You need to be able to explain:

• What is discriminative vs generative AI?

• Why is there a need for synthetic data (in general)?

• Why is there a need for synthetic data in finance?

• What is data generation vs data augmentation?

• What models / techniques have been applied historically in synthetic data?

• What are the potential advantages/limitations of GANs and other similar techniques? (can you generate a comparison table?)

• How are GANs constructed? What is the Generator (noise vector)? What is the

Discriminator?

• What is the difference between GANs and TimeGANs?

• How does the whole adversarial training process look like?

• What are the metrics / success criteria when generating synthetic data?

• What is the loss function(s) used? \*\* (this is a nice point of research for timeseries)

• What tools/libraries/packages etc are there available to use? Would you use PyTorch or Tensorflow? Which out of the two provides the best implementations and why?

• Can we prove GAN-based approaches outperform other techniques like VAE, ….. etc?

• What are diffusion models?

Total Links

* <https://www.confident-ai.com/blog/how-to-generate-synthetic-data-using-llms-part-1>
* <https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them>
* <https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575>
* <https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce>
* <https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11>
* <https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd>
* <https://arxiv.org/pdf/2402.14042>
* <https://towardsai.net/p/l/gans-for-synthetic-data-generation>
* <https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a>
* <https://papers.nips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf>
* <https://github.com/eriklindernoren/Keras-GAN>
* <https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518>

**What is discriminative vs generative AI?**

Discriminative AI:

Discriminative models are designed to classify input data into predefined classes by learning the decision boundaries that separate these classes. They essentially model the conditional probability of the target variable given the input data (P(y|x)). For example, logistic regression predicts the probability that an input belongs to a particular class. Random Forests (RF) and Support Vector Machines (SVM) are also popular discriminative models. These models are used in various applications such as spam detection, image recognition, and medical diagnosis, where the goal is to categorize data accurately (Sources: [Confident AI Blog](https://www.confident-ai.com/blog/how-to-generate-synthetic-data-using-llms-part-1), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

Generative AI:

Generative models, on the other hand, aim to understand and model the entire distribution of the input data. They generate new data points that resemble the original data by modeling the joint probability (p(x,y)). Examples of generative models include Latent Dirichlet Allocation (LDA), Variational Autoencoders (VAE), and Generative Adversarial Networks (GANs). GANs are particularly notable for their ability to generate highly realistic images, videos, and even text. In GANs, a generator creates synthetic data, while a discriminator evaluates its realism, creating a feedback loop that improves the quality of the generated data over time (Sources: [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them), [Towards AI](https://towardsai.net/p/l/gans-for-synthetic-data-generation)).

The key difference between discriminative and generative models lies in their objectives: discriminative models are used for classification tasks, distinguishing between different classes in the data, while generative models are used for creating new data points that mimic the original data distribution, useful for applications like data augmentation and anomaly detection.

**Why is there a need for synthetic data (in general)?**

Synthetic data has become increasingly important for several reasons:

Privacy Preservation:

In many industries, especially healthcare and finance, data privacy is a significant concern. Synthetic data allows organizations to share and analyze data without compromising individual privacy. By replacing real data with synthetic counterparts, companies can comply with stringent privacy regulations such as GDPR and HIPAA while still leveraging valuable datasets for research and development (Sources: [Confident AI Blog](https://www.confident-ai.com/blog/how-to-generate-synthetic-data-using-llms-part-1), [Towards AI](https://towardsai.net/p/l/gans-for-synthetic-data-generation)).

Data Scarcity:

In some fields, collecting large amounts of real-world data can be challenging or expensive. Synthetic data provides a solution by creating additional data points that reflect the properties of the original dataset. This is particularly useful in training machine learning models where a large and diverse dataset can significantly improve model performance. For instance, in autonomous driving, synthetic data can be used to simulate rare but critical driving scenarios that might not be present in the collected data (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Towards AI](https://towardsai.net/p/l/gans-for-synthetic-data-generation)).

Cost Efficiency:

Generating synthetic data can be more cost-effective and time-efficient compared to the traditional data collection process, which involves data gathering, cleaning, and annotation. Synthetic data generation methods, such as GANs, can quickly produce large volumes of data, reducing the need for extensive manual effort. This efficiency makes it easier to iterate on model development and testing (Sources: [Towards AI](https://towardsai.net/p/l/gans-for-synthetic-data-generation), [Generating Synthetic Data with LLMs - Part 1 - Confident AI](https://www.confident-ai.com/blog/how-to-generate-synthetic-data-using-llms-part-1)).

**Why is there a need for synthetic data in finance?**

The financial sector presents unique challenges that make synthetic data particularly valuable:

Regulatory Compliance and Privacy:

Financial data often contains sensitive information, making it critical to comply with privacy regulations. Synthetic data helps by providing anonymized datasets that can be used for analysis and model training without exposing sensitive information. This enables financial institutions to innovate and improve their services while adhering to regulations like GDPR and the Gramm-Leach-Bliley Act (Sources: [Medium - Synthetic Data Generation Using GANs Part 2](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce), [Protect and Extend - Using GANs for Synthetic Data Generation of Time-Series Medical Records](https://arxiv.org/pdf/2402.14042)).

Risk Management:

Synthetic data can be used to create various hypothetical scenarios to stress-test financial models. This is crucial for assessing the robustness and resilience of financial strategies under different market conditions. For example, banks can use synthetic data to simulate economic downturns or market crashes and evaluate their risk management strategies without the need for real historical data (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)).

Fraud Detection:

Financial institutions need to continuously improve their fraud detection systems. Synthetic data can generate diverse and realistic fraudulent activity patterns that can be used to train machine learning models, making them more robust and effective in identifying and preventing fraud. By simulating various types of fraudulent behaviors, these models can better anticipate and respond to new fraud tactics (Sources: [Towards Data Science](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

**What is data generation vs data augmentation?**

Data Generation:

Data generation involves creating entirely new data points that mimic the properties of real data. This is typically done using models like GANs, VAEs, and Large Language Models (LLMs). The generated data can be used to supplement existing datasets or create entirely new datasets, which is useful when the available data is insufficient or when data privacy needs to be maintained. For example, GANs can generate realistic images that can be used in training computer vision models, while VAEs can generate diverse synthetic data for various applications (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

Data Augmentation:

Data augmentation involves creating new data points by modifying existing data. This process enhances the diversity of the training dataset by applying transformations such as rotation, flipping, cropping, and adding noise to images, or generating new samples by oversampling minority classes in imbalanced datasets. Data augmentation is widely used to improve the robustness and performance of machine learning models. For instance, in image classification tasks, augmented data helps models generalize better by exposing them to a wider variety of examples during training (Sources: [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a), [Exploring GANs to Generate Synthetic Data](https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518)).

**What models/techniques have been applied historically in synthetic data?**

Several models and techniques have historically been used to generate synthetic data, each with its strengths and applications:

SMOTE and ADASYN:

SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) are techniques used to address class imbalance in datasets. These methods generate synthetic samples for the minority class to create a more balanced dataset, which helps in improving the performance of classification models. They are particularly useful in scenarios like fraud detection and medical diagnosis where minority class instances are critical but rare (Sources: [GANs for Synthetic Data Generation](https://towardsai.net/p/l/gans-for-synthetic-data-generation), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

Variational Autoencoders (VAEs):

VAEs are generative models that create new data instances by learning the distribution of the input data. They encode the data into a latent space and then decode it to generate new, synthetic data points. VAEs are useful in generating diverse data samples and have applications in image generation, anomaly detection, and more (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)).

Generative Adversarial Networks (GANs):

GANs consist of a generator and a discriminator network that compete to produce realistic synthetic data. The generator creates synthetic data, while the discriminator evaluates its authenticity. This adversarial process helps the generator improve the quality of the synthetic data over time. GANs are widely used for generating realistic images, videos, and even text (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

**What are the potential advantages/limitations of GANs and other similar techniques? (comparison table)**

|  |  |  |
| --- | --- | --- |
| Technique | Advantages | Limitations |
| GANs | High-quality data generation, captures complex distributions | Mode collapse, difficult to train, requires large datasets (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Exploring GANs to Generate Synthetic Data](https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518)) |
| VAEs | Captures overall data distribution well, easier to train | Generates blurrier, less realistic data compared to GANs (Sources: [GANs for Synthetic Data Generation](https://towardsai.net/p/l/gans-for-synthetic-data-generation), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)) |
| SMOTE/ADASYN | Simple to implement, effective for class balancing | Limited to tabular data, can lead to overfitting if not used properly (Sources: [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)) |

**How are GANs constructed? What is the Generator (noise vector)? What is the Discriminator?**

GANs Construction:

* Generator: The generator network uses a random noise vector as input and generates synthetic data that mimics real data. The noise vector injects randomness, providing diversity in the generated output. The generator learns to transform this noise into data that resembles the real data distribution. For example, in image generation, the generator might start with random noise and produce an image that looks realistic to a human observer (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).
* Discriminator: The discriminator network acts as a classifier to differentiate between real and synthetic data. It assigns a probability score to the data it evaluates, indicating whether it believes the data is real or generated by the generator. The discriminator provides feedback to the generator, helping it improve its ability to produce realistic data over time. The adversarial nature of this relationship drives both networks to improve continuously (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)).

**What is the difference between GANs and TimeGANs?**

GANs:

GANs are primarily used for generating static data like images and text. They learn the distribution of the training data and generate new samples that mimic this distribution. The main application areas of GANs include image synthesis, video generation, and style transfer. GANs are effective for tasks where the temporal component is not a factor (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

TimeGANs:

TimeGANs are specifically designed for generating time-series data. They incorporate recurrent components to capture temporal dependencies within the data, making them suitable for sequential data generation like financial transactions, stock prices, and medical records. TimeGANs ensure that the generated sequences maintain the temporal coherence and patterns observed in the original data. This makes them particularly useful in applications requiring the preservation of temporal dynamics (Sources: [Protect and Extend - Using GANs for Synthetic Data Generation of Time-Series Medical Records](https://arxiv.org/pdf/2402.14042), [Medium - Synthetic Data Generation Using GANs Part 2](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce)).

**How does the whole adversarial training process look like?**

Adversarial training involves the following steps:

Generator Training:

The generator creates synthetic data to fool the discriminator. It uses a noise vector as input and generates data that mimics the real data distribution. The generator's objective is to produce data that is indistinguishable from real data by the discriminator. Over iterations, the generator improves its ability to create realistic data by learning from the feedback provided by the discriminator (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

Discriminator Training:

The discriminator learns to differentiate between real and synthetic data. It evaluates the data generated by the generator and assigns a probability score indicating whether the data is real or fake. The discriminator provides feedback to the generator, which helps the generator improve its data generation process. The discriminator's goal is to accurately identify real versus fake data, making it a crucial part of the adversarial training loop (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)).

Adversarial Loop:

Both networks are trained simultaneously in a loop. The generator improves its ability to generate realistic data, while the discriminator gets better at identifying fake data. This process continues until the discriminator can no longer reliably distinguish between real and synthetic data, indicating that the generator has learned to produce highly realistic data. This dynamic and competitive training process drives the improvement of both models over time (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

**What are the metrics/success criteria when generating synthetic data?**

Metrics and success criteria include:

Quality of Generated Data (QoG):

The quality of synthetic data is measured by how closely it resembles real data. Statistical tests like autocorrelation analysis can be used to compare the distribution of synthetic and real data. High-quality synthetic data should have similar statistical properties to the original data, ensuring that models trained on synthetic data perform comparably to those trained on real data (Sources: [Protect and Extend - Using GANs for Synthetic Data Generation of Time-Series Medical Records](https://arxiv.org/pdf/2402.14042), [Medium - Synthetic Data Generation Using GANs Part 2](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce)).

Privacy Preservation:

It is crucial that synthetic data does not leak any real data. This can be assessed using membership inference attacks, where the goal is to determine if a specific data point was part of the training dataset. High-quality synthetic data should pose minimal risk of revealing information about the original dataset, thus maintaining privacy (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Detecting the Unseen: Anomaly Detection with GANs](https://medium.com/data-reply-it-datatech/detecting-the-unseen-anomaly-detection-with-gans-8b20f3056a11)).

Model Performance:

Evaluating the performance of machine learning models trained on synthetic data compared to those trained on real data is essential. Metrics such as F1 score, RMSE, and accuracy are used to assess the effectiveness of synthetic data in various applications. High-quality synthetic data should enable models to perform similarly to models trained on real data, ensuring that the synthetic data is a viable substitute for real data in training and testing scenarios (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

**What is the loss function(s) used? (this is a nice point of research for time-series)**

In GANs, typical loss functions include:

Binary Cross Entropy (BCE):

BCE is used to evaluate the realism of generated data by calculating the loss for both real and fake data. This loss function helps train the discriminator and the generator. The discriminator aims to maximize the likelihood of correctly classifying real and synthetic data, while the generator aims to minimize the discriminator's ability to distinguish between real and fake data (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

Wasserstein Loss:

Used in Wasserstein GANs (WGANs), the Wasserstein loss function measures the distance between the real and generated data distributions. This loss function helps to solve the vanishing gradient problem commonly seen in traditional GANs and provides a more stable training process. WGANs are known for generating higher quality data and being more resilient to training issues (Sources: [GANs for Synthetic Data Generation](https://towardsai.net/p/l/gans-for-synthetic-data-generation), [Exploring GANs to Generate Synthetic Data](https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518)).

Custom Loss Functions for Time-Series:

For time-series data, loss functions might include temporal loss components to ensure generated sequences maintain temporal coherence. This involves using recurrent neural network architectures and designing loss functions that capture temporal dependencies, which is critical for applications like financial forecasting and medical record synthesis (Sources: [Protect and Extend - Using GANs for Synthetic Data Generation of Time-Series Medical Records](https://arxiv.org/pdf/2402.14042), [Medium - Synthetic Data Generation Using GANs Part 2](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce)).

**What tools/libraries/packages etc are there available to use? Would you use PyTorch or TensorFlow? Which out of the two provides the best implementations and why?**

Tools/Libraries:

1. PyTorch: PyTorch is known for its dynamic computation graphs, which are easier for debugging and more intuitive for developing GAN models. It has extensive support for GAN implementations and a vibrant community that contributes to various advanced models and techniques. PyTorch is often favored in research environments due to its flexibility and ease of use (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).
2. TensorFlow: TensorFlow offers robust support for large-scale deployments and comprehensive documentation. It supports GAN implementations and is widely used in industry settings. TensorFlow is known for its scalability and production-readiness, making it suitable for applications that require stable and scalable models in a production environment (Sources: [Medium - Synthetic Data Generation Using GANs Part 1](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

Best Implementations:

* PyTorch is often preferred for research and development due to its flexibility and ease of use.
* TensorFlow is favored in production environments for its scalability and robust deployment capabilities (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [How to Generate Synthetic Tabular Data](https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a)).

**Can we prove GAN-based approaches outperform other techniques like VAE?**

Studies and comparisons have shown that GAN-based approaches often outperform other techniques like VAEs in generating high-quality synthetic data, especially in terms of realism and diversity. GANs have been shown to produce more realistic images and better capture complex distributions compared to VAEs, which may generate blurrier and less detailed outputs. For instance, GANs are able to generate high-resolution images that are indistinguishable from real images, whereas VAEs might struggle with fine details and texture (Sources: [Generative AI: Synthetic Data Generation with GANs using PyTorch](https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd), [GANs for Synthetic Data Generation](https://towardsai.net/p/l/gans-for-synthetic-data-generation), [Fantastic GANs](https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them)).

**What are diffusion models?**

Diffusion models are a type of generative model that simulate the way substances like gas or particles spread. They iteratively add noise to the data and then learn to reverse this process, creating high-quality samples from noise. Diffusion models have shown promise in generating high-fidelity images and other complex data types. This approach leverages the natural diffusion process, where data is gradually corrupted by noise, and then the model learns to reverse this corruption step by step. This method is particularly effective in generating data with intricate details and structures, making it useful in applications like image synthesis and text generation (Sources: [Protect and Extend - Using GANs for Synthetic Data Generation of Time-Series Medical Records](https://arxiv.org/pdf/2402.14042), [Medium - Synthetic Data Generation Using GANs Part 2](https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce)).