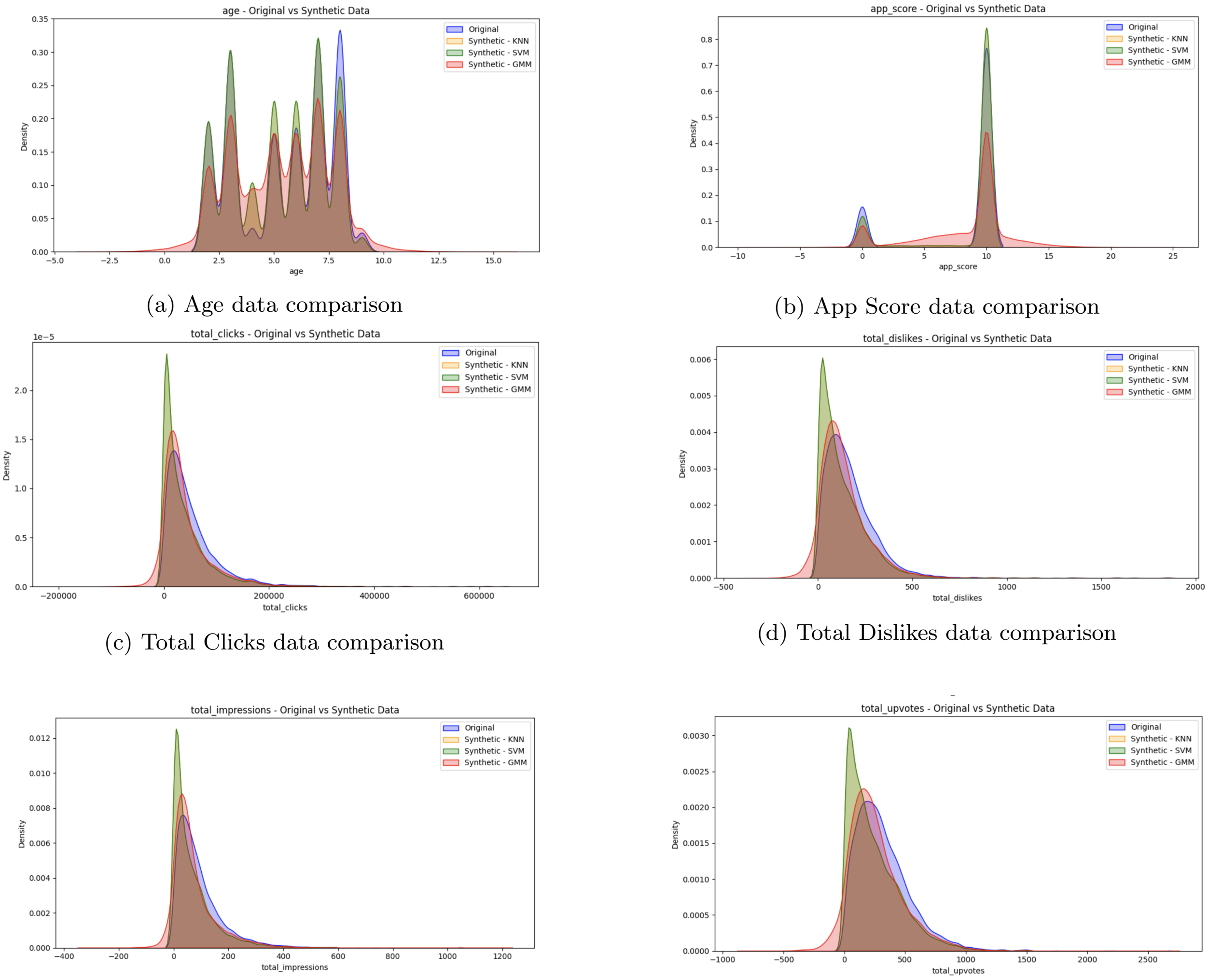
(a) CTR vs Internet Type

(b) CTR vs Gender

(c) CTR vs Series Group

(d) CTR vs Refresh Times

Figure 8: Click Through Rate, as Compared to User Features



(a) Age data comparison

(b) App Score data comparison

(c) Total Clicks data comparison

(d) Total Dislikes data comparison

(e) Total Impressions data comparison

(f) Total Upvotes data comparison

Figure 9: Synthetic vs Original Data Density Comparison