UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

MÉTODOS PARA LA GENERACIÓN DE DATOS ARTIFICIALES TABULARES

MARC NEBOT MOYANO

Director/a: JAVIER BÉJAR ALONSO (Departamento de Ciencias de la Computación)

Titulación: Grado en Ingeniería Informática (Computación)

Memoria del trabajo de fin de grado

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

25/01/2024

Resumen

Este proyecto implica la creación de una librería en Pytorch que incluye varios métodos para generar datos tabulares artificiales, como GANs, VAEs, Bit Diffusion y un modelo llamado Data Augmentation basado en autoencoders de eliminación de ruido. El objetivo de la librería es proporcionar herramientas accesibles para que los usuarios experimenten y evalúen enfoques generativos.

La evaluación de los modelos muestra que su rendimiento varía según el contexto y las características del conjunto de datos. Aunque los resultados son buenos, aún hay margen para incorporar metodologías adicionales y mejorar la calidad de los datos sintéticos.

En resumen, la librería ha logrado generar datos artificiales de buena calidad, y con más dedicación, se puede mejorar aún más. Esto establece una base sólida para futuras investigaciones en generación de datos sintéticos.

Abstract

This project involves the creation of a Pytorch library that includes various methods for generating artificial tabular data, such as GANs, VAEs, Bit Diffusion, and a model called Data Augmentation based on denoising autoencoders. The library's purpose is to provide accessible tools for users to experiment with and evaluate generative approaches.

The evaluation of the models reveals that their performance varies depending on the context and dataset characteristics. Although the results are good, there is still room to incorporate additional methodologies and improve the quality of synthetic data.

In summary, the library has successfully generated high-quality artificial data, and with further dedication, further improvements can be made. This establishes a solid foundation for future research in synthetic data generation.

Resum

Aquest projecte implica la creació d'una llibreria en Pytorch que inclou diversos mètodes per generar dades tabulars artificials, com ara GANs, VAEs, Bit Diffusion i un model anomenat Data Augmentation basat en autoencoders de supressió de soroll. La finalitat de la llibreria és proporcionar eines accessibles perquè els usuaris experimentin i avaluïn enfocaments generatius.

L'avaluació dels models revela que el seu rendiment varia segons el context i les característiques del conjunt de dades. Tot i que els resultats són bons, encara hi ha espai per incorporar metodologies addicionals i millorar la qualitat de les dades sintètiques.

En resum, la llibreria ha aconseguit generar dades artificials de bona qualitat, i amb més dedicació, es poden fer millores addicionals. Això estableix una base sòlida per a futures investigacions en la generació de dades sintètiques.

Índice

1. Introducción..... 10
1.1. Contextualización..... 10
1.2. Problema a resolver..... 10
1.3. Actores implicados..... 12
2. Justificación..... 13
2.1. El mercado actual..... 13
2.1.1. OpenAI GPT-3 y GPT-4..... 13
2.1.2. Scikit-learn..... 13
2.1.3 Tensorflow y Pytorch..... 13
2.1.4 Faker..... 13
2.1.5. Data Wig..... 14
2.1.6. Catboost..... 14
2.2. Prueba y error..... 14
3. Alcance..... 16
3.1. Objetivos..... 16
3.2. Requerimientos..... 17
3.3. Obstáculos y riesgos..... 18
4. Metodología..... 19
5. Planificación..... 20
5.1. Tareas..... 20
5.1.1. Gestión del Proyecto..... 20
5.1.2. Desarrollo del Proyecto..... 21
5.1.3. Imprevistos..... 23
5.2. Sprints..... 24
5.3. Recursos necesarios..... 26
5.3.1. Recursos humanos..... 26
5.3.2. Recursos de software..... 26
5.3.3. Recursos de hardware..... 27
5.3.4. Otros recursos..... 27
5.4. Gestión de riesgos..... 28

5.5. Cambios respecto a la planificación inicial..... 29
6. Presupuesto..... 30
6.1. Costes de personal..... 30
6.2. Costes genéricos..... 33
6.2.1. Costes de hardware..... 33
6.2.2. Costes de software..... 33
6.2.3. Costes eléctricos..... 34
6.2.4. Costes del espacio de trabajo..... 34
6.2.5. Costes de internet..... 35
6.2.6. Costes genéricos totales..... 35
6.3. Contingencias..... 35
6.4. Imprevistos..... 36
6.5. Coste total del proyecto..... 37
6.6. Control de gestión..... 38
7. Sostenibilidad..... 39
7.1. Autoevaluación..... 39
7.2. Dimensión económica..... 39
7.3. Dimensión social..... 40
7.4. Dimensión ambiental..... 40
8. Aspectos Legales..... 41
8.1. Leyes y normativas..... 41
8.1.1. Reglamento General de Protección de Datos (GDPR) - UE..... 41
8.1.2. Ley Orgánica de Protección de Datos Personales..... 41
8.1.3. Consideraciones éticas y ENIA en España..... 42
8.2. Licencias..... 42
9. Implementación del proyecto..... 43
9.1. Estructura del proyecto..... 44
9.1.1. Módulo de carga de los datos y preprocesamiento..... 45
9.1.2. Módulo de arquitectura de modelos..... 45
9.1.3. Módulo de entrenamiento de modelos..... 45
9.1.4. Módulo de evaluación y visualización de los datos generados..... 45
9.1.5. Complementos adicionales del proyecto..... 45

9.2. Preproceso y lectura de datos....46
9.3. Redes generativas adversativas (GAN).... 47
9.3.1. Arquitectura GAN.... 47
9.3.2. Entrenamiento GAN....49
9.3.3. Redes generativas adversativas condicionales (CGAN).... 49
9.4. Variational Autoencoder (VAE).... 50
9.4.1. Arquitectura VAE....50
9.4.2. Entrenamiento VAE.... 53
9.5. Bit Diffusion....54
9.5.1. Arquitectura Bit Diffusion.... 54
9.5.2. Entrenamiento Bit Diffusion.... 57
9.6. Data Augmentation.... 58
9.6.1. Arquitectura Data Augmentation....58
9.6.2. Entrenamiento Data Augmentation.... 59
10. Evaluación de los modelos....60
10.1. Métricas....60
10.1.1. Prueba de Kolmogorov-Smirnov.... 60
10.1.2. Chi-square....61
10.2. Visualización....62
10.2.1. Principal Components.... 62
10.2.2. Distributed Stochastic Neighbor Embedded.... 62
10.2.3. Gráfico de densidad....63
11. Tests y resultados....64
11.1. Conjuntos de datos pequeños....64
11.1.1. Test de GAN....65
11.1.2. Test de VAE....73
11.1.3. Test de Bit Diffusion....80
11.1.4. Test de Data Augmentation....86
11.2. Conjuntos de datos grandes....92
11.2.1. Test de GAN.... 93
11.2.2. Test de VAE.... 96
11.2.3. Test de Bit Diffusion.... 98

11.2.4. Test de Data Augmentation..... 100
11.3. Conclusiones de los resultados de los modelos.....103
12. Relación entre el proyecto y la carrera..... 104
13. Futuros proyectos..... 105
14. Conclusiones.....106
Referencias.....107

Índice de figuras

Figura 1: Curva ajustada a un conjunto de datos artificiales.....11
Figura 2: El proceso del modelo de la experimentación..... 15
Figura 3: Ciclos y sprints de la metodología ágil.....19
Figura 4: Diagrama de Gantt.....25
Figura 5: Estructura modular del proyecto y posible uso mediante script..... 44
Figura 6: Arquitectura de GAN..... 47
Figura 7: Arquitectura de VAE.....50
Figura 8: Arquitectura de Bit Diffusion..... 54
Figura 9: Arquitectura de Data Diffusion..... 58
Figura 10: Captura de las pérdidas por época de generador y discriminador.....65
Figuras 11 y 12: Gráficos de densidad de las características reales y generadas..... 68
Figuras 13 y 14: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados..... 68
Figura 15: Captura del log que muestra las pérdidas de las últimas épocas de generador y discriminador..... 69
Figuras 16 y 17: Gráficos de densidad de las características reales y generadas.....71
Figuras 18 y 19: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados.....72
Figura 20: Pérdidas del modelo VAE (10 épocas).....73
Figuras 21 y 22: Gráficos de densidad de las características 1 y 3 de los datos reales y generados..... 74
Figuras 23 y 24: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados.....75
Figura 25: Pérdidas del modelo VAE (100 épocas).....76

Figura 26: Gráficos de densidad de la característica 2 de los datos reales y generados..... 78
Figuras 27: Unión de los gráficos de densidad de todas las características..... 78
Figuras 28 y 29: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados..... 79
Figura 30: Pérdidas del modelo Bit Diffusion (10 épocas)..... 80
Figuras 31 y 32: Gráficos de densidad de las características 1 y 3 de los datos reales y generados..... 81
Figuras 33 y 34: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados..... 82
Figura 35: Pérdidas del modelo Bit Diffusion (100 épocas)..... 83
Figuras 36 y 37: Gráficos de densidad de las características 1 y 3 de los datos reales y generados..... 84
Figuras 38 y 39: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados..... 85
Figura 40: Pérdidas del modelo Data Augmentation (10 épocas)..... 86
Figuras 41 y 42: Gráficos de densidad de las características 2 y 4 de los datos reales y generados..... 87
Figuras 43 y 44: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados..... 88
Figura 45: Pérdidas del modelo Data Augmentation (100 épocas)..... 89
Figuras 46 y 47: Gráficos de densidad de las características 2 y 4 de los datos reales y generados..... 90
Figuras 48 y 49: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados..... 91
Figuras 50 y 51: Gráficos de densidad de las características 1 y 3 de los datos reales y generados..... 94
Figura 52: Gráfico que muestra el t-SNE de GAN de los datos reales comparándolos con los generados..... 95
Figuras 53 y 54: Gráficos de densidad de las características 2 y 4 de los datos reales y generados..... 97
Figura 55: Gráfico que muestra el t-SNE de VAE de los datos reales comparándolos con los generados..... 97
Figuras 56 y 57: Gráficos de densidad de las características 1 y 4 de los datos reales y generados..... 99

Figura 58: Gráfico que muestra el t-SNE de Bit Diffusion de los datos reales comparándolos con los generados.....100

Figuras 59 y 60: Gráficos de densidad de las características 2 y 4 de los datos reales y generados.....101

Figura 61: Gráfico que muestra el t-SNE de Data Augmentation de los datos reales comparándolos con los generados.....102

Índice de tablas

Tabla 1: Tareas del proyecto..... 24

Tabla 2: Sueldo medio de roles seleccionados.....31

Tabla 3: Coste total de personal..... 32

Tabla 4: Coste total eléctrico..... 34

Tabla 5: Coste total genérico..... 35

Tabla 6: Costes imprevistos.....36

Tabla 7: Coste total del proyecto..... 37

Tabla 8: Resultados Kolmogorov-Smirnov para GAN (10 épocas)..... 66

Tabla 9: Resultados Kolmogorov-Smirnov para GAN (100 épocas).....70

Tabla 10: Resultados Kolmogorov-Smirnov para VAE (10 épocas)..... 74

Tabla 11: Resultados Kolmogorov-Smirnov para VAE (100 épocas).....77

Tabla 12: Resultados Kolmogorov-Smirnov para Bit Diffusion (10 épocas)..... 80

Tabla 13: Resultados Kolmogorov-Smirnov para Bit Diffusion (100 épocas)..... 83

Tabla 14: Resultados Kolmogorov-Smirnov para Data Augmentation (10 épocas)..... 86

Tabla 15: Resultados Kolmogorov-Smirnov para Data Augmentation (100 épocas)..... 89

Tabla 16: Resultados Kolmogorov-Smirnov para GAN (conjunto grande).....93

Tabla 17: Resultados Kolmogorov-Smirnov para VAE (conjunto grande)..... 96

Tabla 18: Resultados Kolmogorov-Smirnov para Bit Diffusion (conjunto grande)..... 98

Tabla 19: Resultados Kolmogorov-Smirnov para Data Augmentation (conjunto grande)... 100

Tabla 20: Resumen de los mejores resultados Kolmogorov-Smirnov..... 103

1. Introducción

El proyecto *Métodos para la generación de datos artificiales tabulares* corresponde al trabajo de fin de grado en Ingeniería Informática (TFG) para la Facultad de Informática de Barcelona (FIB) en la especialidad de computación, este proyecto se realizará con prudencia y precisión para obtener los mejores resultados posibles.

1.1. Contextualización

En la era de la información y la tecnología, los datos se han convertido en un recurso de sumo valor. La toma de decisiones, la inteligencia empresarial y el desarrollo de sistemas de aprendizaje automático dependen en gran medida de la disponibilidad de conjuntos de datos adecuados y relevantes. Sin embargo, la obtención de datos de calidad a menudo es un desafío, y en muchas ocasiones, los datos reales pueden ser escasos, costosos de obtener o incluso erróneos.

Es aquí donde la generación de datos artificiales tabulares se ha convertido en un campo de estudio esencial en la ciencia de datos y la inteligencia artificial. Este trabajo de fin de grado se enfoca en explorar y analizar una serie de métodos y técnicas diseñados para crear datos tabulares de manera artificial. Estos datos generados cumplen un papel fundamental en la investigación, el desarrollo de modelos y la evaluación de algoritmos, proporcionando una alternativa valiosa cuando los datos reales son limitados o inaccesibles.

1.2. Problema a resolver

En el ámbito de la ciencia de datos y la inteligencia artificial, uno de los problemas más habituales es la disponibilidad y calidad de los conjuntos de datos, por lo tanto, es aquí donde se necesitan ideas y soluciones innovadoras.

El problema central que este trabajo se propone abordar es cómo superar esta barrera fundamental en el proceso de investigación y desarrollo en ciencia de datos e inteligencia artificial. La generación de datos artificiales tabulares se presenta como una solución potencial a este problema. Sin embargo, es esencial comprender las técnicas disponibles y determinar cuáles son más adecuadas en diferentes contextos y aplicaciones.

Por otro lado, se debe considerar la necesidad de garantizar que los datos generados sean fiables, como se observa en la Figura 1, se trata de una curva ajustada a los datos artificiales que han sido generados por un método estadístico, esto es, lo que se debe conseguir con este trabajo, la semejanza y correlación entre datos reales y datos sintéticos.

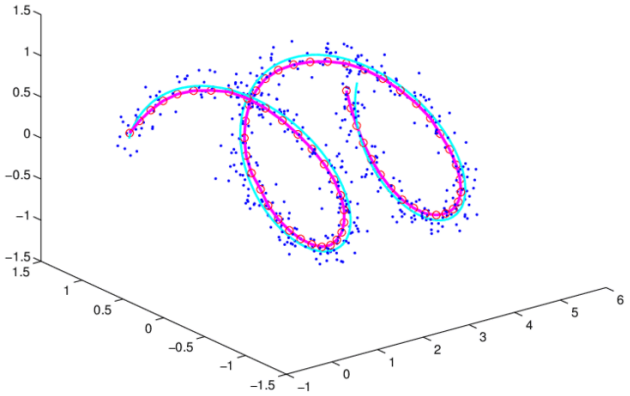


Figura 1: Curva ajustada a un conjunto de datos artificiales
Fuente: www.researchgate.net

Este trabajo se enfocará en la exploración de las técnicas y enfoques para la generación de datos artificiales tabulares, con el objetivo de proporcionar una respuesta efectiva al desafío de la disponibilidad de datos en la ciencia de datos y la inteligencia artificial. A lo largo de esta investigación, se buscará identificar las ventajas, desventajas y aplicaciones prácticas de estos métodos, con el propósito de brindar una comprensión sólida de cómo abordar y resolver el problema fundamental de la escasez de datos de calidad en estos campos.

1.3. Actores implicados

Uno de los principales objetivos de la generación de datos artificiales es la provisión de información a individuos o instituciones que no constan de esta y, por lo tanto, es necesaria la intervención artificial, como se ha mencionado. Estos actores, cada uno con sus propios intereses y responsabilidades, contribuyen de manera significativa al desarrollo y la utilización de datos sintéticos, en consecuencia, es importante identificar y presentar los posibles interesados en la generación de datos artificiales:

- **Científicos de Datos y Analistas**: Los científicos de datos y analistas son los principales usuarios de datos sintéticos en sus investigaciones y proyectos. Dependiendo de la calidad y representatividad de los datos generados, pueden lograr avances significativos.

- **Instituciones de Investigación y Académicas:** Las instituciones de investigación y académicas desempeñan un papel crucial en la promoción de la generación de datos artificiales a través de la investigación y la educación.

- **Empresas y Organizaciones:** Las empresas y organizaciones utilizan datos sintéticos para el desarrollo y la evaluación de algoritmos de aprendizaje automático, la creación de prototipos de productos y la capacitación de personal sin exponer datos sensibles o confidenciales que permitan fugas de información.

- **Gobiernos**: Los gobiernos o reguladores tienen un interés en garantizar que la generación de datos sintéticos cumpla con las normativas de privacidad y protección de datos. Pueden establecer directrices y regulaciones para el uso responsable de datos artificiales.

- **Sociedad**: La sociedad en su conjunto se beneficia de la generación de datos sintéticos, ya que promueve la investigación científica, la innovación tecnológica y la toma de decisiones informadas en diversos sectores, como por ejemplo en el ámbito de la salud. Estos datos artificiales permiten a la sociedad no exponerse a pruebas o estadísticas que vulneren su privacidad e integridad.

Se ha podido observar, la importancia de la identificación de los actores que desarrollan un papel importante en el ámbito de los datos artificiales, así como las ventajas que pueden proporcionar a cada colectivo mencionado.

2. Justificación

El mercado de generación de datos sintéticos ha experimentado un crecimiento notable en los últimos años, impulsado por la creciente demanda de datos de alta calidad para aplicaciones en ciencia de datos, inteligencia artificial y aprendizaje automático. Este auge ha dado lugar a la proliferación de diversas herramientas y enfoques para la creación de datos artificiales.

2.1. El mercado actual

Algunas de las herramientas más destacadas y ampliamente utilizadas en la actualidad incluyen:

2.1.1. OpenAI GPT-3 y GPT-4

Los modelos de lenguaje de *OpenAI*, como *GPT-3* y su sucesor *GPT-4*, pueden generar texto artificial de alta calidad. Estos modelos son capaces de generar datos tabulares y textuales que son útiles para tareas de análisis y modelado de datos.

2.1.2. Scikit-learn

Esta popular biblioteca de *Python* se utiliza ampliamente en la ciencia de datos y el aprendizaje automático. Aunque su enfoque principal es el aprendizaje supervisado y no la generación de datos, incluye algunas funciones que permiten la creación de datos artificiales para trabajos de clasificación y regresión.

2.1.3 Tensorflow y Pytorch

Estas bibliotecas de *deep learning* proporcionan herramientas para crear modelos generativos, como las Redes Generativas Adversarias (GANs) y los Autoencoders Variacionales (VAE), que son ampliamente utilizados para generar datos sintéticos en diversos dominios.

2.1.4 Faker

Faker es una biblioteca de *Python* que se utiliza para generar datos sintéticos y ficticios, como nombres de personas, direcciones, números de teléfono y más. Es útil para la generación de datos de prueba o simulación.

2.1.5. Data Wig

Data Wig es una biblioteca de código abierto que utiliza modelos de aprendizaje automático para imputar y generar valores de datos faltantes en conjuntos de datos tabulares.

2.1.6. Catboost

Esta biblioteca de aprendizaje automático incluye una función para generar características sintéticas que pueden mejorar el rendimiento de los modelos de clasificación.

Se observa que, en la actualidad, existen varios recursos para generar datos sintéticos en caso de ser necesario, pero este proyecto se centrará en la creación de una librería en *Pytorch*.

2.2. Prueba y error

En la búsqueda de métodos efectivos para la generación de datos sintéticos tabulares, la exploración y experimentación desempeñan un papel crucial. El proceso de desarrollo y perfeccionamiento de técnicas de generación de datos a menudo se basa en un enfoque de experimentación, donde se evalúan y comparan diversos métodos para determinar su eficacia en diferentes contextos y conjuntos de datos.

En el marco de este Trabajo de Fin de Grado, se propone la creación de una librería de *PyTorch* dedicada a la generación de datos artificiales tabulares. Esta librería servirá como plataforma para probar una variedad de métodos tanto innovadores como establecidos. La idea es explorar y evaluar métodos basados en modelos de aprendizaje automático, técnicas estadísticas tradicionales y enfoques híbridos que combinen múltiples estrategias de generación.

El proceso de prueba y error implica la implementación y experimentación con estos métodos en diferentes conjuntos de datos representativos. Se llevarán a cabo análisis exhaustivos para evaluar la calidad y la utilidad de los datos artificiales generados, como se puede observar en el proceso de la Figura 2 [1]. Se medirán métricas de calidad, como la similitud con datos reales, la capacidad para preservar patrones subyacentes y la aplicabilidad en tareas de análisis y modelado de datos.

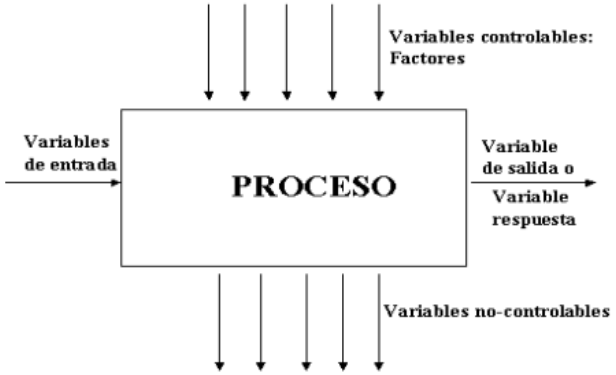


Figura 2: El proceso del modelo de la experimentación

Fuente: Captura de pantalla de la Universidad de Granada (www.ugr.es) [1]

Uno de los objetivos fundamentales de este proceso de prueba y error es proporcionar una comprensión más profunda de cómo funcionan estos métodos en diferentes escenarios y qué métodos son más efectivos para abordar desafíos específicos de generación de datos. Esto permitirá obtener una visión más completa de las ventajas y desventajas de cada enfoque, así como recomendaciones sobre cuándo y cómo utilizarlos de manera más efectiva.

El análisis resultante de la librería de generación de datos sintéticos y los métodos evaluados será un componente valioso de este trabajo, ya que proporcionará una guía práctica y entorno de experimentación para profesionales de la ciencia de datos y el aprendizaje automático en la elección de métodos de generación de datos que se adapten mejor a sus necesidades y escenarios específicos.

3. Alcance

Una vez contextualizado este trabajo, se procederá a detallar los objetivos, junto con sus correspondientes requisitos y riesgos. Mediante una evaluación minuciosa, se obtendrá una comprensión más amplia con el propósito de prevenir posibles errores a lo largo de la ejecución de esta investigación.

3.1. Objetivos

El objetivo principal de este trabajo es crear un entorno de experimentación de distintos métodos para la generación de datos artificiales tabulares. Para lograr este objetivo principal, se desglosan los siguientes objetivos específicos:

- **Investigar y analizar** las técnicas y enfoques más relevantes utilizados en la generación de datos tabulares artificiales, incluyendo modelos basados en aprendizaje automático, técnicas estadísticas y otras estrategias innovadoras.

- **Diseñar y desarrollar una librería de *PyTorch*** dedicada a la generación de datos artificiales tabulares. Esta librería servirá como plataforma para probar y comparar diversos métodos de generación de datos artificiales tabulares, tanto tradicionales como innovadores.

- **Implementar una serie de experimentos** en conjuntos de datos representativos con el propósito de evaluar y comparar la calidad de los datos artificiales generados por los diferentes métodos.

- **Analizar los resultados de los experimentos** con las métricas pertinentes y proporcionar recomendacione**s** basadas en evidencia sobre cuándo y cómo utilizar métodos específicos de generación de datos artificiales en diversas situaciones.

- **Evaluar y abordar consideraciones éticas** y de privacidad relacionadas con la generación y uso de datos artificiales, en todo momento este trabajo pretende garantizar la privacidad de los datos e integridad de los usuarios.

Este trabajo también pretende contribuir al campo de la generación de datos artificiales tabulares mediante la presentación de resultados, conclusiones y perspectivas para futuras investigaciones.

3.2. Requerimientos

Para llevar a cabo la investigación y el desarrollo de la librería de generación de datos sintéticos tabulares de manera efectiva, es esencial establecer una serie de requerimientos que guíen el proceso y aseguren la calidad del trabajo. Los requerimientos clave incluyen:

1. En primer lugar, se necesitan conjuntos de datos representativos, los cuales serán extraídos de plataformas que proveen *data sets* de libre acceso para experimentar. Estos deberán ser de distintos tamaños para evaluar la eficacia de los métodos de generación.

2. En segundo lugar, se necesitará la disponibilidad de recursos computacionales adecuados, incluyendo hardware con capacidad de procesamiento y memoria suficientes para llevar a cabo experimentos y entrenar modelos de generación de datos masivos, este proyecto necesitará la capacidad del ***Barcelona Supercomputing Center - Centro Nacional de Supercomputación (BSC-CNS)*** para la generación masiva de datos artificiales tabulares.

3. En tercer lugar, la utilización de entornos de desarrollo y software de programación eficiente, como ***PyTorch*** y bibliotecas adicionales para el análisis de datos y visualización.

4. Finalmente, pero no menos importante la consideración y respeto de las regulaciones de privacidad de datos, asegurando que los datos utilizados y generados cumplan con los estándares éticos y legales.

3.3. Obstáculos y riesgos

En el desarrollo de la librería de generación de datos artificiales tabulares, se deben considerar los siguientes obstáculos y riesgos.

Uno de los desafíos potenciales radica en la calidad de los datos de entrada. La generación de datos artificiales de alta calidad depende en gran medida de la calidad de los datos de referencia utilizados como punto de partida. Si los datos de entrada son deficientes, sesgados o incompletos, esto puede influir negativamente en la calidad de los datos sintéticos generados.

Además, la complejidad de algunos métodos de generación de datos artificiales tabulares puede ser un obstáculo. Implementar y ajustar algoritmos sofisticados, especialmente aquellos basados en *deep learning*, puede requerir un conocimiento técnico avanzado y recursos computacionales significativos.

También existe la posibilidad de que los resultados obtenidos no sean óptimos. A pesar de la experimentación y el análisis, es posible que ninguno de los métodos de generación de datos se adapte perfectamente, lo que podría resultar en datos artificiales tabulares de baja calidad e insatisfactorios.

Asimismo, la capacidad limitada de recursos computacionales, como el hardware del ordenador, puede ser un obstáculo si no se cuenta con la capacidad de procesamiento y memoria suficientes para llevar a cabo experimentos y entrenar modelos generativos.

Finalmente, pero no menos importante, uno de los riesgos podría ser la falta de tiempo para desarrollar este proyecto, ya que está siendo desarrollado a contrarreloj y podría no alcanzar el óptimo por falta de tiempo.

4. Metodología

Para llevar a cabo este proyecto se seguirá una metodología ágil. Esta metodología se caracteriza por su enfoque flexible y adaptativo, por lo tanto, esta es la metodología ideal para este tipo de proyecto.

La metodología ágil se dividirá en sprints, con una duración definida. Cada sprint se centrará en objetivos y tareas específicas y se llevará a cabo siguiendo los principios ágiles, lo que permite la adaptación continua a medida que se avanza en el proyecto:

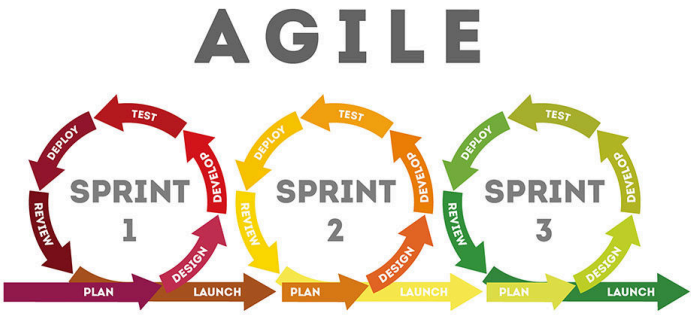


Figura 3: Ciclos y sprints de la metodología ágil.
Fuente: www.ebf.com.es

Para la planificación y gestión del proyecto, se utilizarán herramientas de gestión de proyectos, como *Jira*. Este proporciona un marco sólido para la planificación de tareas, la asignación de recursos, el seguimiento del progreso y la gestión de problemas.

La comunicación eficiente con el tutor será fundamental en el proyecto, y se establecerán reuniones eventuales de seguimiento para revisar el progreso, discutir los desafíos y tomar decisiones.

El control de versiones mediante *Git* garantizará un seguimiento preciso de la evolución del proyecto y permitirá la colaboración en el desarrollo de código.

La documentación será un componente crítico del proyecto, y se mantendrá una documentación completa y actualizada que incluirá los informes de experimentos y cualquier otro material necesario para comprender y utilizar la librería de generación de datos artificiales.

5. Planificación

Al planificar el proyecto, resulta fundamental evaluar su duración y establecer un cronograma realista. Considerando que este trabajo corresponde a 18 créditos académicos, y que cada crédito se estima que requiere una dedicación de alrededor de 30 horas, por lo tanto, el tiempo total necesario para el proyecto sería de 540 horas. Sin embargo, es crucial determinar cuántas horas se deben trabajar diariamente para alcanzar el número total de horas requeridas.

El proyecto tuvo inicio el 2 de septiembre de 2023 y está programado para finalizar el 14 de enero de 2024. Esto nos da un período total de 135 días. Dentro de este plazo, se incluyen días laborables y fines de semana, excluyendo festivos. Si se distribuyen equitativamente las 540 horas necesarias durante este período, se debería asignar un horario de trabajo que consista en aproximadamente 4 horas al día, todos los días, incluyendo fines de semana.

Este enfoque permitiría una distribución uniforme de la carga de trabajo a lo largo del proyecto y aseguraría que se cumplan los objetivos en el período programado. Es importante tener en cuenta que los ajustes en el horario de trabajo pueden ser necesarios en función de las necesidades específicas del proyecto y las circunstancias cambiantes.

5.1. Tareas

A continuación, se detallarán las tareas esenciales de este proyecto, incluyendo la identificación de los recursos requeridos para cada una de ellas y la estimación del tiempo necesario para su ejecución.

5.1.1. Gestión del Proyecto

Esta sección corresponde al curso de GEP y representa un valor de 3 créditos académicos, lo que equivale alrededor de 90 horas de trabajo en total. Las tareas que se detallarán en esta área están vinculadas a la documentación del proyecto, la cual es esencial para su desarrollo adecuado.

- **Contexto y Alcance [T1.1]**: Esta tarea ocupa los 4 primeros puntos del proyecto donde se justifican las decisiones tomadas y se contextualiza el proyecto. Para elaborar esta sección se han empleado tres tipos de recursos y se estima que tomará aproximadamente 25 horas.

- **Planificación Temporal [T1.2]**: Para llevar a cabo esta tarea, es necesario haber completado previamente la primera tarea. En este caso, se trata de la planificación temporal del proyecto, con el objetivo de cumplir los plazos establecidos. Se estima que requerirá unas 20 horas de trabajo.

- **Presupuesto y Sostenibilidad [T1.3]**: Se requiere realizar un análisis de costos de todo el proyecto antes de su ejecución y evaluar su viabilidad, además de considerar el impacto socioambiental. Se estima que esta tarea tomará alrededor de 30 horas y su ejecución dependerá de la finalización de la Tarea 1.2.

- **Comunicaciones [T1.4]**: De manera similar a la memoria, las reuniones se programarán y llevarán a cabo durante la ejecución del proyecto. Estas se irán realizando dependiendo de la disponibilidad del alumno y del profesor.

5.1.2. Desarrollo del Proyecto

Para llevar a cabo este proyecto, se implementará la metodología *Agile*, como se ha mencionado previamente. En este contexto, se llevarán a cabo un total de 10 sprints, durante los cuales se abordarán las siguientes tareas:

1. **Investigación y análisis de métodos [T2.1]:** En esta tarea, se invertirán aproximadamente 30 horas en la revisión de la literatura relevante. Esto incluirá la exploración de investigaciones previas, la lectura de artículos académicos y la consulta de recursos relacionados. A medida que se avance en esta revisión, se seleccionarán los métodos y enfoques que considero más adecuados para este proyecto de generación de datos artificiales tabulares.

2. **Diseño [T2.2]:** El diseño de la librería de generación de datos artificiales tabulares es una tarea central en este proyecto. Se estima que llevará aproximadamente 30 horas. Esto implica definir la estructura de la librería, las interfaces de usuario (si corresponde), la arquitectura técnica y la interacción con otras herramientas y bibliotecas relacionadas.

3. **Implementación [T2.3]:** La fase de implementación es donde el proyecto cobra vida y los métodos seleccionados se transforman en una librería de *PyTorch* dedicada a la generación de datos artificiales tabulares:

a. **Desarrollo de la librería [T2.3.1]:** En esta tarea será implementada la librería donde se escribirá el código pertinente para que todos los módulos queden interconectados y será la base que conectará los métodos seleccionados. Se estima que llevará unas 60 horas de trabajo.

b. **Implementación de Métodos Seleccionados [T2.3.2]:** Dentro del desarrollo de la librería, se dedicarán alrededor de 120 horas a la implementación de los métodos de generación de datos que previamente se investigó y se seleccionaron. Cada método requerirá su propio enfoque y ajustes específicos.

4. **Experimentación [T2.4]:** Un aspecto crítico de la implementación es la realización de pruebas exhaustivas para garantizar que la librería funcione correctamente y genere datos tabulares de alta calidad:

   a. **Selección de Conjuntos de Datos [T2.4.1]:** En primer lugar, serán dedicadas alrededor de 10 horas a la cuidadosa selección de conjuntos de datos que sean representativos y adecuados para evaluar los métodos de generación de datos. Esto incluirá la búsqueda, evaluación y preparación de los conjuntos de datos.

   b. **Configuración de Experimentos [T2.4.2]:** En segundo lugar y se trabajará 15 horas en la configuración de los experimentos. Esto implica definir los parámetros experimentales, establecer métricas de evaluación y planificar la ejecución de los experimentos.

   c. **Ejecución de Experimentos [T2.4.3]:** La ejecución de los experimentos es una fase intensiva que requerirá alrededor de 40 horas de trabajo. Durante este período, será utilizada la librería de *PyTorch* para generar datos artificiales tabulares y se registrarán los resultados obtenidos.

   d. **Análisis de Resultados [T2.4.4]:** Una vez finalizados los experimentos, se dedicará aproximadamente 20 horas al análisis de los resultados. Se evaluará la calidad de los datos generados por cada método y se comparará su rendimiento en términos de métricas relevantes.

e. **Generación de Recomendaciones [T2.4.5]:** Dependiendo de los resultados de los experimentos, se emplearán alrededor de 5 horas para generar recomendaciones y conclusiones fundamentadas sobre cuándo y cómo utilizar métodos específicos de generación de datos artificiales en diferentes situaciones.

5. **Documentación [T2.5]**: La documentación es esencial para asegurar que otros puedan comprender y utilizar la librería. Esta tarea abarca desde el inicio del proyecto hasta su entrega final, y en ella se redactarán los objetivos logrados y sus respectivos resultados. Aunque no depende directamente de otras tareas previas, sí se llevará a cabo de manera paralela con el avance del proyecto. Se dedicarán alrededor de 100 horas a la creación de una documentación completa que describa cómo utilizar la librería, sus métodos y todo lo que deba contener la memoria cuando finalice el proyecto.

5.1.3. Imprevistos

La fase de imprevistos está destinada a abordar posibles contratiempos y desafíos inesperados que puedan surgir durante el desarrollo del proyecto. Se destinarán 20 horas para identificar, analizar y responder eficazmente a cualquier problema imprevisto que pueda afectar el curso del proyecto.

Etiqueta	Nombre	Horas	Dependencias	Recursos
T1	**Gestión del Proyecto**	**90h**	-	-
T1.1	Contexto y Alcance	25h	-	R1,R2,R3
T1.2	Planificación Temporal	20h	T1.1	R1,R2,R3
T1.3	Presupuesto y Sostenibilidad	30h	T1.2	R1,R2,R3
T1.4	Comunicaciones	15h	-	R1,R2
T2	**Desarrollo del Proyecto**	**430h**	**-**	-
T2.1	Investigación y análisis de métodos	30h	T1	R1,R2,R3
T2.2	Diseño	30h	T2.1	R1,R2,R3
T2.3	Implementación	180h	T2.2	R1,R2,R3
T2.3.1	Desarrollo de la librería	60h	T2.2	R1,R2,R3
T2.3.2	Implementación de Métodos Seleccionados	120h	T2.3.1	R1,R2,R3
T2.4	Experimentación	90h	T2.3	R1,R2,R3
T2.4.1	Selección de Conjuntos de Datos	10h	-	R1,R2,R3
T2.4.2	Configuración de Experimentos	15h	T2.3, T2.4.1	R1,R2,R3
T2.4.3	Ejecución de Experimentos	40h	T2.3, T2.4.2	R1,R2,R3
T2.4.4	Análisis de Resultados	20h	T2.4.3	R1,R2,R3
T2.4.5	Generación de Recomendaciones	5h	T2.4.4	R1,R2,R3
T2.5	Documentación	100h	T2	R1,R2,R3
Imprevistos		**20h**		
Total		**540h**		

Tabla 1: Tareas del proyecto
Fuente: Elaboración propia

5.2. Sprints

El proyecto se ha estructurado en un total de 10 sprints, y cada período en el diagrama de *Gantt* abarca de 9 a 11 días. A continuación, se presenta el diagrama de *Gantt*, que muestra las fechas de inicio y finalización de todas las tareas mencionadas anteriormente. Es importante destacar que la tarea de "Comunicaciones" no tiene un tiempo específico asignado en el diagrama, ya que se llevarán a cabo cuando sea necesario y estén disponibles tanto el tutor como el alumno. Además, se puede observar cierta superposición entre las tareas, puesto que no existen dependencias entre ellas. Esta superposición tiene el propósito de diversificar las tareas y fomentar la productividad en el desarrollo del proyecto.

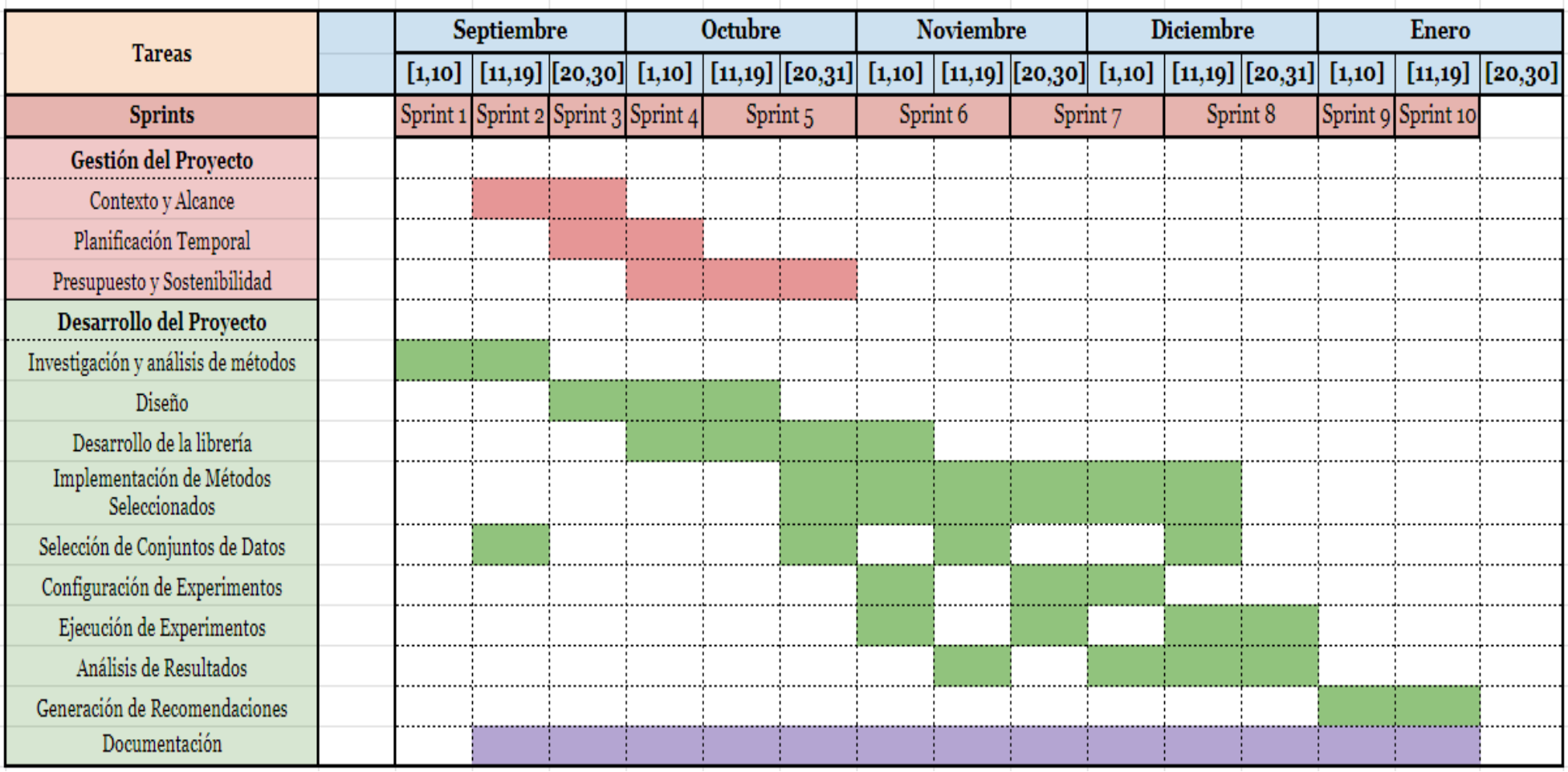


Figura 4: Diagrama de Gantt

Fuente: Elaboración propia

5.3. Recursos necesarios

Para llevar a cabo con éxito el proyecto de desarrollo de la librería de generación de datos sintéticos tabulares, se identifican los siguientes recursos esenciales que serán utilizados por el alumno (responsable del proyecto) y el director del proyecto (profesor):

5.3.1. Recursos humanos

A continuación, serán explicados los recursos humanos que se utilizarán en este proyecto, referenciados en la Tabla 1 como *R1*:

- **Alumno (Responsable del Proyecto):** El alumno desempeñará un papel central en la investigación, desarrollo y documentación del proyecto. Será responsable de la implementación de métodos, experimentos, seguimiento de tareas y comunicación.

- **Director del Proyecto (Profesor):** El director del proyecto, en su rol de supervisor y guía, proporcionará orientación técnica, supervisará el progreso, ofrecerá asesoramiento y evaluación, y participará en reuniones de seguimiento.

5.3.2. Recursos de software

A continuación, serán explicados los recursos de software que se utilizarán en este proyecto, referenciados en la Tabla 1 como *R2*:

- ***Git*:** Se utilizará *Git* como sistema de control de versiones para rastrear y gestionar los cambios en el código fuente y la documentación. Facilitará la colaboración en el desarrollo.

- ***Jira*:** *Jira* será la herramienta principal de gestión de proyectos para planificar tareas, asignar recursos, realizar seguimiento del progreso y gestionar problemas y riesgos.

- ***VisualStudio*:** Se empleará *VisualStudio* u otro entorno de desarrollo integrado (IDE) adecuado para la programación y desarrollo de la librería.

- ***Google Meet y Google Calendar*:** Estas herramientas se utilizarán para programar y llevar a cabo reuniones de seguimiento, revisión de avances y comunicación entre el alumno y el director del proyecto.

- **Software de Ciencia de Datos:** Se emplearán herramientas específicas de ciencia de datos como *Python, NumPy*, pandas, scikit-learn y *PyTorch* para implementar y evaluar los métodos de generación de datos.

5.3.3. Recursos de hardware

A continuación, serán explicados los recursos de software que se utilizarán en este proyecto, referenciados en la Tabla 1 como *R3*:

- **Equipo del Alumno**: El hardware del alumno consiste en un sistema con 16 GB de *RAM*, un procesador *AMD Ryzen* 5 5600G y una gráfica *NVIDIA GeForce* RTX 3060. Estos recursos proporcionan una capacidad de procesamiento adecuada para llevar a cabo tareas de entrenamiento de modelos de aprendizaje automático y experimentos computacionalmente intensivos.

5.3.4. Otros recursos

- **Conjuntos de Datos**: Se requerirá acceso a conjuntos de datos tabulares diversificados y representativos que abarquen diferentes dominios y tamaños, según las necesidades de la investigación.

- **Hardware Adicional:** En caso de que se requiera hardware adicional, como unidades de almacenamiento o recursos de cómputo en la nube, se considerará su adquisición o provisión según la demanda del proyecto.

La identificación y gestión adecuada de estos recursos son esenciales para garantizar el progreso fluido y el éxito del proyecto, permitiendo que el alumno y el director del proyecto trabajen de manera eficiente y efectiva en la creación de la librería de generación de datos sintéticos tabulares.

5.4. Gestión de riesgos

En el desarrollo de este proyecto, es esencial una adecuada gestión de riesgos para abordar posibles obstáculos y garantizar el logro de los objetivos. Los riesgos mencionados incluyen:

- **Calidad de los Datos de Entrada**: Existe el riesgo de que los datos de referencia utilizados para la generación de datos artificiales tabulares no sean de alta calidad, lo que podría afectar negativamente la calidad de los datos artificiales. Para mitigar este riesgo, se realizará una cuidadosa selección y preparación de los datos de entrada.

- **Complejidad de los Métodos**: La complejidad de algunos métodos de generación de datos artificiales puede ser un desafío, especialmente en términos de conocimiento técnico y recursos computacionales. Si este riesgo se materializa, se buscarán soluciones alternativas o simplificaciones de métodos.

- **Resultados Subóptimo**s: Existe la posibilidad de que los resultados obtenidos no alcancen el nivel deseado. Para abordar este riesgo, se realizará un análisis exhaustivo y, si es necesario, se explorarán nuevos métodos o enfoques.

- **Recursos Computacionales Limitados:** La capacidad limitada de recursos computacionales, como hardware insuficiente, puede ralentizar el proceso de desarrollo y experimentación. En caso de que surjan problemas, se considerará la optimización de recursos y la distribución adecuada del tiempo.

- **Restricción de Tiempo**: Dado que el proyecto se desarrolla en un período de tiempo ajustado, existe el riesgo de no lograr un nivel óptimo debido a limitaciones de tiempo. Para abordar este riesgo, se evaluará continuamente la distribución de horas en el proyecto y se considerarán ajustes si es necesario.

La gestión de riesgos será un proceso dinámico a lo largo del proyecto. Si el número de horas previsto en el plan no se adapta a la realidad, se realizarán ajustes y redistribuciones de tiempo según sea necesario para garantizar el progreso y la consecución de los objetivos del proyecto.

5.5. Cambios respecto a la planificación inicial

En cuanto a los cambios realizados en la planificación inicial, es importante destacar que no se han producido cambios significativos en la estructura general del proyecto. No obstante, se ha incorporado una modificación importante en el tiempo dedicado al aprendizaje de las Redes Generativas Adversarias (GANs).

Esta adición, con un total de 30 horas adicionales, se ha introducido para abordar la necesidad de adquirir conocimientos específicos sobre GANs, los cuales no estaban contemplados en las habilidades previas al inicio del TFG. Este ajuste se consideró esencial para fortalecer la base técnica y garantizar un enfoque más completo en las tecnologías relevantes para los objetivos del proyecto.

Por otro lado, este cambio solo afectará específicamente la fase de planificación en lo que respecta a la implementación de los métodos relacionados con las GANs que es en la fase en la que se encuentra el proyecto en la actualidad. Es esencial señalar que esta ampliación en el tiempo no conllevará un aumento en el presupuesto del proyecto, ya que se considera una medida de contingencia dentro del marco presupuestario inicial y un riesgo que ya se había contemplado. De esta manera, se busca garantizar la efectividad y la calidad de la implementación, adaptándose a las necesidades emergentes del desarrollo del TFG.

6. Presupuesto

A continuación, se describirá el presupuesto estimado del proyecto, teniendo en cuenta el personal involucrado, los costes generales, y los recursos tecnológicos necesarios. Este análisis financiero proporcionará una visión clara de los recursos económicos requeridos para llevar a cabo con éxito todas las etapas del proyecto.

6.1. Costes de personal

En el proyecto se requerirá un conjunto de roles clave que desempeñarán funciones esenciales desde el inicio hasta la finalización del proyecto de generación de datos artificiales tabulares. Estos roles incluyen:

- **Director de Proyecto**: El director de proyecto será responsable de supervisar y coordinar todas las actividades, garantizando que el proyecto avance de acuerdo con la planificación establecida.

- **Desarrollador de Software**: El desarrollador de software será el encargado de escribir el código y desarrollar la librería de PyTorch para la generación de datos artificiales tabulares.

- **Científico de Datos**: El científico de datos desempeñará un papel clave en la fase de experimentación, donde llevará a cabo la ejecución de experimentos, analizará los resultados y generará recomendaciones basadas en evidencias.

- **Documentador Técnico**: El documentador técnico se encargará de crear una documentación completa y clara sobre el uso de la librería y sus métodos.

- **Analista de Riesgos**: El analista de riesgos jugará un papel importante en la gestión de riesgos, identificando posibles obstáculos y ayudando a diseñar estrategias para mitigarlos.

- **Otros roles (Opcionales):** Dependiendo de las necesidades específicas del proyecto, pueden requerirse roles adicionales, como expertos en ética de datos o especialistas en seguridad de la información, para abordar aspectos éticos y de privacidad, pero en este proyecto no se tendrán en cuenta y se darán por hecho que no son necesarios, puesto que otro rol es capaz de abordar este tema.

A continuación, se muestra la Tabla 2 con la media de sueldos correspondientes a los roles mencionados. Los datos se han obtenido de Talent [2].

Etiqueta	Posición	Precio/hora (neto)	Precio/hora (bruto SS)
P1	Director de Proyecto	14,79€/h	19,23€/h
P2	Desarrollador de Software	11,33€/h	14,74€/h
P3	Científico de Datos	13,41€/h	17,44€/h
P4	Documentador Técnico	10,16€/h	13,21€/h
P5	Analista de Riesgos	10,58€/h	13,76€/h

Tabla 2: Sueldo medio de roles seleccionados
Fuente: Elaboración propia

Estos roles trabajarán en conjunto para llevar a cabo todas las fases del proyecto, desde la investigación y el desarrollo hasta la experimentación y la gestión de riesgos, asegurando así un enfoque completo y exitoso en la generación de datos artificiales tabulares.

Gracias a esta información, en la Tabla 3 se estima el coste de las tareas asignadas de manera más precisa.

Etiqueta	Nombre	Horas	Posición	Coste
T1	**Gestión del Proyecto**	**90h**	-	1.730,7€
T1.1	Contexto y Alcance	25h	P1	480,75€
T1.2	Planificación Temporal	20h	P1,P5	384,6€
T1.3	Presupuesto y Sostenibilidad	30h	P1,P5	576,9€
T1.4	Comunicaciones	15h	P1	288,45€
T2	**Desarrollo del Proyecto**	**430h**	**-**	6.509,2€
T2.1	Investigación y análisis de métodos	30h	P3	523,2€
T2.2	Diseño	30h	P2	442,2€
T2.3	Implementación	180h	-	2653,2€
T2.3.1	Desarrollo de la librería	60h	P2	884,4€
T2.3.2	Implementación de Métodos Seleccionados	120h	P2	1.768,8€
T2.4	Experimentación	90h	-	1.569,6€
T2.4.1	Selección de Conjuntos de Datos	10h	P3	174,4€
T2.4.2	Configuración de Experimentos	15h	P3	261,6€
T2.4.3	Ejecución de Experimentos	40h	P3	697,6€
T2.4.4	Análisis de Resultados	20h	P3	348,8€
T2.4.5	Generación de Recomendaciones	5h	P3	87,2€
T2.5	Documentación	100h	P4	1.321€
Total		**540h**		**8.239,9€**

Tabla 3: Coste total de personal

Fuente: Elaboración propia

6.2. Costes genéricos

A continuación, se considerarán todos los gastos y costes adicionales que no estén relacionados con el personal involucrado en el proyecto. Estos costes abarcarán una variedad de aspectos, como recursos tecnológicos, software, materiales, servicios externos, gastos generales de funcionamiento y cualquier otro desembolso necesario para llevar a cabo el proyecto de manera eficiente y efectiva.

6.2.1. Costes de hardware

Los costes de hardware en este proyecto se refieren a los gastos asociados a los recursos tecnológicos físicos necesarios para su ejecución. Esto incluye el valor de los componentes de hardware utilizados, como el ordenador personal del estudiante (con un coste de 1300 €) y la pantalla utilizada (evaluada en aproximadamente 200 €).

Por otra parte, hay que tener en cuenta que, en promedio, un componente de hardware puede mantenerse en perfectas condiciones durante aproximadamente 4 años, lo que equivale a 48 meses. Dado que planeamos utilizar este hardware específico para el proyecto durante aproximadamente 5 meses, realizando 4 horas cada día, es necesario realizar una estimación del costo total proyectado para este período. Una vez han sido conocidos los valores se puede hacer una estimación del coste utilizando la siguiente fórmula:

$$\frac{5\ meses \times 30\ días \times 4\ horas}{48\ meses \times 30\ dias \times 24\ horas} \times (1.300€ + 200€) = \mathbf{26{,}04€}$$

Estos costes se integran en la estimación global de los recursos económicos del proyecto, siendo elementos esenciales para llevar a cabo las diversas tareas relacionadas con el desarrollo, investigación y experimentación.

6.2.2. Costes de software

En relación con los costes de software, es relevante destacar que en este proyecto no se contempla ningún gasto relacionado con la adquisición de software o licencias. Esto se debe a que se planea utilizar exclusivamente software de código abierto y herramientas gratuitas para llevar a cabo todas las actividades necesarias en el proyecto. Esta elección de utilizar software gratuito es fundamental para mantener el proyecto dentro de los límites presupuestarios y optimizar la eficiencia en términos de recursos económicos.

6.2.3. Costes eléctricos

En este apartado se toma en consideración el consumo eléctrico asociado a los elementos utilizados en el proyecto. En este contexto, se utilizará un ordenador con un consumo de aproximadamente 30 W, una pantalla con un consumo de alrededor de 23 W y la iluminación de la habitación, que tiene un gasto de 17 W. Estos dispositivos estarán en funcionamiento durante el período de ejecución del proyecto.

Para calcular los costes eléctricos, se debe tener en cuenta la potencia total de estos dispositivos y la cantidad de tiempo que permanecerán encendidos. Además, se considerará la tarifa eléctrica, que es de 0,25 € por kilovatio-hora (kWh).

Los costes eléctricos se determinarán multiplicando la potencia total (en kilovatios, kW) por el tiempo en funcionamiento (en horas) y luego multiplicando este valor por la tarifa eléctrica por kWh. Esto proporcionará una estimación de los costes eléctricos totales asociados al proyecto. El control y la gestión eficiente del consumo eléctrico son importantes para optimizar los recursos económicos del proyecto, se puede ver este cálculo realizado en la Tabla 4.

Dispositivo	Consumo	Horas	Precio
Ordenador	30W	540h	4,05€
Pantalla	23W	540h	3,01€
Iluminación	17w	540h	2,3€
Total			**9,36€**

Tabla 4: Coste total eléctrico
Fuente: Elaboración propia

6.2.4. Costes del espacio de trabajo

Otro gasto que se debe considerar es el espacio de trabajo que se utilizará durante los próximos cinco meses, durante el desarrollo del proyecto. En este caso, se trata de una habitación de aproximadamente diez metros cuadrados, con un costo de 15,55 € por metro cuadrado al mes en promedio. Haciendo los cálculos, esto resulta en un costo mensual de alrededor de 155,50 €, lo que suma un total de aproximadamente 777,50 € para los cinco meses de uso.

6.2.5. Costes de internet

El coste de internet se refiere al gasto relacionado con la conexión a internet utilizada durante la ejecución del proyecto. En España, el costo promedio mensual de una línea de teléfono fija más internet es de alrededor de 53 € al mes.

Si planeamos utilizar esta conexión a internet durante los cinco meses de duración del proyecto, el costo total estimado sería de aproximadamente 53 € al mes multiplicado por 5 meses, lo que resulta en un total de unos 265 €.

6.2.6. Costes genéricos totales

A continuación se adjunta la tabla de costes genéricos totales:

Tipo de coste	Precio
Hardware	26,04€
Software	0€
Eléctrico	9,36€
Espacio de trabajo	777,50€
Internet	265€
Total	**1.077,9€**

Tabla 5: Coste total genérico
Fuente: Elaboración propia

6.3. Contingencias

Por otro lado, es importante tener en cuenta que durante el proyecto pueden surgir imprevistos que supongan un costo adicional. Siguiendo la práctica común en España, se debería destinar un 23,6% del presupuesto total para contingencias. Esto se traduce en una cantidad de 2.232,34 € reservados específicamente para afrontar cualquier imprevisto que pueda surgir a lo largo del proyecto, garantizando así una gestión financiera sólida y la capacidad de hacer frente a situaciones imprevistas.

6.4. Imprevistos

La gestión de posibles obstáculos es fundamental para prever y mitigar riesgos durante el desarrollo del proyecto. Con base en los posibles contratiempos previamente identificados, se ha calculado el posible impacto económico de estos imprevistos:

- **Complejidad de algunos métodos**: En caso de enfrentar problemas derivados de la complejidad de los métodos, se considera una adición de 20 horas de trabajo al desarrollador de software, con un costo total de 294,8 euros. Se estima un riesgo del 10 % debido a la naturaleza técnica y sofisticada de los algoritmos.

- **Resultados no óptimos**: Si los resultados obtenidos no cumplen con los estándares deseados, se prevé una adición de 15 horas de trabajo al científico de datos, con un costo total de 261,6 euros. Se estima un riesgo del 8 % dada la posibilidad de que los métodos de generación de datos no se ajusten perfectamente a las expectativas.

- **Capacidad limitada de recursos**: En el caso de encontrarse con limitaciones en los recursos informáticos, se ha considerado un costo total de 150 euros. Se estima un riesgo del 12 % debido a la necesidad de contar con una capacidad de procesamiento y memoria adecuada para el desarrollo de los experimentos.

- **Falta de tiempo**: Si se experimenta una presión de tiempo significativa, se considera una adición de 30 horas de trabajo, 25 para el desarrollador y 5 horas para el científico de software, con un costo total de 455,7 €. Se estima un riesgo del 15 % debido a la naturaleza acelerada del proyecto.

Imprevisto	Coste	Riesgo	Coste total
Complejidad	294,8€	10%	29,48€
Resultados no óptimos	261,6€	8%	20,93€
Recursos limitados	150€	12%	22,5€
Falta de tiempo	455,7€	15%	68,36€
Total			**141,27€**

Tabla 6: Costes imprevistos
Fuente: Elaboración propia

6.5. Coste total del proyecto

A continuación en la Tabla 7 se detalla el coste total del proyecto:

Tipo de coste	Precio
Personal	8.239,9€
Genérico	1.077,9€
Contingencias	2.232,34€
Imprevistos	141,27€
Total	**11.691,41€**

Tabla 7: Coste total del proyecto
Fuente: Elaboración propia

En caso de que no se presenten sucesos que requieran el uso del presupuesto de contingencias o imprevistos, los fondos restantes podrían destinarse a mejorar la infraestructura tecnológica del proyecto. Por ejemplo, podríamos invertir en la adquisición de hardware más avanzado para fortalecer nuestras capacidades de investigación y desarrollo. Este hardware mejorado no solo beneficiaría el proyecto actual, sino que también sentaría las bases para futuras investigaciones y proyectos, permitiéndonos explorar de manera más efectiva nuevas técnicas y enfoques en el campo de la generación de datos tabulares artificiales.

Además, podríamos considerar destinar parte del presupuesto sobrante a proporcionar capacitación adicional al equipo. Al invertir en la mejora de las habilidades y el conocimiento del equipo en el ámbito de la generación de datos artificiales, podríamos fortalecer nuestra capacidad para abordar futuros desafíos y proyectos con un mayor grado de competencia. Este enfoque no solo beneficiaría directamente el proyecto actual, sino que también impulsaría el potencial de futuras iniciativas en el campo de la investigación de datos y la inteligencia artificial.

6.6. Control de gestión

El punto de control de gestión económica del proyecto implica una evaluación periódica del costo y el consumo al final de cada sprint. Para calcular estas desviaciones, se seguirá el siguiente procedimiento:

- **Desviación del Coste**: Esta métrica se obtiene restando el coste estimado del proyecto al coste real y multiplicándolo por las horas reales invertidas. Si el resultado de esta fórmula es positivo, indica que el proyecto está gastando más de lo previsto, lo que podría requerir utilizar el presupuesto de contingencias para cubrir este exceso. En cambio, si la desviación es negativa, significa que el proyecto se encuentra dentro del presupuesto estimado.

- **Desviación del Consumo**: Para calcular esta desviación, se resta la cantidad de horas estimadas del proyecto a las horas reales invertidas y se multiplica por el coste estimado. Cuando el resultado es positivo, indica que el proyecto está consumiendo menos recursos de los previstos, lo que podría sugerir una eficiencia en la gestión del presupuesto. Por otro lado, si la desviación es negativa, significa que el proyecto está utilizando más recursos de los planificados.

Este seguimiento constante asegura una gestión financiera efectiva y permite tomar medidas adecuadas en función de las desviaciones encontradas.

7. Sostenibilidad

7.1. Autoevaluación

En la autoevaluación de la sostenibilidad del proyecto, he observado que existen ciertos aspectos en los que poseo un mayor grado de conocimiento y conciencia, como es el caso de las dimensiones ambientales y económicas. No obstante, he identificado que no estoy acostumbrado a analizar el impacto social que puede causar un proyecto. Durante el proceso, he descubierto que existen métricas destinadas a medir el impacto social de un proyecto, un conocimiento que antes no poseía.

Además, he notado que, si bien en el ámbito universitario se han abordado estos temas de manera gradual y progresiva, especialmente en relación con las dimensiones ambientales y económicas, todavía existe un margen para profundizar en el enfoque social, ya que el impacto de los proyectos a nivel social puede ser a veces desconocido pero a su vez lo suficiente importante para tenerlo en cuenta.

7.2. Dimensión económica

En lo que respecta a la dimensión económica de la sostenibilidad, es importante destacar que este proyecto ha abordado este aspecto de manera adecuada. A lo largo de la carrera, he adquirido un sólido conocimiento sobre la gestión económica de proyectos y la importancia de mantener presupuestos sostenibles en distintas asignaturas.

Durante el desarrollo de este proyecto, se han aplicado las prácticas aprendidas para garantizar un uso eficiente de los recursos económicos disponibles. Se ha establecido un presupuesto cuidadosamente planificado y se ha realizado un seguimiento constante de los gastos para mantenerse dentro de los límites establecidos. Esta gestión económica rigurosa ha permitido utilizar eficazmente los recursos financieros.

Por otro lado, la creación de una librería de *PyTorch* dedicada a la generación de datos artificiales tabulares puede proporcionar a las empresas y organizaciones una herramienta valiosa para generar datos sintéticos de alta calidad de manera eficiente. Esto podría llevar a un ahorro significativo de recursos, ya que las empresas no tendrían que invertir tiempo y dinero en la recopilación y limpieza de grandes conjuntos de datos reales. En este sentido, el proyecto podría contribuir positivamente a la eficiencia económica de las empresas que utilicen esta librería.

7.3. Dimensión social

En lo que respecta a la dimensión social de la sostenibilidad, este proyecto también puede tener un impacto significativo, ya sea de manera positiva o negativa.

Por un lado, el proyecto podría contribuir positivamente a la sociedad al proporcionar una herramienta útil para la generación de datos artificiales de alta calidad. Esto podría ser beneficioso para investigadores, empresas y organizaciones que trabajan en una amplia variedad de campos, incluyendo la investigación científica, el desarrollo de productos, la toma de decisiones basada en datos y más. Al facilitar el acceso a datos sintéticos, el proyecto podría fomentar la innovación y el progreso en diversas áreas.

Sin embargo, también es importante considerar posibles impactos negativos. Si los datos sintéticos generados no son representativos o no se ajustan adecuadamente a ciertos contextos, podrían tomarse decisiones erróneas o sesgadas basadas en ellos, lo que podría tener un impacto negativo en la toma de decisiones y, en última instancia, en la sociedad.

7.4. Dimensión ambiental

En el ámbito de la sostenibilidad ambiental, este proyecto podría generar un impacto positivo al reducir la necesidad de recopilar y almacenar grandes cantidades de datos reales. Esto podría contribuir a la disminución de la carga ambiental asociada con la infraestructura de almacenamiento de datos y la recopilación de datos en el mundo digital.

No obstante, es fundamental garantizar que el proceso de generación de datos sea eficiente en términos de recursos computacionales y energía para minimizar cualquier impacto ambiental negativo. En última instancia, el objetivo es contribuir a la sostenibilidad ambiental al reducir la dependencia de datos reales, siempre teniendo en cuenta la eficiencia en el uso de recursos.

8. Aspectos Legales

8.1. Leyes y normativas

8.1.1. Reglamento General de Protección de Datos (GDPR) - UE

En el marco de este proyecto, es crucial abordar las implicaciones del GDPR de la UE. Al trabajar con datos que podrían incluir información personal, se deben seguir estrictamente los principios fundamentales del GDPR [3].

Principalmente siempre se debe obtener consentimiento claro y específico del implicado, aplicar una recopilación de datos mínima y garantizar la seguridad. Además, se deben respetar los derechos individuales, realizar evaluaciones de impacto para la protección de datos y mantener registros detallados del uso de los datos recopilados.

Este enfoque ético y legal garantiza un tratamiento justo y seguro de los datos personales, promoviendo la transparencia y responsabilidad.

8.1.2. Ley Orgánica de Protección de Datos Personales

Por otro lado, también se debe considerar la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD) [4] en España. Este marco legal requiere un enfoque proactivo en la protección de la privacidad y derechos digitales.

Se debe garantizar la legitimidad en la obtención y tratamiento de datos personales, así como aplicar medidas de seguridad robustas en los procesos de trata y recopilación de datos personales. Además, se deben respetar los derechos de los individuos, incluido el derecho al olvido y la limitación del tratamiento de datos, dando el derecho a cada persona de modificar o eliminar sus propios datos.

El cumplimiento riguroso con la LOPDGDD refuerza la integridad y responsabilidad en el manejo de datos personales durante la generación artificial de datos.

8.1.3. Consideraciones éticas y ENIA en España

En el contexto del marco ético en inteligencia artificial española respaldado por la Estrategia Nacional de IA (ENIA)[5] en España, se establecen pautas fundamentales para la generación de datos artificiales. Estas directrices destacan la importancia de la transparencia, imparcialidad y responsabilidad en los algoritmos de generación.

La ENIA subraya la necesidad de evitar sesgos y discriminación, abogando por la equidad en la generación de datos. La ética en inteligencia artificial demanda una comprensión profunda de las implicaciones sociales y una toma de decisiones transparente en todas las etapas del proceso, tanto en la generación como recopilación de datos.

Considerando las leyes y regulaciones previas en España, es esencial tenerlas en cuenta para evitar cualquier infracción. Además, es crucial seguir prácticas éticas que garanticen la total transparencia en la ejecución del trabajo.

8.2. Licencias

Este TFG hace uso de librerías open source, como PyTorch, y otros elementos sin restricciones específicas de licencia. Por lo tanto, el proyecto se considera libre de limitaciones asociadas a licencias, permitiendo flexibilidad en su utilización y distribución.

9. Implementación del proyecto

Este apartado se centra en la descripción detallada del preproceso de los datos y la implementación de los modelos para la generación de datos artificiales. Se abordará desde la estructura inicial del proyecto hasta los resultados, detallando los métodos seleccionados para la generación de datos, los retos técnicos enfrentados y las soluciones adoptadas para resolverlos.

El objetivo principal de estas implementaciones ha sido el desarrollo de una biblioteca capaz de simular de forma precisa datos tabulares utilizando distintos métodos de generación. Para ello, se han explorado y aplicado técnicas de Redes Generativas Adversativas (GAN), Autoencoders Variacionales (VAE), Bit Diffusion y métodos de aumentación de datos, cada uno seleccionado por su potencial en la generación de datos realistas.

Durante el desarrollo, se han presentado varios desafíos técnicos y de diseño algorítmico, proporcionando una oportunidad para profundizar en el funcionamiento de estos modelos y aplicar soluciones innovadoras y eficaces.

En colaboración con el profesor encargado de tutorizar este trabajo de fin de grado, se realizó una búsqueda exhaustiva de referencias y recursos, destacando la librería ydata-synthetic [6] y SDV [7] como unas de las pocas existentes en el ámbito de generación de datos sintéticos. Estas librerías han servido como fuentes de inspiración y referencia para el enfoque y la implementación de los modelos del proyecto, ya que como se ha mencionado, actualmente existe muy poca información respecto a la generación de datos tabulares.

A continuación, se expondrán cada uno de los métodos implementados, explicando su fundamento teórico, su aplicación práctica en el marco del proyecto y cómo cada uno contribuye al objetivo de generar datos tabulares sintéticos de alta calidad.

9.1. Estructura del proyecto

La librería creada tiene el nombre de "GATDM (Generative Artificial Tabular Data Methods)" que se estructura en varios módulos clave. A continuación, en la Figura 5 se muestra la estructura del proyecto:

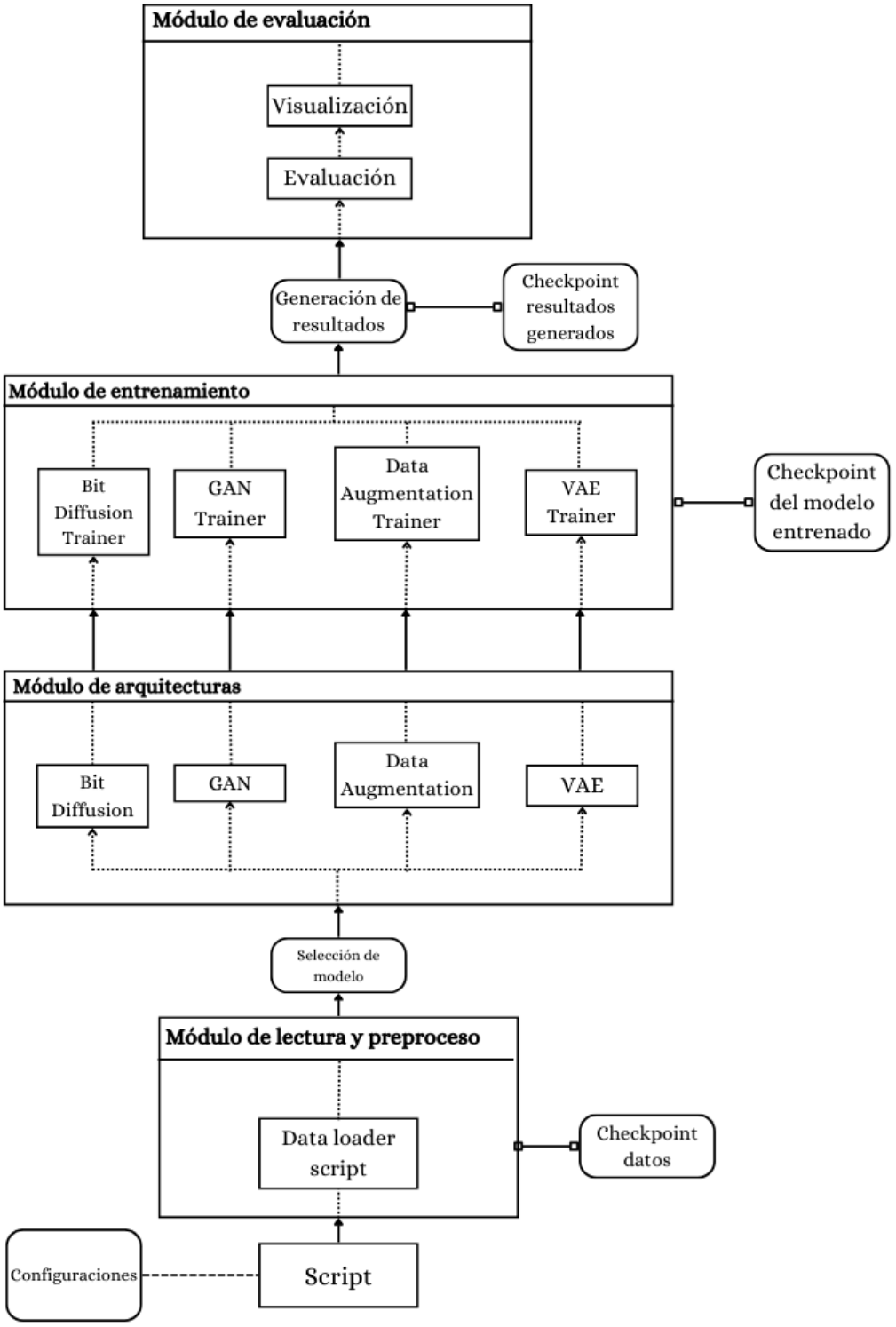


Figura 5: Estructura modular del proyecto y posible uso mediante script

Fuente: Elaboración propia usando Canva

9.1.1. Módulo de carga de los datos y preprocesamiento

Es el módulo principal y se encarga de la carga y el preprocesamiento de *datasets*. Este módulo garantiza que los datos estén preparados adecuadamente para ser utilizados por los modelos generativos.

9.1.2. Módulo de arquitectura de modelos

Este módulo es el núcleo de la librería y contiene las arquitecturas de los modelos de generación de datos. Incluye implementaciones de Bit Diffusion, Redes Generativas Adversativas (GAN) y un modelo que ha sido llamado *Data Augmentation* que utiliza técnicas de un *denoising autoencoder*. Cada modelo en este módulo ha sido cuidadosamente diseñado y optimizado para simular de manera efectiva datos tabulares.

9.1.3. Módulo de entrenamiento de modelos

En este módulo se encuentran las rutinas de entrenamiento para los modelos mencionados. Define la lógica de cómo se deben entrenar estos modelos, manejando aspectos como la optimización de parámetros y la gestión de los ciclos de entrenamiento, se ha separado de la arquitectura para proporcionar para proporcionar una mayor modularidad y facilitar la personalización y la experimentación con diferentes configuraciones de entrenamiento.

Esto permite que los usuarios ajusten y modifiquen las rutinas de entrenamiento según sus necesidades específicas, sin afectar la arquitectura subyacente de los modelos. La separación también ayuda en la mantenibilidad y escalabilidad del código, posibilitando la incorporación de nuevos modelos y métodos de entrenamiento de forma más eficiente y ordenada.

9.1.4. Módulo de evaluación y visualización de los datos generados

Este módulo auxiliar incluye herramientas esenciales como métricas para evaluar la calidad de los datos generados y funcionalidades para la construcción de visualizaciones. Estas herramientas son vitales para la validación y el análisis comparativo entre los datos sintéticos y los reales.

9.1.5. Complementos adicionales del proyecto

Además de estos módulos, la librería *GATDM* incluye varios componentes adicionales:

- **Carpeta *datasets***: Alberga las subcarpetas *generated_data* y *original_data*, destinadas a almacenar los datos generados y los *datasets* originales, respectivamente.

- **Script de demostración**: Un ejemplo práctico del uso de la librería, mostrando cómo interactuar con los módulos y utilizar sus funcionalidades.

- **Archivo config.py**: Permite personalizar parámetros clave del entrenamiento y configuraciones específicas para cada modelo, brindando flexibilidad y adaptabilidad.

- **Licencia, README, archivos de requerimientos y setup**: Estos componentes adicionales facilitan la comprensión, uso y distribución de la librería, asegurando que los usuarios tengan toda la información y herramientas necesarias para su implementación efectiva.

9.2. Preproceso y lectura de datos

A continuación se describe el funcionamiento del módulo que se encarga de cargar los datos y pre procesarlos. Este módulo lleva a cabo un preproceso básico que aborda dos aspectos fundamentales: la imputación de valores faltantes y la conversión de variables categóricas a formato numérico mediante *one hot encoding*.

La imputación de valores se realiza de manera simple, pero efectiva. Para las variables numéricas, se utiliza la media para reemplazar los valores faltantes, mientras que para las variables categóricas se emplea el valor más frecuente. Esta aproximación básica garantiza que el *dataset* esté completo para su uso en los modelos, incluso si accidentalmente se incluyen datos con valores faltantes.

Además, se lleva a cabo una normalización de los datos numéricos, ajustándolos en el rango entre 0 y 1. Esta normalización es esencial para el desempeño efectivo de los modelos generativos, permitiendo que operen en un rango de datos estandarizado.

Es importante destacar que, aunque el módulo ofrece un preproceso general que es adecuado para una gran variedad de situaciones, se debería realizar un análisis y preproceso detallado del conjunto de datos antes de utilizar la librería. Un preproceso específico y cuidadoso puede mejorar notablemente la calidad de los datos generados, ya que cada conjunto de datos puede tener características y necesidades únicas.

9.3. Redes generativas adversativas (GAN)

Las Redes Generativas Adversativas (GAN) han emergido como una herramienta poderosa en el ámbito de la generación de datos artificiales, imágenes y otro contenido artificial.

En este apartado se detalla la implementación de GAN en el proyecto, destacando tanto la estructura de los modelos como la lógica de entrenamiento. La capacidad de las redes GAN para aprender e imitar la distribución de los datos reales las convierte en una opción ideal para este propósito.

9.3.1. Arquitectura GAN

La arquitectura de GAN consta de dos componentes esenciales: el generador y el discriminador como se observa en la Figura 6, cada uno con características únicas diseñadas para simular eficazmente las distribuciones de datos reales.

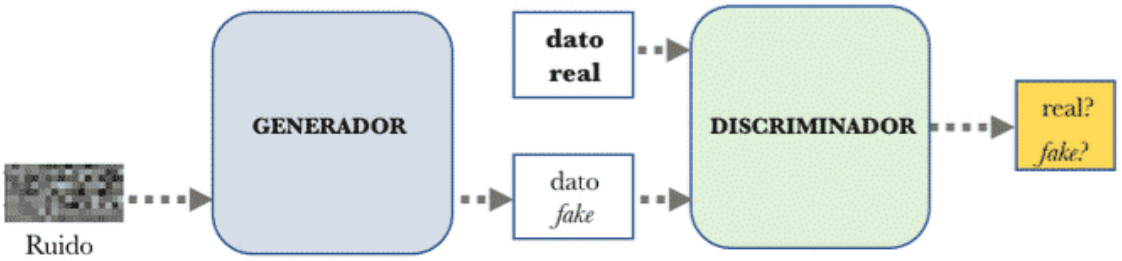


Figura 6: Arquitectura de GAN
Fuente: https://torres.ai/generative-adversarial-networks/

Generador

La estructura del Generador se compone de una serie de capas, alternando entre capas lineales y activaciones LeakyReLU.

La activación de *LeakyReLU* ayuda a evitar el problema de las neuronas inactivas y facilita el aprendizaje. Seguidamente, después de cada activación *LeakyReLU*, se utiliza la normalización por lotes, que contribuye a la estabilidad durante el entrenamiento.

Por otro lado, la inicialización de los pesos de las capas lineales se realiza mediante la técnica de inicialización de *Kaiming*, optimizada para la función de activación LeakyReLU. Esta metodología busca minimizar los problemas de desvanecimiento o explosión de gradientes, situaciones comunes en las redes GAN que pueden comprometer significativamente el aprendizaje del modelo. En particular, estos problemas pueden llevar a que el modelo deje de aprender de manera efectiva, o incluso detener completamente su aprendizaje, a menudo causado por una inicialización de pesos inapropiada.

La elección de la inicialización de *Kaiming* se fundamenta en evitar precisamente este tipo de inconvenientes, garantizando un flujo de gradientes más estable durante el entrenamiento. Esta decisión se tomó después de observar que, en las primeras etapas del desarrollo del proyecto, se presentaban problemas de aprendizaje en la red debido a una inicialización de pesos que no era óptima.

Finalmente, se aplica la activación sigmoidal en la última capa que garantiza que los datos generados se mantengan dentro del rango [0, 1], acorde con los datos normalizados que se manejan

Discriminador

El diseño del discriminador incorpora la normalización espectral en sus capas lineales, una elección que sirve para equilibrar la dinámica entre el generador y el discriminador. Esta técnica ayuda a estabilizar el entrenamiento de la *GAN*, controlando la magnitud de los pesos y evitando que el discriminador se vuelva demasiado dominante. La normalización espectral es particularmente útil en situaciones donde el discriminador supera al generador, lo que puede llevar a un colapso del modelo.

Para añadir no linealidad y robustez, el discriminador también utiliza la activación *LeakyReLU* y capas de *dropout*. La activación *LeakyReLU* permite que se mantenga un flujo pequeño de gradientes, lo que contribuye a mantener la efectividad del entrenamiento. El dropout, por otro lado, reduce el riesgo de sobreajuste al desactivar aleatoriamente ciertas neuronas durante el entrenamiento, lo que obliga al modelo a aprender características más generalizables.

Adicionalmente, se implementó una técnica conocida como *feature matching*. Esta técnica refuerza la habilidad del discriminador para concentrarse en las características clave y sus distribuciones en los datos reales, facilitando así al generador imitar estas propiedades con mayor precisión.

La decisión de incorporar *feature matching* y capas *dropout* surgió como respuesta a las observaciones iniciales, donde se notó que los datos generados no lograban capturar la variabilidad de los datos reales de manera satisfactoria. Con esta implementación, se logró generalidad y variabilidad en los datosa artificiales

9.3.2. Entrenamiento GAN

El entrenamiento de GAN se enfoca en lograr un equilibrio entre el generador y el discriminador. Al principio se detectó que el discriminador dominaba sobre el generador, lo que llevó a implementar ajustes estratégicos, como incrementar la frecuencia de entrenamiento del generador.

En la práctica, el discriminador se entrena tanto con datos reales (añadiendoles un ruido ligero) como con datos falsos generados utilizando el optimizador Adam y la función de pérdida *binary cross entropy*. Las pérdidas se calculan y combinan para ambos tipos de datos para actualizar los pesos del discriminador. El generador, por su parte, se enfoca en crear datos que el discriminador clasifique como reales. La implementación de *feature matching* resultó ser clave, ya que permitió al generador capturar de manera más efectiva la esencia de los datos reales.

Un gran reto en el entrenamiento de GAN es lograr la convergencia entre el generador y el discriminador, ya que un generador demasiado potente puede llevar a resultados no óptimos, mientras que un discriminador más fuerte puede impedir que el generador aprenda eficazmente. Al aumentar la frecuencia de entrenamiento del generador, se buscaba equilibrar esta dinámica, permitiendo que el generador adquiera más peso y efectividad.

Estos ajustes y mejoras en la arquitectura y estrategia de entrenamiento han resultado en una notable mejora en la calidad de los datos generados, demostrando la eficacia de las GAN en la generación de datos tabulares artificiales cosa que se evaluará en siguientes apartados.

9.3.3. Redes generativas adversativas condicionales (CGAN)

Las CGANs son una variante de las GANs tradicionales, diferenciándose por incorporar información que las condiciona, como etiquetas o características específicas que harán que genere una parte de la población. Esta condición permite dirigir y especificar la generación de datos hacia categorías o atributos particulares.

La arquitectura y el entrenamiento de las CGANs siguen los mismos principios que las GANs convencionales, con la única variación de incluir y gestionar la información condicional. Este enfoque hace a las CGANs herramientas muy efectivas para tareas que sufran un desbalance de datos y generación de datos, donde se necesitan resultados más focalizados y ajustados a condiciones específicas que afecten a una parte de la población de los datos reales.

9.4. Variational Autoencoder (VAE)

El Variational Autoencoder (VAE) implementado en la librería GATDM es una herramienta poderosa para aprender representaciones profundas y generativas de datos tabulares. Utilizando una arquitectura de codificador-decodificador, el VAE aprende a comprimir los datos en un espacio latente y luego los reconstruye, manteniendo las características estadísticas clave de los datos originales. A continuación se explicará en detalle esta arquitectura y la estrategia de entrenamiento que se ha empleado.

9.4.1. Arquitectura VAE

La arquitectura del VAE consta de dos partes principales: un codificador y un decodificador como se puede observar en la Figura 7.

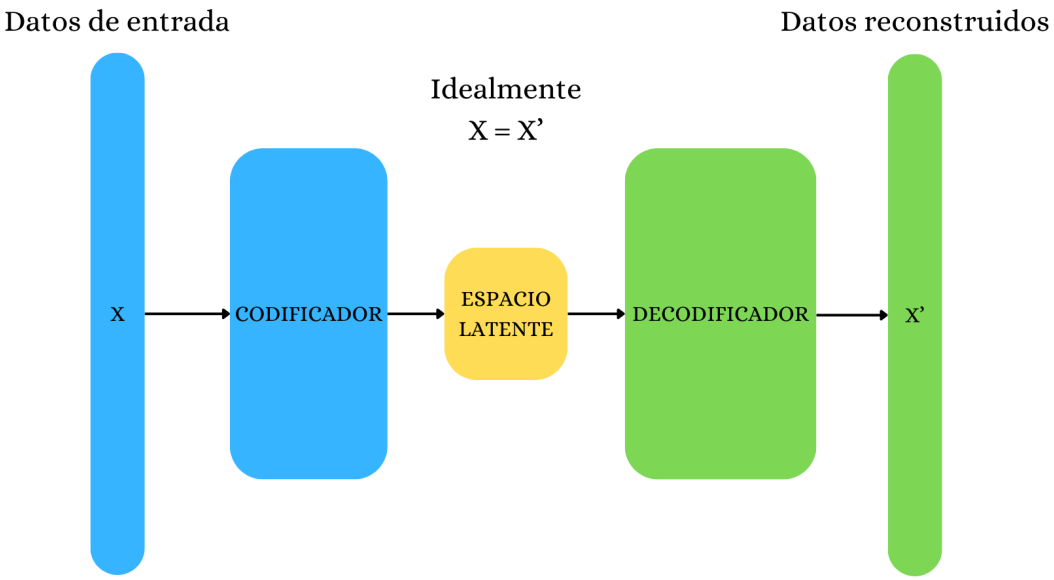


Figura 7: Arquitectura de VAE
Fuente: Elaboración propia

Codificador

La arquitectura del codificador del VAE transforma los datos de entrada en un espacio latente, proporcionando dos conjuntos de valores clave: la media y la desviación estándar logarítmica. Estos valores serán esenciales para representar la distribución de los datos en este espacio latente reducido. El uso de la media y la desviación estándar logarítmica en este contexto es fundamental para el proceso de reparametrización del VAE, permitiendo la generación de nuevas muestras a partir del espacio latente de manera eficiente y controlada.

La arquitectura del codificador se construye con capas lineales seguidas de activaciones *ReLU*, que son cruciales para introducir no linealidades en el modelo. Esto permite que el codificador capture relaciones complejas y no lineales dentro de los datos. La elección de *ReLU* como función de activación se debe a su eficacia en evitar el problema del desvanecimiento de gradientes, manteniendo así un aprendizaje efectivo durante el entrenamiento. Estas activaciones fueron incorporadas después de observar que al principio el modelo producía resultados totalmente lineales y esto no era conveniente debido a que era necesaria la no linealidad para capturar mejor la variabilidad de los datos reales.

Seguidamente, se incorporaron técnicas de regularización como *BatchNorm* y *dropout* en el codificador. *BatchNorm* ayuda a estabilizar y acelerar el proceso de aprendizaje normalizando las entradas de cada capa, mientras que el dropout añade un elemento de aleatoriedad al proceso de entrenamiento, desconectando aleatoriamente ciertas neuronas. Esto evita que el modelo se sobreajuste a los datos de entrenamiento, fomentando así una mejor generalización a nuevos datos.

La inclusión de estas capas de regularización se decidió tras observar que, al igual que en la red GAN, el modelo inicial tendía a no capturar adecuadamente la variabilidad de los datos reales y mostraba signos de sobreajuste.

En conjunto, estas decisiones de diseño arquitectónico y estrategias de regularización aseguran que el codificador del VAE sea capaz de aprender representaciones latentes efectivas y robustas de los datos reales, produciendo así datos de mayor calidad.

Decodificador

El decodificador del VAE desempeña un papel fundamental en el proceso de generación de datos, tomando los valores del espacio latente que genera el codificador y los reconstruye en una salida similar a la de los datos reales. Esta reconstrucción es vital para aprender a replicar la distribución de los datos de entrada.

La arquitectura del decodificador refleja la del codificador, utilizando capas lineales con activaciones *ReLU*. Esta simetría garantiza que el proceso de codificación y decodificación sea coherente y equilibrado. Las activaciones *ReLU*, al igual que en el codificador, permiten al decodificador captar y reproducir las complejidades y no linealidades de los datos.

El aspecto distintivo del decodificador es su capa final, que emplea una función de activación sigmoidal. Esta elección es estratégica, ya que la función *Sigmoid* comprime las salidas a un rango entre 0 y 1 ya que es la estandarización que se escogió durante el preprocesado de datos. Esto asegura así que los datos generados por el VAE sean directamente comparables con los datos reales.

La técnica de reparametrización es un componente fundamental en la estructura del VAE que se ha implementado, y juega un papel crucial para hacer viable el entrenamiento del modelo. En un VAE, la reparametrización se utiliza para abordar el reto que implica entrenar con capas estocásticas, como es el caso de la capa que genera la distribución latente.

En esta implementación, la reparametrización se realiza en la capa latente del modelo. Esta capa produce dos salidas: una que representa las medias (μ) y otra para las desviaciones estándar logarítmicas (log(σ)). Estos dos valores definen una distribución normal desde la cual el modelo muestrea para generar los datos.

En lugar de muestrear directamente de esta distribución, lo cual sería un proceso estocástico difícil de entrenar mediante *backpropagation*, se aplica la técnica de reparametrización.

La reparametrización funciona generando primero un valor aleatorio (e) de una distribución normal estándar, y luego escalando y trasladando este valor con las medias y desviaciones estándar producidas por el modelo. Matemáticamente, esto se expresa como:

$$z = \mu + \epsilon \times e^{log(\sigma)} = \mu + \epsilon \times \sigma$$

Este enfoque mantiene la aleatoriedad necesaria para la generación de datos, pero de una manera que permite que el gradiente fluya a través de la red durante el entrenamiento. Así, facilitamos el entrenamiento eficiente del modelo mediante *backpropagation*, manteniendo al mismo tiempo la capacidad del VAE para aprender representaciones complejas y variadas de los datos de entrada.

9.4.2. Entrenamiento VAE

El proceso de entrenamiento del VAE se maneja con cuidado para asegurar una buena generación de datos. El entrenamiento involucra los siguientes aspectos clave:

- **Función de pérdida**: La función de pérdida se compone de dos elementos principales: la pérdida de reconstrucción y la divergencia Kullback-Leibler (KL). Esta combinación se representa matemáticamente como:

$$Loss_{VAE} = Loss_{recon} + \beta \, x \, Loss_{KL}$$

Donde:

1. **Pérdida de reconstrucción ($Loss_{recon}$)** [8]: Es la medida del error cuadrático medio (MSE) entre los datos originales x y los datos reconstruidos $\hat{x}$ por el VAE. Matemáticamente, se calcula como la suma del cuadrado de las diferencias entre los valores originales y reconstruidos, dividida por el número total de observaciones:

$$Loss_{recon} = \frac{1}{2}\Sigma_{i=1}^{N}(x_i - \hat{x}_i)^2$$

2. **Divergencia Kullback Leibler ($Loss_{KL}$)** [9]**:** Representa cuánto se desvía la distribución aprendida por el VAE en el espacio latente, viene dada por la siguiente fórmula:

$$Loss_{recon} = \frac{1}{2}\Sigma_{i=1}^{N}(1 + log(\sigma_i^2) - \mu_i^2 - \sigma_i^2)$$

3. **Factor beta(β)**. Es un hiperparámetro que pondera la importancia de la divergencia KL en relación con la pérdida de reconstrucción. Un valor más alto de β da más importancia a la regularización impuesta por la divergencia KL.

- **Optimización**: Se utiliza el optimizador Adam para ajustar los pesos de la red. Además, se implementa una técnica de *early stopping* para prevenir el sobreajuste, deteniendo el entrenamiento si la pérdida de validación no mejora después de un número predefinido de épocas.

- **Incremento de β**: Durante el entrenamiento se observó que el modelo priorizaba la reconstrucción sobre la variabilidad de los datos así que se implementó β. El valor de β se incrementa gradualmente hasta un máximo predefinido. Este enfoque ayuda a estabilizar el entrenamiento inicial y mejora gradualmente la fidelidad de los datos generados.

La combinación de estas técnicas hace que el VAE sea una herramienta eficaz para aprender complejas distribuciones de datos y generar nuevos ejemplos que reflejen las características estadísticas de los conjuntos de datos tabulares. Estas técnicas se han ido aplicando después de ir viendo los fallos que iba cometiendo el modelo, como sobreajuste o la priorización de la reconstrucción sobre la variabilidad.

9.5. Bit Diffusion

El modelo Bit Diffusion, adaptado de las técnicas de difusión de imágenes, representa un enfoque innovador en la generación de datos tabulares. Este *Bit Diffusion* se basa en el trabajo realizado en *Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning* [10] el cual se centra en la generación de imágenes artificiales y no datos tabulares artificiales que es lo que se pretende conseguir. Esta implementación se ha desarrollado para capturar y reproducir la complejidad inherente a los conjuntos de datos.

9.5.1. Arquitectura Bit Diffusion

La arquitectura de Bit Diffusion implementada en este proyecto (Figura 8) es una adaptación única del modelo de difusión utilizado en la generación de imágenes [11].

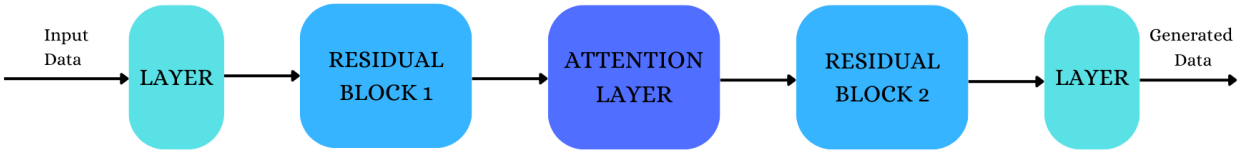


Figura 8: Arquitectura de Bit Diffusion

Fuente: Elaboración propia

Esta versión está específicamente diseñada para manejar la complejidad y las características de los datos tabulares. Los elementos clave de esta arquitectura incluyen:

Capa de entrada

La capa de entrada es la primera fase de procesamiento en la red Bit Diffusion. Esta capa tiene la función esencial de transformar los datos de entrada, que están en su dimensión original, a un espacio latente más adecuado para el procesamiento posterior por la red.

Se implementa utilizando una capa lineal (nn.Linear en *PyTorch*), que mapea la dimensión de entrada a una dimensión oculta (*hidden_dim*). Esta transformación es fundamental para preparar los datos para las etapas de procesamiento más avanzadas que siguen.

Bloque residual

Los bloques residuales representan un componente clave para la conservación y transmisión efectiva de información a través de las distintas capas de la red. Su estructura y función se detallan a continuación para proporcionar una comprensión más profunda de su papel en el modelo.

La estructura se compone de varias capas lineales, cada una seguida por una activación *ReLU*. Esta función de activación, es crucial para introducir elementos no lineales en el modelo sin comprometer la velocidad del proceso de entrenamiento. La no linealidad es un aspecto vital para que la red pueda captar y modelar relaciones complejas presentes en los datos y captar en su totalidad la variabilidad de los datos, cosa que al principio no lograba hacer, como pasaba en los otros modelos descritos anteriormente.

Lo que diferencia a estos bloques es la suma de la entrada inicial del bloque al resultado obtenido tras pasar por todas las capas. Esta suma directa de la entrada original a la salida procesada del bloque es lo que se conoce como "conexión residual" y es el aspecto distintivo de estos bloques.

Esta técnica de conexión residual es fundamental para contrarrestar el problema de desvanecimiento de gradientes que ha sido mencionado anteriormente. Al sumar la entrada original del bloque al resultado de las capas, se asegura que parte de la información pase sin cambios a través de la red, ayudando a mantener los gradientes dentro de un rango adecuado durante el entrenamiento.

Bloque de atención

La capa de atención es un componente crucial que permite al modelo analizar y entender las relaciones complejas dentro de los datos tabulares. Su implementación y funcionamiento se detallan a continuación, mostrando cómo esta capa mejora significativamente la capacidad del modelo para generar datos realistas y coherentes.

La capa comienza con el mecanismo de atención multi-cabeza, donde se divide la atención en varias 'cabezas', cada una concentrándose en diferentes aspectos de los datos. Esta estructura permite que el modelo procese y entienda múltiples relaciones en los datos simultáneamente, lo que aumenta su capacidad para capturar la complejidad de los conjuntos de datos tabulares.

Para cada cabeza, se generan las consultas, claves y valores (Q, K, V) mediante una transformación lineal de los datos de entrada. Este proceso se realiza utilizando la operación *to_qkv*, que transforma los datos y luego los divide en tres partes iguales, asignando una a cada uno de estos componentes críticos de la atención. Una vez generados, se reorganizan para adaptarse al cálculo de la atención.

El cálculo de la atención se realiza mediante un producto escalar entre las consultas y las claves, seguido de una normalización *softmax*. Esto determina la importancia relativa de cada parte de los datos, permitiendo que el modelo se concentre en las áreas más relevantes. Los pesos de atención resultantes se aplican luego a los valores, seleccionando y ponderando características clave basadas en su relevancia.

Una característica única de esta capa de atención es cómo maneja los datos tabulares, que normalmente están en formato 2D. Para adaptar la atención que suelen estar diseñadas para datos 3D como imágenes, se añade una dimensión adicional a los datos, transformándolos de 2D a 3D. Este paso es esencial, ya que permite que el mecanismo de atención funcione de manera efectiva, ya que está diseñado para procesar datos con esa forma tridimensional. Una vez que la atención se ha aplicado, se elimina esta dimensión adicional, devolviendo los datos a su estructura original 2D.

Capa de salida

La capa de salida es la etapa final en la arquitectura Bit Diffusion. Su función es transformar los datos procesados por la red nuevamente a la dimensión original de los datos de entrada.

Esta utiliza otra capa lineal para revertir la transformación hecha por la capa de entrada, seguida de una activación sigmoidal. La activación sigmoidal, como en los modelos anteriores, permite que los datos de salida estén entre 0 y 1.

Estas capas y bloques permiten que el modelo procese y genere datos capturando tanto las características de nivel bajo como las relaciones complejas presentes en los conjuntos de datos.

9.5.2. Entrenamiento Bit Diffusion

El proceso de entrenamiento se basa en varios principios clave, que se detallan a continuación:

- **Aplicación de ruido**: Durante el entrenamiento, se introduce ruido en los datos de entrada. Esta técnica consiste en añadir un elemento aleatorio a los datos, lo que es crucial para enseñar al modelo a reconstruir los datos originales incluso cuando están distorsionados o alterados.

  La cantidad de ruido añadido se controla mediante un nivel de ruido máximo, que se ajusta de forma aleatoria en cada lote de datos. Este enfoque permite que el modelo aprenda a manejar y compensar las variaciones y perturbaciones en los datos.

- **Optimización y pérdidas**: Para la optimización del modelo, se emplea el algoritmo Adam como en anteriores modelos, una elección común para entrenar redes neuronales debido a su eficiencia y capacidad para manejar gradientes dispersos.

  La función de pérdida utilizada es la pérdida de error cuadrático medio (MSE). Esta métrica calcula la diferencia entre los datos reconstruidos por el modelo y los datos originales, la cual proporciona una medida cuantitativa de la precisión de la reconstrucción del modelo.

- **Ciclos iterativos de aprendizaje**: El modelo se entrena a lo largo de varias épocas, en las que ajusta gradualmente sus parámetros para mejorar la precisión de la reconstrucción de datos. En cada época, se procesa el conjunto completo de datos, aplicando ruido y entrenando el modelo para reconstruir los datos de entrada a partir de sus versiones ruidosas.

A través de la aplicación de ruido y el refinamiento iterativo, el modelo aprende a capturar las características esenciales de los datos reales, lo que le permite generar versiones sintéticas que mantienen la integridad y las cualidades fundamentales de los datos originales. Los hiperparámetros ajustados en el entrenamiento se han ido probando y finalmente, se han quedado los que mejores resultados daban.

9.6. Data Augmentation

El modelo llamado *Data Augmentation* implementado en este proyecto se enfoca en enriquecer el conjunto de datos existente mediante la creación de variantes modificadas de los datos originales utilizando técnicas de los modelos *denoising autoencoders*. Este enfoque ayuda a mejorar la robustez y la generalización de los modelos que utilizan estos datos para el entrenamiento.

9.6.1. Arquitectura Data Augmentation

La arquitectura de este modelo se caracteriza por su simplicidad y eficacia. La estructura es la siguiente:

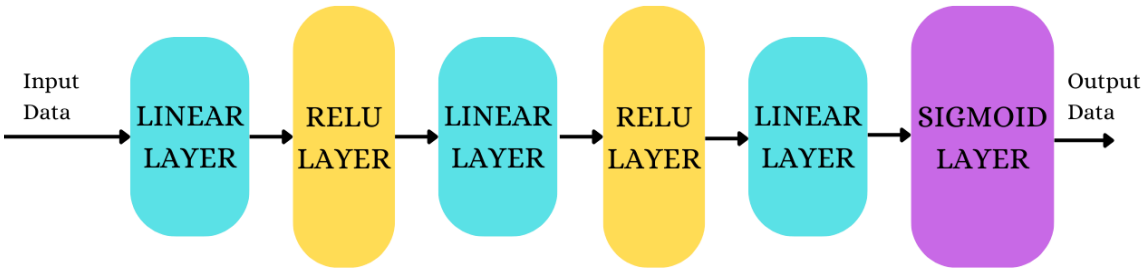


Figura 9: Arquitectura de Data Diffusion

Fuente: Elaboración propia

El modelo inicia con una capa lineal que transforma los datos de entrada al espacio de un tamaño de dimensión oculta, esta transformación es seguida por una activación *ReLU*. La secuencia de capas lineales y activaciones *ReLU* se repite, permitiendo al modelo aprender y adaptar los datos de entrada de manera efectiva.

La última capa del modelo es también una capa lineal seguida por una activación sigmoidal también para garantizar que los datos estén entre 0 y 1.

Además de la estructura básica del modelo, se incluye una función que permite generar datos aumentados a partir de muestras aleatorias. Esto facilita la creación de datos nuevos y variados, basados en las características aprendidas por el modelo.

9.6.2. Entrenamiento Data Augmentation

El proceso de entrenamiento de este modelo se centra en enseñar al modelo a generar versiones modificadas de los datos de entrada. Para esto, se siguen los siguientes pasos:

- **Aplicación de ruido y máscara**: Se añaden dos técnicas de aumentación de datos, la aplicación ruido y máscara aleatoria. El ruido se aplica a los datos de entrada, simulando variaciones menores, mientras que la máscara aleatoria oculta partes de los datos, forzando al modelo a reconstruir estas áreas.

- **Optimización y función de pérdida**: Se utiliza el optimizador Adam y la función de pérdida de error cuadrático medio (MSE) para ajustar los parámetros del modelo. Esto permite que el modelo aprenda a reconstruir los datos originales a partir de las versiones ruidosas y enmascaradas.

- **Ciclos iterativos de aprendizaje**: El modelo se entrena a través de múltiples épocas, donde en cada iteración ajusta sus parámetros para mejorar la calidad de los datos generados. La pérdida se calcula comparando los datos reconstruidos con los originales, lo que permite evaluar y mejorar continuamente la capacidad de generación del modelo.

Este modelo ofrece una herramienta valiosa para enriquecer conjuntos de datos tabulares, mejorando la diversidad y calidad de los datos disponibles para tareas de aprendizaje automático y análisis de datos. Sin embargo, no siempre es correcto aplicar un modelo tan sencillo para generar datos, esto es debido a que su simplicidad permite generar datos solo a través de leves modificaciones en los datos reales.

10. Evaluación de los modelos

En la evaluación de modelos de generación de datos artificiales, es fundamental contar con métricas precisas que puedan medir la efectividad de estos modelos en replicar la distribución de los datos reales y visualizaciones donde se puedan comparar los datos reales con los datos generados. Este apartado se centrará en las métricas y visualizaciones utilizadas para evaluar los modelos presentados en este trabajo.

10.1. Métricas

Las métricas son herramientas esenciales para cuantificar la calidad y la precisión de los datos generados por los modelos, dos métricas comúnmente utilizadas en este contexto son la prueba de *Kolmogorov-Smirnov* y la prueba *Chi-square* [12].

10.1.1. Prueba de Kolmogorov-Smirnov

La prueba de *Kolmogorov-Smirnov* se ha empleado como una de las métricas clave en este proyecto para evaluar la calidad de los datos generados artificialmente. Esta es reconocida como una de las herramientas estadísticas más efectivas y ampliamente utilizadas, su prueba permite comparar las distribuciones de dos conjuntos de datos independientes.

El núcleo de esta prueba es el cálculo del estadístico D que representa la máxima diferencia absoluta entre las funciones de distribución acumulativa empírica (*CDF*) de los dos conjuntos de datos comparados. Matemáticamente, se define de la siguiente manera:

$$D = max_x |F_{1,n}(x) - F_{2,n}(x)$$

Aquí $F_{1,n}(x)$ y $F_{2,n}(x)$ son las *CDFs* empíricas de los datos reales y generados. Este cálculo se aplica individualmente a cada característica dentro de los conjuntos de datos, permitiendo un análisis detallado y dimensional de las similitudes y diferencias en sus distribuciones.

Un valor elevado de D indica una mayor discrepancia entre las dos distribuciones. Sin embargo, para determinar la significancia estadística de esta diferencia, se utiliza el valor 'p' asociado al estadístico 'D'.

El valor p en la prueba de Kolmogorov-Smirnov indica la probabilidad de observar una diferencia tan grande como D (o mayor) entre las distribuciones, bajo la hipótesis nula de que ambas muestras provienen de la misma distribución. Un valor p bajo (generalmente menor que 0.05) sugiere que las diferencias observadas son significativas y que las muestras probablemente provienen de diferentes distribuciones. Por otro lado, un valor p alto implica que no hay evidencia suficiente para rechazar la hipótesis nula y se puede considerar que las muestras son de distribuciones similares.

En la práctica, esta prueba se ha aplicado comparando independientemente cada característica de los datos generados con su contraparte en los datos reales. Los resultados de la prueba de *Kolmogorov-Smirnov* proporcionan una base sólida para evaluar cuán bien los modelos generativos pueden replicar las características estadísticas de los datos reales, siendo una métrica fundamental para validar la calidad de los datos sintéticos producidos.

10.1.2. Chi-square

La prueba *Chi-square* es una métrica esencial para evaluar la similitud entre las distribuciones de variables categóricas. Esta prueba compara las frecuencias observadas en los datos con las frecuencias que se esperarían si los datos generados siguieran la misma distribución que los datos reales. Matemáticamente, el estadístico *Chi-square* χ^2 se calcula como:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

donde O_i es la frecuencia observada y E_i es la frecuencia esperada para cada categoría. Un valor p bajo en esta prueba indica una diferencia significativa entre las distribuciones observadas y esperadas, sugiriendo que el modelo no ha replicado con precisión las características categóricas de los datos originales.

10.2. Visualización

La visualización juega un papel crucial en la evaluación y comparación de los datos generados y reales. En este proyecto, se han empleado distintas técnicas de visualización para analizar las diferencias y similitudes entre los datos reales y los datos artificiales. A continuación, se describen las técnicas utilizadas:

10.2.1. Principal Components

La técnica del análisis de componentes principales (PCA) ha sido utilizada como una herramienta clave para evaluar y comparar los datos tabulares reales con los generados por los modelos.

Al aplicar PCA a los conjuntos de datos tanto reales como generados, conseguimos reducir su complejidad dimensional, proyectando estos datos en un espacio de dos dimensiones. Esta proyección se centra en las dos primeras componentes principales, que son las direcciones en el espacio de características que retienen la mayor varianza de los datos. Al visualizar estos datos proyectados, se obtiene una perspectiva clara y directa sobre cómo los datos generados se alinean con los datos reales en términos de su estructura y distribución.

Esta comparación visual es crucial para entender la eficacia de los modelos generativos. Si los datos generados se alinean estrechamente con los datos reales en el espacio de PCA, esto indica que los modelos han sido capaces de capturar y replicar las principales tendencias y variaciones presentes en los datos originales. En otras palabras, una distribución similar en el espacio de PCA sugiere que los patrones y relaciones fundamentales en los datos reales han sido efectivamente modelados y recreados por los generativos.

10.2.2. Distributed Stochastic Neighbor Embedded

En este proyecto, la técnica t-SNE ha sido fundamental para evaluar cómo los modelos generativos reproducen las complejidades y características de los datos reales. Al trabajar con datos de alta dimensión, como es el caso de los conjuntos de datos tabulares, es esencial disponer de métodos que permitan visualizar y comparar de manera efectiva las estructuras y patrones subyacentes en los datos.

t-SNE, con su habilidad para preservar estructuras locales y revelar agrupaciones, se convierte en una herramienta poderosa en este contexto. Al aplicar t-SNE tanto a los datos reales como a los generados, se pueden proyectar en un espacio bidimensional, facilitando la visualización y comparación directa de sus estructuras. Esta representación bidimensional ayuda a identificar si los datos generados por los modelos han capturado fielmente las agrupaciones y relaciones presentes en los datos reales.

Por ejemplo, si los datos reales exhiben ciertos grupos o patrones, esperaríamos que los datos generados reflejen de manera similar estas características. La habilidad de t-SNE para conservar las relaciones locales entre puntos hace que sea una técnica particularmente adecuada para esta tarea. Mediante la visualización de los datos en un plano bidimensional, se facilita enormemente la interpretación y el análisis comparativo, permitiendo identificar visualmente las similitudes y diferencias entre los datos originales y los sintéticos.

10.2.3. Gráfico de densidad

Los gráficos de densidad son herramientas valiosas para comparar las distribuciones de datos en una o varias dimensiones. Se han generado gráficos de densidad para cada característica individual, así como un gráfico combinado para todas las características, tanto para los datos reales como para los generados. Estos gráficos proporcionan una comparación visual detallada de las distribuciones, permitiendo identificar similitudes y diferencias en la densidad de los datos en varios rangos de valores. Esto nos sirve también para comprobar si los datos artificiales son capaces de generar muestras en los grupos faltantes de los datos reales.

Además, para un análisis más profundo, se han creado gráficos de barras para cada variable categórica, comparando las frecuencias de las categorías en los datos reales y generados. Estos gráficos facilitan la evaluación de la capacidad del modelo para replicar con precisión las proporciones de categorías presentes en los datos originales.

11. Tests y resultados

En este apartado, se presentan los tests realizados para evaluar el rendimiento de los modelos generativos implementados en el proyecto. Estas pruebas son fundamentales para entender la eficacia y adaptabilidad de cada modelo bajo distintas condiciones de tamaño de datos y número de épocas de entrenamiento.

Los hiperparámetros generales constantes utilizados en todas las pruebas son:

- Tasa de aprendizaje (*LEARNING_RATE*): 0.001
- Tamaño del lote (*BATCH_SIZE*): 32
- División de validación (*VALIDATION_SPLIT*): 0.2

Estos parámetros ayudan a comparar los resultados entre los distintos modelos. Además, se diferenciarán los tests entre conjuntos de datos pequeños y grandes, y se variará el número de épocas para observar la influencia en el aprendizaje y la posibilidad de sobreajuste. Estas pruebas proporcionan resultados valiosos para futuras investigaciones y mejoras en la generación de datos artificiales.

11.1. Conjuntos de datos pequeños

En la siguiente sección, se examina el comportamiento de los modelos frente a conjuntos de datos de tamaño reducido que contienen distribuciones sencillas. Este enfoque nos permite evaluar la capacidad de los modelos para capturar la variabilidad de los datos en escenarios limitados, lo cual es crucial para comprender su eficacia en entornos de datos restringidos.

El conjunto de datos elegido para esta prueba es el conjunto de datos Iris que contiene tres grupos de datos bien diferenciados (más adelante reducido a dos para darle simplicidad al conjunto de datos). Este *dataset* incluye cinco variables numéricas y una variable categórica. Sin embargo, ha sido reducido el número total de columnas a cuatro, eliminando la columna que actúa como identificador y las columnas que presentaban distribuciones similares. Esto se hizo con el propósito de evaluar cómo reaccionan nuestros modelos ante diversas distribuciones sencillas de datos y relaciones poco complejas.

Para estos tests, ha sido configurado el siguiente hiperparámetro:

- Número de muestras generadas (*NUM_SAMPLES*): 100

Este ajuste se realiza con el objetivo de mantener una proporción cercana a la del conjunto de datos original, el cual consta de aproximadamente 150 muestras. Esto proporciona una comparación equitativa entre los datos generados y los datos reales, permitiendo una evaluación más precisa de la eficiencia de los modelos en la generación de datos artificiales bajo restricciones de tamaño.

11.1.1. Test de GAN

Ahora será examinada la arquitectura GAN a través de pruebas con distintas épocas de entrenamiento. Estos tests revelarán cómo mejora o varía el rendimiento del modelo a lo largo del tiempo, proporcionando información valiosa sobre su eficacia y la duración óptima de entrenamiento. Para las pruebas realizadas con la arquitectura GAN, se han establecido los siguientes hiperparámetros específicos que son los que mejores resultados han proporcionado:

- Dimensión oculta (*GAN_HIDDEN_DIM*): 128
- Tasa de aprendizaje (*GAN_LEARNING_RATE*): 0.0002
- Devaluación de los pesos (*GAN_WEIGHT_DECAY*): 1e-5

Generación de datos con 10 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 10

Pérdidas del generador y discriminador

El primer indicador que observamos son las pérdidas del generador y discriminador.

```
Epoch [1/10], Step [1/4], D Loss: 1.3842649459838867, G Loss: 0.6994814276695251
Epoch [2/10], Step [1/4], D Loss: 1.3824641704559326, G Loss: 0.7049832940101624
Epoch [3/10], Step [1/4], D Loss: 1.3764405250549316, G Loss: 0.7114260792732239
Epoch [4/10], Step [1/4], D Loss: 1.3761348724365234, G Loss: 0.7164302468299866
Epoch [5/10], Step [1/4], D Loss: 1.3754847049713135, G Loss: 0.7229324579238892
Epoch [6/10], Step [1/4], D Loss: 1.3746525049209595, G Loss: 0.7241939306259155
Epoch [7/10], Step [1/4], D Loss: 1.370042324066162, G Loss: 0.7343989014625549
Epoch [8/10], Step [1/4], D Loss: 1.368981957435608, G Loss: 0.7421402335166931
Epoch [9/10], Step [1/4], D Loss: 1.373640537261963, G Loss: 0.7479435801506042
Epoch [10/10], Step [1/4], D Loss: 1.3759585618972778, G Loss: 0.7485883235931396
```

Figura 10: Captura del log que muestra las pérdidas por época de generador y discriminador.

Fuente: Captura del log, elaboración propia.

En la Figura 10 se observa que la pérdida del discriminador tiende a disminuir con cada época, lo que puede indicar que está logrando diferenciar con mayor precisión entre los datos reales y los generados. Por otro lado, la pérdida del generador muestra una tendencia al alza, sugiriendo que aún lucha por generar datos que el discriminador clasifique como reales.

Esta situación señala que aún no se ha alcanzado un estado de convergencia entre el discriminador y el generador. Idealmente, se busca un punto de equilibrio donde ambos tengan un desempeño similar, lo que indicaría que el generador ha aprendido a crear datos indistinguibles de los reales para el discriminador. La tendencia actual sugiere que el generador necesita mejorar y adaptar sus generaciones para engañar más efectivamente al discriminador

Prueba de Kolmogorov-Smirnov

La prueba de Kolmogorov-Smirnov se ha aplicado para comparar la distribución de las características de los datos reales y generados. Se han obtenido los siguientes resultados estadísticos y valores p:

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.34	0.026	0.42	0.002	0.29	0.084	0.31	0.054

Tabla 8: Resultados Kolmogorov-Smirnov para GAN (10 épocas)
Fuente: Elaboración propia

Para la primera característica, el estadístico de *Kolmogorov-Smirnov* es 0.34 con un valor p de 0.026, lo que sugiere que hay diferencias estadísticamente significativas entre las distribuciones de los datos reales y generados para esta característica, aunque no son extremadamente pronunciadas.

La segunda característica muestra un estadístico de 0.42 y un valor p de 0.002, indicando una discrepancia más marcada entre las distribuciones y proporcionando evidencia más fuerte de diferencias significativas.

La tercera característica presenta un estadístico de 0.29 con un valor p de 0.084, lo que implica que las diferencias en las distribuciones para esta característica no son estadísticamente significativas al nivel convencional del 0.05, aunque están cerca del umbral.

Finalmente, la cuarta característica tiene un estadístico de 0.31 y un valor p de 0.054, situándose en un margen que sugiere una posible diferencia significativa, pero que no alcanza el umbral estándar de significación estadística.

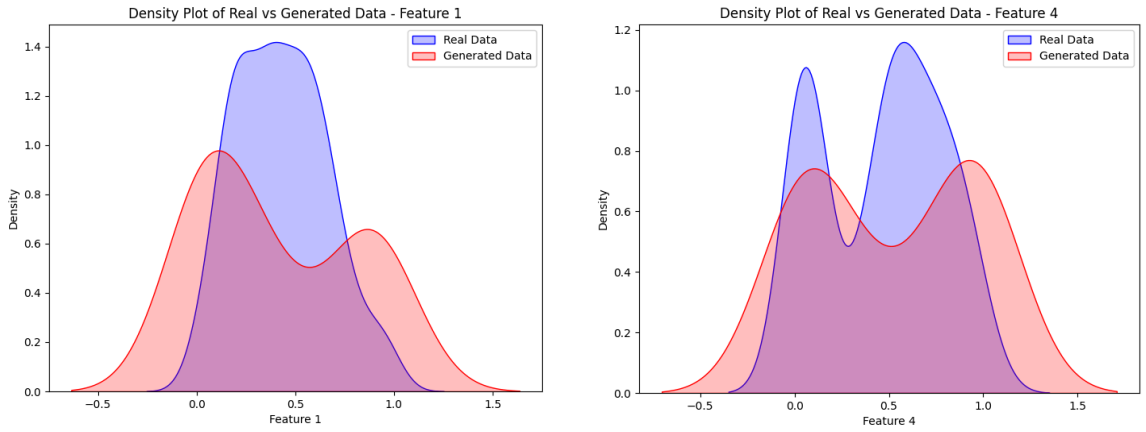
Estos resultados indican que, si bien el modelo ha logrado emular con cierto grado de precisión las distribuciones de los datos reales, aún existen diferencias significativas en al menos dos de las características. El desafío es ajustar el modelo para mejorar la similitud de las distribuciones generadas con las reales, minimizando las diferencias estadísticas detectadas por la prueba de Kolmogorov-Smirnov.

Visualización de los datos reales y de los datos artificiales

Las gráficas de densidad reflejadas en las Figuras 11 y 12 proporcionan una comparación visual entre la distribución de los datos reales y los generados de dos características específicas.

En la primera gráfica correspondiente a la Figura 11, se observa que la densidad de los datos generados sigue una tendencia similar a los datos reales, aunque con diferencias notables en la forma y el pico de la distribución. Esto sugiere que el modelo generativo ha aprendido algunos aspectos de la distribución real, pero aún hay margen de mejora para capturar con precisión las características subyacentes de los datos reales, esto es debido a que se le ha dado un margen de 10 épocas para aprender.

En la segunda gráfica, correspondiente a la Figura 12 se aprecia una mayor superposición entre las distribuciones de los datos reales y generados, lo que indica una mejor capacidad del modelo para replicar esta característica específica del conjunto de datos real. Aunque las dos distribuciones no son idénticas, la superposición sugiere que el modelo está capturando eficazmente la variabilidad y la tendencia central de los datos reales para esta característica.

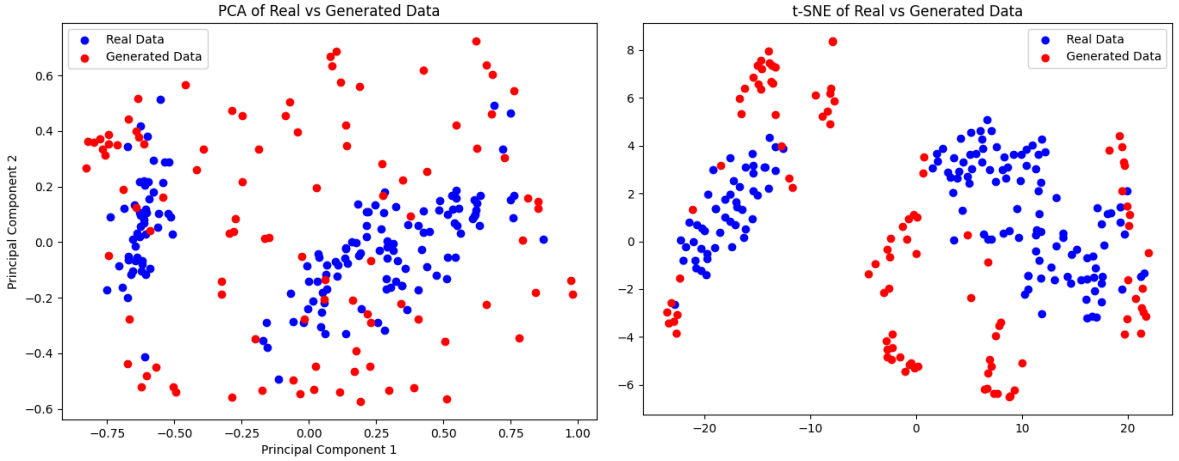


Figuras 11 y 12: Gráficos de densidad de la primera y la cuarta característica real y generada.

Fuente: Elaboración propia usando la librería seaborn.

En el siguiente análisis, se utilizará la visualización de Componentes Principales (PCA) para explorar la variabilidad de los datos y la estructura subyacente de las características. El PCA es particularmente útil para identificar las direcciones de máxima varianza en los datos de alta dimensión y para ver si los datos generados capturan la estructura de covarianza de los datos reales.

Por otro lado, se utilizará el gráfico t-SNE (Distributed Stochastic Neighbor Embedding) para examinar las agrupaciones y la separación de los clusters en el conjunto de datos. t-SNE es efectivo para visualizar la organización y las agrupaciones en espacios de alta dimensión, lo que nos permite observar si los datos generados forman grupos similares a los de los datos reales y si las separaciones entre diferentes clases o categorías se mantienen en el conjunto sintetizado.



Figuras 13 y 14: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn.

En el gráfico de PCA (Figura 13), los puntos azules representan los datos reales y los rojos los datos generados. Se observa una dispersión similar entre ambos, lo que sugiere que las características generales y la variabilidad de los datos sintéticos tienen una correlación cercana con los datos reales. Esto es un indicativo positivo de que el modelo generativo está captando la estructura subyacente de los datos originales.

Por otro lado, en el gráfico de t-SNE (Figura 14), donde también se emplea la misma codificación de colores, se puede observar que los datos generados tienden a formar agrupaciones similares a los datos reales. Esto es especialmente significativo ya que t-SNE es eficaz para visualizar agrupaciones y estructuras locales en conjuntos de datos de alta dimensión, lo que sugiere que el modelo no solo está aprendiendo representaciones globales, sino que también es capaz de captar las complejidades y matices a nivel más granular.

Generación de datos con 100 épocas

A continuación se modificará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 100

Pérdidas del generador y discriminador

El primer indicador que observamos de nuevo son las pérdidas del generador y discriminador.

```
Epoch [85/100], Step [1/4], D Loss: 1.370816707611084, G Loss: 0.7863143682479858
Epoch [86/100], Step [1/4], D Loss: 1.375443935394287, G Loss: 0.7820829749107361
Epoch [87/100], Step [1/4], D Loss: 1.370719313621521, G Loss: 0.7811371088027954
Epoch [88/100], Step [1/4], D Loss: 1.3786063194274902, G Loss: 0.7780135869979858
Epoch [89/100], Step [1/4], D Loss: 1.3613826036453247, G Loss: 0.7873152494430542
Epoch [90/100], Step [1/4], D Loss: 1.3692907094955444, G Loss: 0.7831404805183411
Epoch [91/100], Step [1/4], D Loss: 1.36824369430542, G Loss: 0.7818410992622375
Epoch [92/100], Step [1/4], D Loss: 1.3721699714660645, G Loss: 0.7785015106201172
Epoch [93/100], Step [1/4], D Loss: 1.3701789379119873, G Loss: 0.7813888192176819
Epoch [94/100], Step [1/4], D Loss: 1.3632616996765137, G Loss: 0.7870872616767883
Epoch [95/100], Step [1/4], D Loss: 1.3675196170806885, G Loss: 0.7741745710372925
Epoch [96/100], Step [1/4], D Loss: 1.3646798133850098, G Loss: 0.7821933031082153
Epoch [97/100], Step [1/4], D Loss: 1.3752669095993042, G Loss: 0.7765695452690125
Epoch [98/100], Step [1/4], D Loss: 1.3763768672943115, G Loss: 0.7783728837966919
Epoch [99/100], Step [1/4], D Loss: 1.3722800016403198, G Loss: 0.7811213135719299
Epoch [100/100], Step [1/4], D Loss: 1.3766241073608398, G Loss: 0.7771623134613037
```

Figura 15: Captura del log que muestra las pérdidas de las últimas épocas de generador y discriminador.

Fuente: Captura del log, elaboración propia.

En la Figura 15 se aprecia una estabilidad entre las pérdidas del generador y del discriminador a medida que avanzan las épocas. Este comportamiento sugiere que, sin modificaciones adicionales en la arquitectura o en los hiperparámetros, el generador ha llegado a un límite en su capacidad de mejora, por lo tanto, incrementar el número de épocas más allá de las 100 podría no resultar en avances significativos.

Prueba de Kolmogorov-Smirnov

Seguidamente se realizará la prueba de Kolmogorov Smirnov:

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.18	0.037	0.11	0.4	0.12	0.3	0.12	0.33

Tabla 9: Resultados Kolmogorov-Smirnov para GAN (100 épocas)

Fuente: Elaboración propia

Tras aumentar el número de épocas de entrenamiento de 10 a 100, observamos una mejora significativa en los resultados de la prueba de Kolmogorov-Smirnov de la Tabla 8.

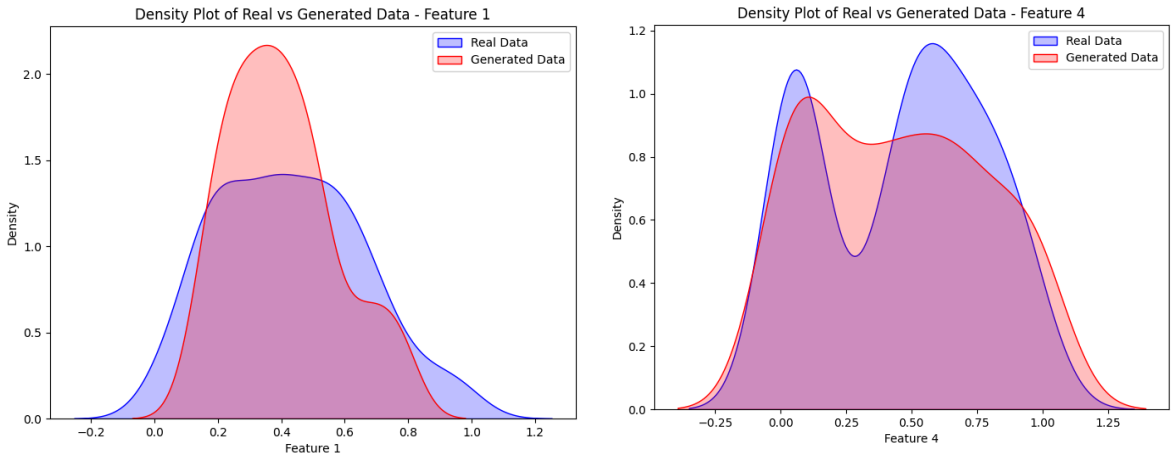
Con 10 épocas, los valores *p* obtenidos fueron generalmente bajos, lo que indica diferencias estadísticamente significativas entre las distribuciones de los datos reales y generados, especialmente en las características 1 y 2, con valores *p* de 0.026 y 0.002 respectivamente. Esto sugiere que el modelo aún no había aprendido a captar adecuadamente la variabilidad de los datos reales y, por lo tanto, generaba muestras que se desviaban significativamente de las reales.

Sin embargo, después de 100 épocas, los valores *p* han aumentado en todas las características, lo que sugiere una disminución en la discrepancia entre los conjuntos de datos. Para la característica 1, el valor *p* pasó de 0.026 a 0.037, lo que aún indica diferencias, pero con menos significancia estadística. Para la característica 2, el valor *p* mejoró considerablemente, pasando de un nivel muy bajo de 0.002 a 0.403, lo que sugiere que no hay evidencia suficiente para afirmar que las distribuciones son diferentes. Las características 3 y 4 también mostraron mejoras, con valores *p* que se movieron de un rango de diferenciación moderada a uno en el que no se puede afirmar con seguridad que exista una diferencia significativa.

Estos cambios indican que el modelo ha mejorado su capacidad de generar datos que se asemejan más a los reales, y que el entrenamiento adicional ha tenido un efecto positivo en la calidad de los datos generados. Aunque aún hay margen para la mejora, particularmente en la característica 1, la tendencia es prometedora y apunta hacia una convergencia entre las distribuciones de los datos reales y generados.

Visualización de los datos reales y de los datos artificiales

Seguidamente se realizará la visualización de las distribuciones:



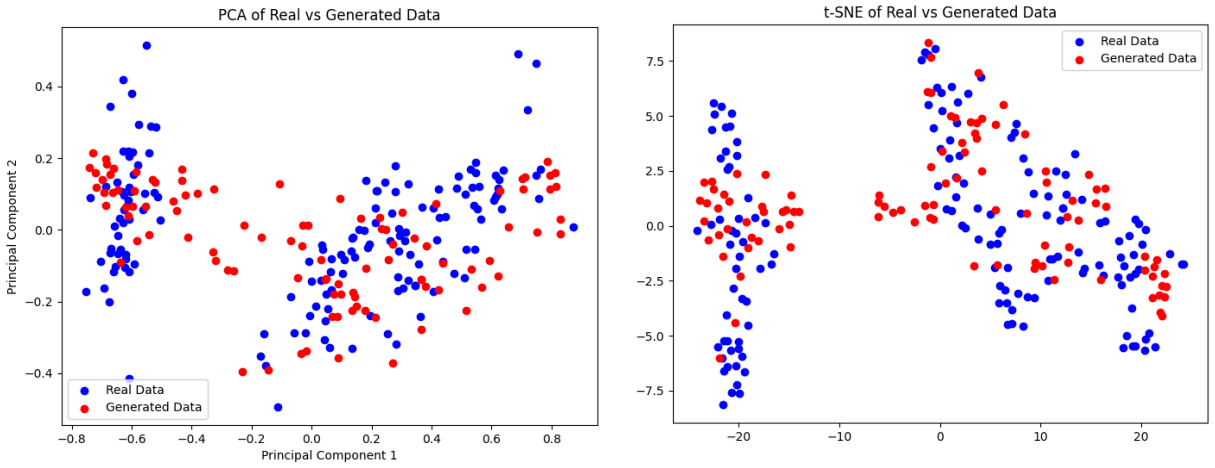
Figuras 16 y 17: Gráficos de densidad de la primera y la cuarta característica real y generada.

Fuente: Elaboración propia usando la librería seaborn.

Observamos en las Figuras 16 y 17 que los picos en los gráficos de densidad de los datos generados se han desplazado más cerca de los correspondientes picos de los datos reales.

Esto sugiere que, con el ajuste de parámetros y un mayor entrenamiento, el modelo está aprendiendo con mayor efectividad las características subyacentes de los datos reales. Los picos representan las modas de las distribuciones y son indicativos de las tendencias centrales en las características de los datos. Una alineación más estrecha de estos picos entre los datos generados y los datos reales implica que la variabilidad y la estructura central de los datos están siendo capturadas de manera más fiel por el modelo generativo.

Además, es relevante mencionar que la presencia de picos de los datos artificiales menos pronunciados y más alineados con los datos reales no sólo mejora la visualización cualitativa, sino que también sugiere una convergencia cuantitativa hacia una distribución más precisa. Esto es indicativo de que el modelo generativo no sólo está aprendiendo a reproducir valores promedio, sino que también está refinando su capacidad para modelar la dispersión y la forma específica de la distribución de los datos reales.



Figuras 18 y 19: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn.

La PCA en la Figura 18, conocida por su eficacia en reducir dimensiones manteniendo la varianza más significativa, muestra en las gráficas una superposición más notable entre los datos reales y los generados. Esto sugiere que, con el aumento de épocas, el modelo ha podido captar con mayor fidelidad la distribución y varianza de los datos originales.

Por otro lado, la visualización t-SNE, en la Figura 19, revela cómo los datos generados reflejan agrupaciones similares a las del conjunto real. Esto es particularmente importante, ya que t-SNE es sensible a la estructura en pequeña escala y su capacidad para mostrar agrupaciones distintas es indicativo de que el modelo ha aprendido a diferenciar entre subgrupos dentro de los datos, acercándose a una representación más auténtica de los datos reales.

Además, se observa que el aumento en el número de épocas de entrenamiento ha mejorado la calidad de los datos generados, lo cual es un indicador de que la continuación del entrenamiento o la optimización de los hiperparámetros puede llevar a mejoras adicionales. Esta progresión hacia una convergencia más estrecha entre los datos reales y generados es un resultado positivo, demostrando que los modelos no solo están aprendiendo patrones generales sino que también están empezando a capturar la complejidad y las sutilezas inherentes a los datos originales.

11.1.2. Test de VAE

Ahora se examinará la arquitectura VAE a través de pruebas con distintas épocas de entrenamiento. Los hiperparámetros seleccionados son:

- Dimensiones ocultas (*VAE_HIDDEN_DIMS*): [128, 256]
- Dimensión Z (*VAE_Z_DIM*): 32
- β máxima (*VAE_MAX_BETA*): 10
- Incremento de β **(***VAE_INCREMENT_BETA*): 0.1
- Paciencia de parada temprana (*EARLY_STOPPING_PATIENCE)*: 10

Generación de datos con 10 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 10

Pérdidas del modelo VAE

Seguidamente se observan las pérdidas del modelo.

```
Epoch[5/10], Step[0/4], Loss: 54.52245330810547, KL: 4.313167095184326, Recon: 31.231351852416992, Beta: 5.399999999999999
Epoch[5/10], Validation Loss: 56.169193267822266
Epoch[6/10], Step[0/4], Loss: 52.32163619995117, KL: 4.428414344787598, Recon: 27.965356826782227, Beta: 5.499999999999998
Epoch[6/10], Validation Loss: 55.826438903808594
Epoch[7/10], Step[0/4], Loss: 44.69834518432617, KL: 2.4329371452331543, Recon: 31.073898315429688, Beta: 5.599999999999998
Epoch[7/10], Validation Loss: 53.67631530761719
Epoch[8/10], Step[0/4], Loss: 44.21644973754883, KL: 1.9928789138793945, Recon: 32.85704040527344, Beta: 5.6999999999999975
Epoch[8/10], Validation Loss: 49.848419189453125
Epoch[9/10], Step[0/4], Loss: 40.8077507019043, KL: 1.9528834819793701, Recon: 29.48102569580078, Beta: 5.799999999999997
Epoch[9/10], Validation Loss: 46.93452072143555
Epoch[10/10], Step[0/4], Loss: 36.24630355834961, KL: 1.2334738969802856, Recon: 28.968807220458984, Beta: 5.899999999999997
Epoch[10/10], Validation Loss: 40.212608337402344
```

Figura 20: Pérdidas del modelo VAE (10 épocas)

Fuente: Captura del log de las pérdidas de VAE, elaboración propia.

En la Figura 20, se puede observar cómo el modelo enfatiza más la reconstrucción de los datos (parámetro *Recon*) que la variabilidad de los datos (parámetro *KL*). Esto implica que no está captando completamente las singularidades de los datos y se está centrando en encontrar el punto óptimo para la reconstrucción, lo que podría dar lugar a que todos los datos generados sean muy similares entre sí.

Prueba de Kolmogorov-Smirnov

Seguidamente se analizarán los resultados de Kolmogorov-Smirnov para VAE.

Primera característica		Segunda característica		Tercera característica		Cuarta característica	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.37	6.4e-08	0.4	3.3e-09	0.46	6.2e-12	0.4	2.37e-09

Tabla 10: Resultados Kolmogorov-Smirnov para VAE (10 épocas)

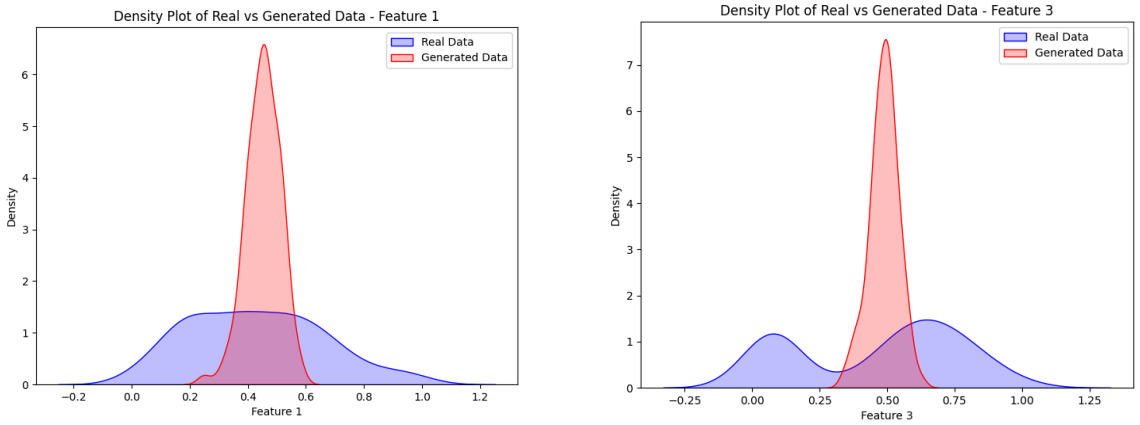
Fuente: Elaboración propia

Los resultados de la prueba de Kolmogorov-Smirnov (KS) para las cuatro características indican de manera consistente diferencias significativas entre los datos generados y las distribuciones reales

Los elevados valores del estadístico KS y los valores p extremadamente bajos en todas las características reflejan que estas no siguen una distribución teórica convencional. Esto sugiere que el modelo no está captando bien la distribución de los datos reales. Esto indica falta de concordancia de las características con las distribuciones de los datos reales.

Visualización de los datos reales y de los datos artificiales

Seguidamente se proseguirá con la visualización de los datos.

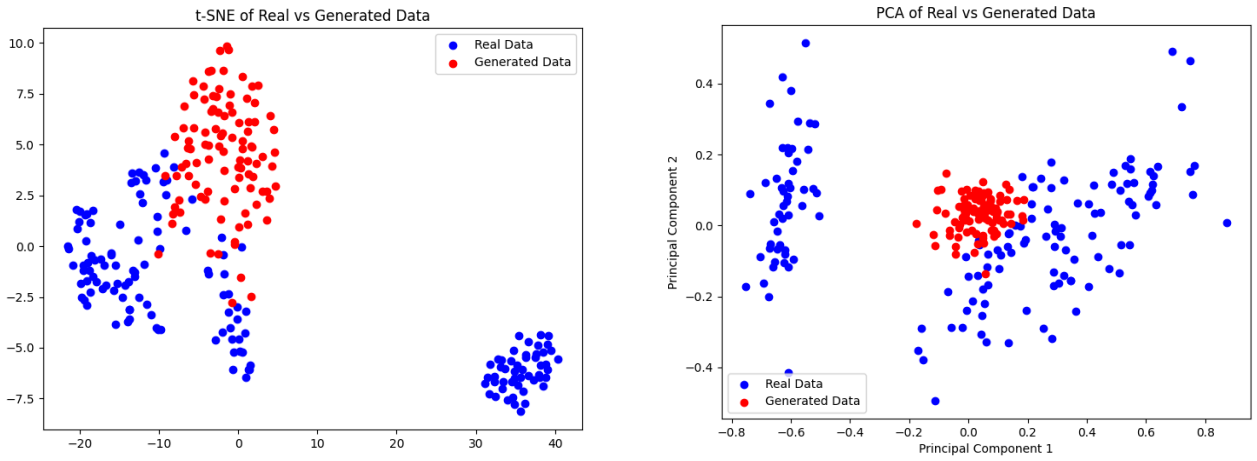


Figuras 21 y 22: Gráficos de densidad de las características 1 y 3 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

Al analizar las gráficas de densidad de los datos reales en comparación con los generados por el modelo VAE de las Figuras 21 y 22, se puede observar una discrepancia notable en la distribución de los datos. El enfoque del modelo VAE en el punto céntrico de los datos se manifiesta en un pico pronunciado en la gráfica de densidad de los datos generados. Este pico central es una indicación de que el modelo está sobreestimando la probabilidad de los valores medios del conjunto de datos, mientras que la variabilidad y la dispersión natural de los datos reales no se reproducen adecuadamente.

En los datos reales, esperaríamos ver una curva de densidad más suave y extendida, lo que indicaría una variedad de valores que contribuyen a la distribución total. En contraste, los datos generados muestran una tendencia a aglomerarse alrededor de un valor central, sugiriendo que el modelo no está capturando la gama completa de posibles variaciones en los datos reales. Este efecto de "pico único" puede ser el resultado de un sobreajuste del modelo a los aspectos más comunes del conjunto de datos de entrenamiento o una limitación en la capacidad del modelo para aprender la estructura subyacente y la variabilidad de los datos.



Figuras 23 y 24: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn.

En la Figura 23, se puede observar que parece que el VAE ha aprendido a reconocer y replicar solo uno de los grupos subyacentes presentes en los datos. Esto es evidente ya que los puntos de datos generados (representados en rojo) se agrupan alrededor de un único grupo, descuidando el otro. Tal comportamiento podría implicar que el modelo se ha centrado en un subconjunto de características que son prominentes en un clúster, pero no ha logrado generalizar la diversidad presente en todo el conjunto de datos.

Por otro lado, la visualización PCA (Figura 24), sugiere una falta de variabilidad en los datos generados. Los datos generados parecen concentrarse alrededor de un punto central en lugar de dispersarse como se observa con los datos reales. Este agrupamiento alrededor de un punto central podría ser indicativo de sobreajuste del modelo, donde el VAE está generando nuevas instancias que están demasiado alineadas con la tendencia central de los datos de entrenamiento, fallando así en reproducir la verdadera varianza y complejidad del conjunto de datos.

Generación de datos con 100 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 100

Pérdidas del modelo VAE

Seguidamente se observanlas pérdidas del modelo con 100 épocas.

```
Epoch[35/100], Step[0/4], Loss: 36.01226806640625, KL: 0.7882103323936462, Recon: 29.39130210876465, Beta: 8.399999999999988
Epoch[35/100], Validation Loss: 31.527053833007812
Epoch[36/100], Step[0/4], Loss: 35.01837158203125, KL: 0.47943827509880066, Recon: 30.943145751953125, Beta: 8.499999999999988
Epoch[36/100], Validation Loss: 31.380634307861328
Epoch[37/100], Step[0/4], Loss: 32.49099349975586, KL: 0.3190775513648987, Recon: 29.746925354003906, Beta: 8.599999999999987
Epoch[37/100], Validation Loss: 31.11282730102539
Epoch[38/100], Step[0/4], Loss: 34.81572723388672, KL: 0.37639376521110535, Recon: 31.541101455688477, Beta: 8.699999999999987
Epoch[38/100], Validation Loss: 31.03205680847168
Early stopping triggered.
```

Figura 25: Pérdidas del modelo VAE (100 épocas)

Fuente: Captura del log de las pérdidas de VAE, elaboración propia.

En la Figura 25, se puede observar lo mismo que observábamos cuando estábamos trabajando con 100 épocas, el modelo se está centrando en la reconstrucción de los datos y no en la variabilidad de los mismos. Esto se observa en los bajos valores de la variable *KL,* la cual cada vez va perdiendo peso hasta que llega un punto en el que se estabiliza y se activa la parada temprana.

Se puede deducir de esta información, que los datos generados por el modelo seguirán teniendo baja variabilidad debido a la importancia que le da el modelo a la reconstrucción y, a pesar de aumentar el valor de β para darle importancia a la variabilidad, no se consigue obtener los resultados óptimos.

Prueba de Kolmogorov-Smirnov

Primera característica		Segunda característica		Tercera característica		Cuarta característica	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.39	9.24e-09	0.29	5.1e-05	0.48	5.3e-13	0.4	4.72e-09

Tabla 11: Resultados Kolmogorov-Smirnov para VAE (100 épocas)
Fuente: Elaboración propia

Comparando los resultados de la prueba *Kolmogorov-Smirno*v entre 10 y 100 épocas, observamos que los valores de estadística *D* no muestran una mejora significativa, y en algunos casos, como en la tercera característica, incluso aumentan, lo que sugiere que la distancia entre las distribuciones real y generada se ha hecho más grande con un mayor número de épocas. Asimismo, los valores *p* son extremadamente bajos en ambas pruebas, indicando que hay diferencias significativas entre las distribuciones comparadas.

La falta de mejora substancial y el incremento en la distancia de la distribución pueden indicar que el modelo VAE está experimentando sobreajuste y podría deberse a varias razones, una de ellas podría ser la falta de datos que se comprobará en los siguientes apartados. En lugar de capturar la verdadera distribución subyacente de los datos, el modelo podría estar memorizando los datos de entrenamiento y perdiendo su capacidad de generalización.

Visualización de los datos reales y de los datos artificiales

En la Figura 26 se muestra una comparativa entre la distribución de los datos reales y los generados por el modelo de VAE tras 100 épocas de entrenamiento para la característica 2.

Al igual que en las observaciones anteriores con 10 épocas, se aprecia que el modelo sigue centrando su generación en torno al valor medio de la característica real. La curva de los datos generados muestra un pico pronunciado en el centro, lo cual indica que el modelo ha aprendido a reproducir el valor más común o promedio de la distribución, pero no su variabilidad completa.

Esta concentración en el punto medio sugiere que el VAE no está capturando adecuadamente la variabilidad intrínseca de los datos reales. En lugar de producir una gama amplia de valores que reflejen la distribución original, el modelo genera valores cercanos a una media central.

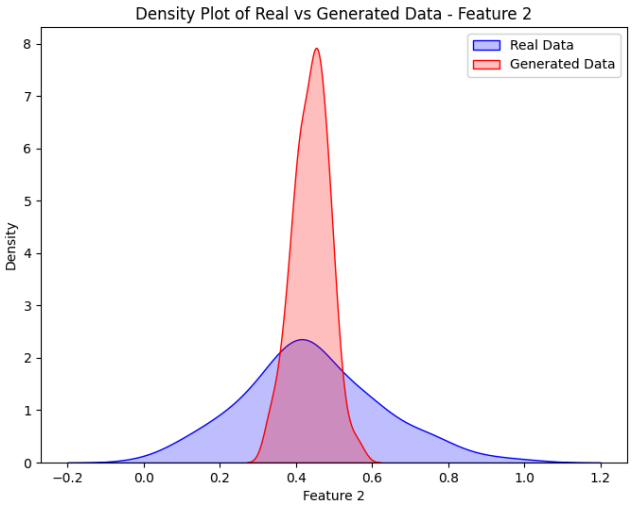
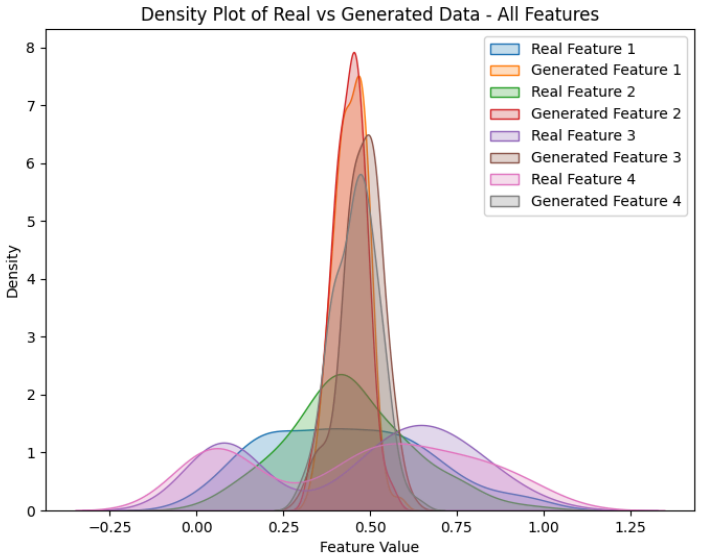


Figura 26: Gráficos de densidad de la característica 2 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

En la Figura 27 se muestra el gráfico de densidad combinado de todas las características de los datos reales. Se observa que el comportamiento es el mismo para todas las características.



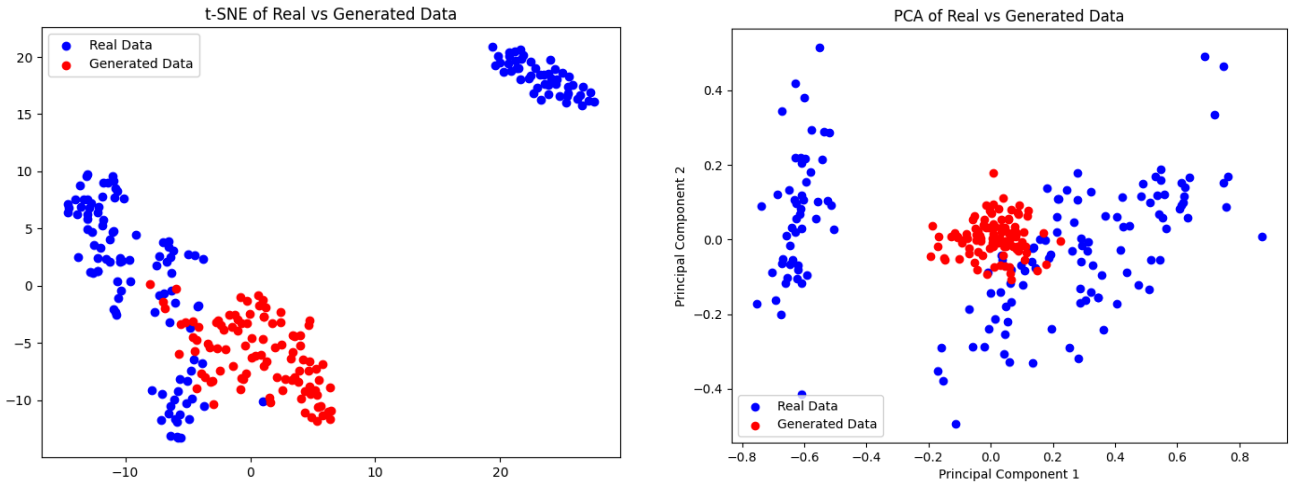
Figuras 27: Unión de los gráficos de densidad de todas las características.

Fuente: Elaboración propia usando la librería seaborn.

Los gráficos obtenidos del t-SNE y PCA (Figuras 28 y 29) revelan patrones similares a los observados previamente con 10 épocas.

En la visualización t-SNE, el modelo sigue evidenciando dificultades para diferenciar entre los dos grupos del conjunto de datos real. Se aprecia que las representaciones generadas se concentran alrededor de una de las agrupaciones, sin llegar a replicar la separación ni la densidad del segundo grupo, lo que indica una limitación en la capacidad del modelo de captar y reproducir la diversidad completa del conjunto de datos.

Por otro lado, el análisis mediante PCA muestra que los datos generados no presentan una variabilidad significativa y tienden a agruparse alrededor de un punto medio. Esta falta de dispersión sugiere que el modelo no ha aprendido a capturar la complejidad subyacente de los datos reales y podría estar sufriendo de sobreajuste, donde el aprendizaje se ha restringido al patrón más frecuente en el conjunto de entrenamiento, en detrimento de la captura de la variabilidad y riqueza del conjunto de datos en su totalidad.



Figuras 28 y 29: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn.

11.1.3. Test de Bit Diffusion

Ahora se examinará la arquitectura Bit Diffusion a través de pruebas con distintas épocas de entrenamiento. Los hiperparámetros seleccionados son:

- Dimensiones ocultas (*BITDIFF_HIDDEN_DIM*): 128
- Máximo nivel de ruido (*MAX_NOISE_LEVEL_BITDIFF*): 0.1

Generación de datos con 10 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 10

Pérdidas del modelo Bit Diffusion

En la Figura 30, se puede observar el *log* de las pérdidas del modelo *Bit Diffusion*, como en otros modelos, el bajo número de épocas no nos asegura la convergencia de las pérdidas. Esto implica que el modelo tiene posibilidad de mejora si aumentamos el número de épocas.

```
Epoch [1/10], Loss: 0.1245
Epoch [2/10], Loss: 0.0911
Epoch [3/10], Loss: 0.0434
Epoch [4/10], Loss: 0.0127
Epoch [5/10], Loss: 0.0111
Epoch [6/10], Loss: 0.0077
Epoch [7/10], Loss: 0.0089
Epoch [8/10], Loss: 0.0092
Epoch [9/10], Loss: 0.0103
Epoch [10/10], Loss: 0.0058
```

Figura 30: Pérdidas del modelo Bit Diffusion (10 épocas)

Fuente: Captura del log de las pérdidas de Bit Diffusion, elaboración propia

Prueba de Kolmogorov-Smirnov

Primera característica		Segunda característica		Tercera característica		Cuarta característica	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.27	0.0002	0.54	6.06e-17	0.4	3.35e-09	0.3	3.12e-05

Tabla 12: Resultados Kolmogorov-Smirnov para Bit Diffusion (10 épocas)

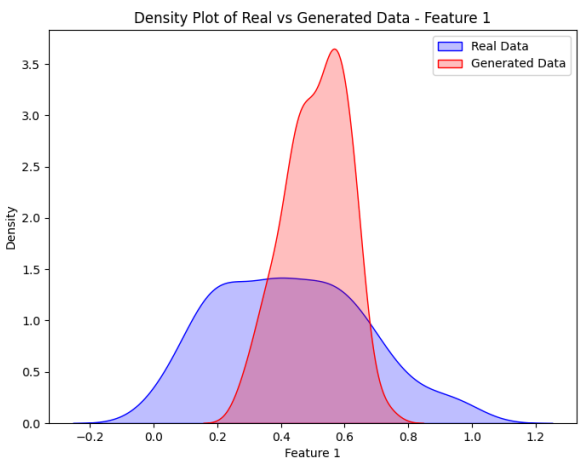
Fuente: Elaboración propia

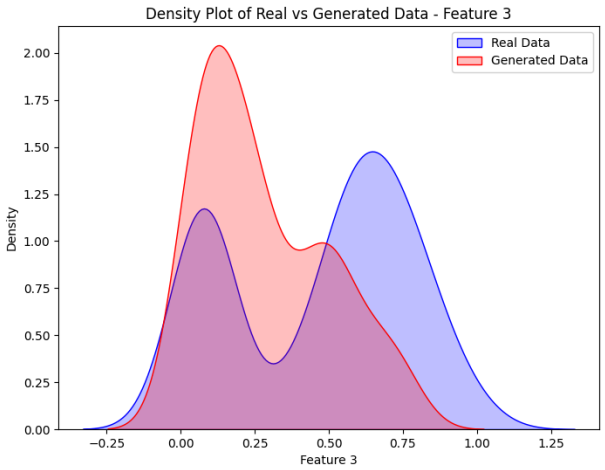
El estadístico de prueba para la primera característica muestra una diferencia notable con un valor *p* bajo, lo que indica que hay cierta divergencia entre los datos reales y generados. Esta diferencia se acentúa en la segunda característica, donde el estadístico alcanza un valor aún mayor y el valor *p* casi nulo, señalando una discrepancia considerable.

Para la tercera y cuarta características, los estadísticos reflejan desviaciones significativas con valores *p* que reafirman la existencia de diferencias entre las distribuciones comparadas. Estos hallazgos sugieren que, aunque el modelo es capaz de aprender patrones de los datos hasta cierto punto, aún no logra una imitación precisa en todas las dimensiones de los datos, esto es debido probablemente a la falta de épocas.

Visualización de los datos reales y de los datos artificiales

Seguidamente se proporcionarán análisis visuales de los datos generados por Bit Diffusion en 10 épocas.





Figuras 31 y 32: Gráficos de densidad de las características 1 y 3 de los datos reales y generados.

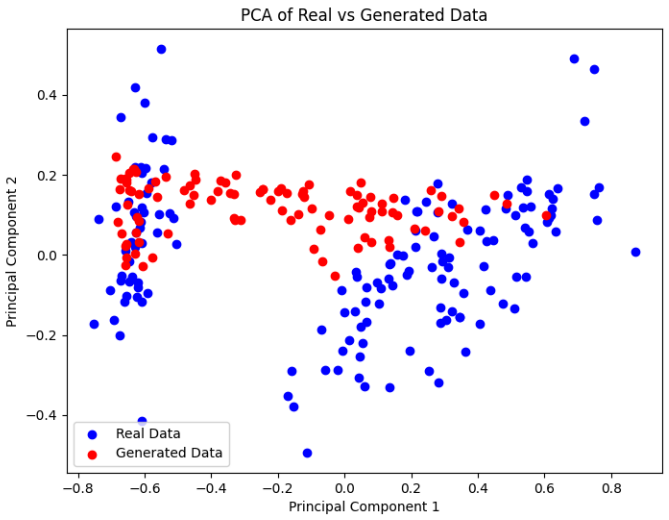
Fuente: Elaboración propia usando la librería seaborn.

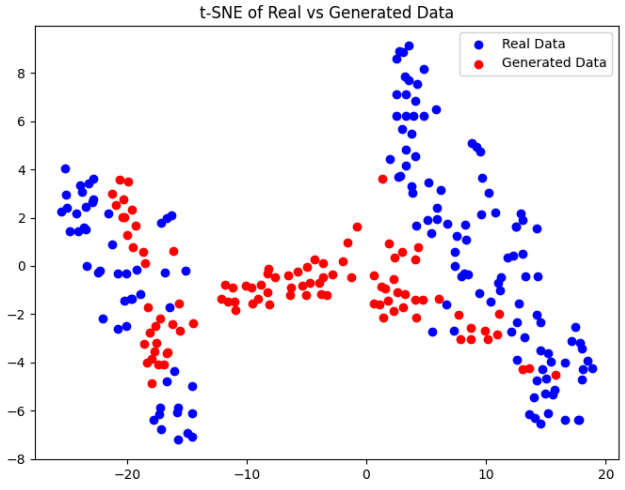
En el primer gráfico (Figura 31), se aprecia que el modelo identifica correctamente la moda de los datos, lo cual es un aspecto positivo ya que indica una comprensión parcial de la distribución subyacente. Sin embargo, la incapacidad para capturar la dispersión completa sugiere limitaciones en la representación de la variabilidad completa de los datos reales.

Por otro lado, el segundo gráfico (Figura 32) muestra una mejora significativa, ya que el modelo no solo detecta la existencia de dos picos en la distribución sino que también logra imitarlos en cierta medida. A pesar de esto, el modelo aún no alcanza la altura precisa de los picos observados en los datos reales.

En el gráfico de PCA (Figura 33), se pueden identificar claramente dos grupos de datos correspondientes a los datos reales, los cuales están bien definidos y separados. Sin embargo, cuando observamos los datos generados, se puede notar que se encuentran dispersos en ambos grupos, formando una especie de conexión entre ellos. Esta observación sugiere que los datos generados no logran captar completamente la variabilidad presente en los datos reales. En otras palabras, aunque los datos generados están presentes en ambos grupos y existe variabilidad entre ellos, no reflejan con precisión las características distintivas de cada uno de ellos.

Por otro lado, en la gráfica de t-SNE también se pueden distinguir claramente dos grupos correspondientes a los datos reales. Sin embargo, la representación de los datos generados en esta proyección da lugar a tres grupos, uno de los cuales se encuentra en el medio, actuando como un puente entre los grupos de datos reales. Esta observación refuerza la idea de que los datos generados no están completamente alineados con los patrones de variabilidad presentes en los datos reales, y en su lugar, crean una nueva agrupación que no se corresponde completamente con ninguno de los grupos reales.





Figuras 33 y 34: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

Generación de datos con 100 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 100

Pérdidas del modelo Bit Diffusion

Seguidamente, se obtienen las pérdidas del modelo (Figura 35) con 100 épocas. En estas sí que se puede observar una convergencia de valores y una reducción notable de las pérdidas.

```
Epoch [96/100], Loss: 0.0023
Epoch [97/100], Loss: 0.0013
Epoch [98/100], Loss: 0.0021
Epoch [99/100], Loss: 0.0024
Epoch [100/100], Loss: 0.0021
```

Figura 35: Pérdidas del modelo Bit Diffusion (100 épocas)

Fuente: Captura del log de las pérdidas de Bit Diffusion, elaboración propia

Prueba de Kolmogorov-Smirnov

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.07	0.85	0.18	0.03	0.29	5.1e-05	0.29	6.49e-05

Tabla 13: Resultados Kolmogorov-Smirnov para Bit Diffusion (100 épocas)

Fuente: Elaboración propia

Los resultados de la prueba de Kolmogorov-Smirnov muestran una mejora notable en comparación con los resultados obtenidos con solo 10 épocas. Destaca especialmente la primera característica, que ahora tiene un estadístico *D* de 0.07 y un valor *p* de 0.8542. Esto indica una similitud notablemente cercana entre los datos generados y los datos originales en esa característica particular.

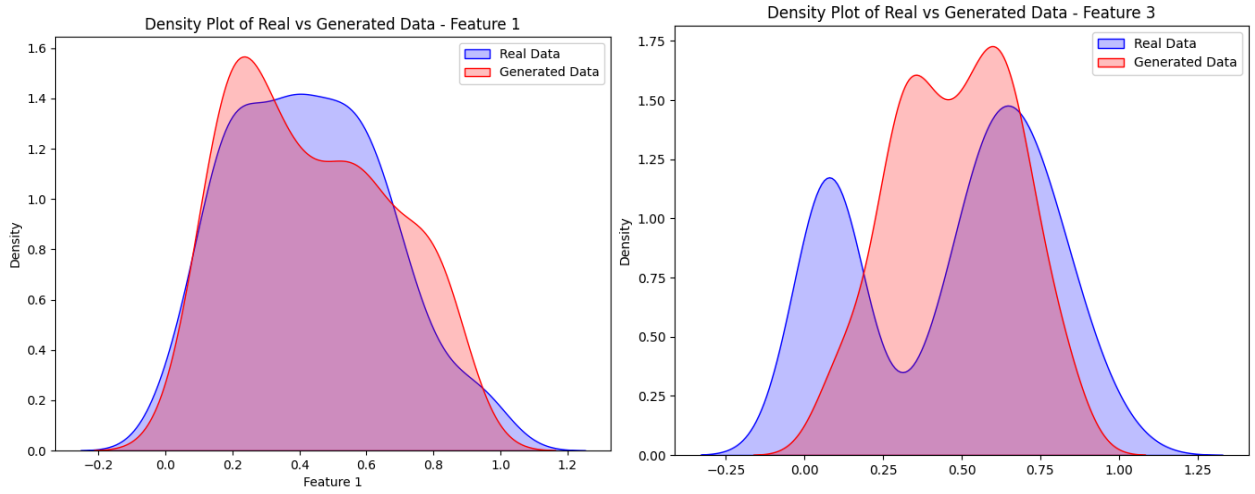
En el caso de la segunda característica, también se observa una mejora sustancial, con un estadístico *D* de 0.1833 y un valor *p* de 0.0323. Aunque aún existe una diferencia significativa en la distribución en comparación con los datos originales, la mejora en comparación con los resultados de 10 épocas es evidente. Esto sugiere un progreso en la capacidad del modelo para generar datos que se asemejen más a los datos reales en esta característica.

Las características 3 y 4 también muestran mejoras considerables, con estadísticos *D* de 0.2933 y 0.29, respectivamente, y valores *p* muy bajos. Aunque todavía existen diferencias significativas con respecto a los datos reales, los resultados son mejores que en 10 épocas.

Visualización de los datos reales y de los datos artificiales

En el primer gráfico de densidad (Figura 36), es evidente que estas dos distribuciones son muy similares, lo que indica que el modelo ha logrado captar con precisión la distribución de la característica 1. Esta similitud en las distribuciones sugiere que los datos generados se ajustan muy bien a los datos reales en esta característica específica, lo que es un resultado altamente satisfactorio.

En contraste, en el segundo gráfico que representa la característica 3, se puede apreciar que el modelo de bit difusión logra capturar la variabilidad de los datos, pero se enfrenta a una distribución compleja. Aunque el modelo es capaz de reflejar la variabilidad general de los datos, no logra replicar con precisión los dos picos exactos presentes en los datos reales. En su lugar, tiende a centrarse en valores intermedios entre ambos picos. Esto indica que, aunque el modelo es competente en capturar la variabilidad general de la característica 3, puede haber limitaciones para modelar distribuciones altamente complejas y bimodales.

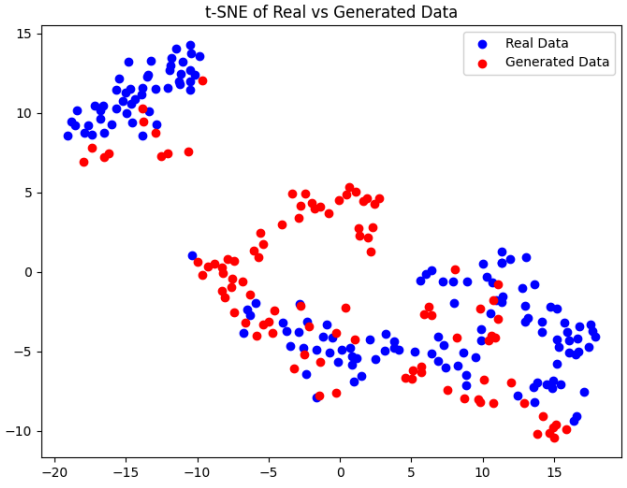


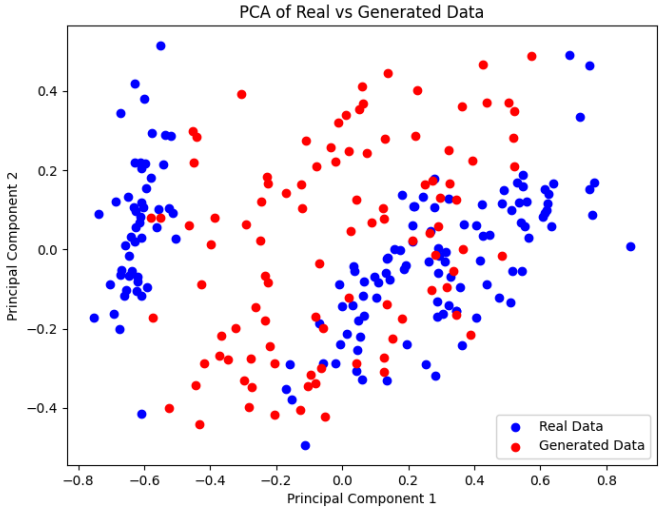
Figuras 36 y 37: Gráficos de densidad de las características 1 y 3 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

En el gráfico de t-SNE (Figura 38), se distinguen claramente dos grupos de datos reales, y es notable que los datos generados logran dividirse en dos grupos que guardan una gran similitud con los datos reales. Esta observación sugiere que el modelo ha sido efectivo al reproducir la estructura y la distribución de los datos originales. La capacidad de los datos generados para reflejar los dos grupos existentes en los datos reales es un resultado prometedor y demuestra que el modelo ha capturado de manera significativa la variabilidad presente en los datos.

Por otro lado, en el gráfico de PCA, donde los datos generados se encuentran dentro de los intervalos de los datos reales, se observa que los datos generados no logran separarse en dos grupos distintos, como ocurre con los datos reales. Aunque los datos generados aplican una variabilidad similar a la de los datos reales, la representación en PCA no permite una división clara en dos grupos como la que se observa en los datos reales. Esto podría indicar que, a pesar de replicar la variabilidad general, los datos generados no capturan completamente las diferencias específicas que existen entre los dos grupos de datos reales.





Figuras 38 y 39: Gráficos que muestran el PCA y t-SNE de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

11.1.4. Test de Data Augmentation

Ahora examinaremos la arquitectura del *Data Augmentation* a través de pruebas con distintas épocas de entrenamiento. Los hiperparámetros seleccionados son:

- Dimensiones ocultas (*DATA_AUGMENTATION_HIDDEN_DIM*): 128

Generación de datos con 10 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 10

Pérdidas del modelo Data Augmentation

Se puede observar en la Figura 40, como en los anteriores modelos, una clara disminución de las pérdidas pero sin saber ciertamente si ha logrado converger.

```
Epoch [1/10], Loss: 0.14840805903077126
Epoch [2/10], Loss: 0.13765376806259155
Epoch [3/10], Loss: 0.12869267538189888
Epoch [4/10], Loss: 0.11832832358777523
Epoch [5/10], Loss: 0.10712404549121857
Epoch [6/10], Loss: 0.09678632207214832
Epoch [7/10], Loss: 0.08434121683239937
Epoch [8/10], Loss: 0.06877422146499157
Epoch [9/10], Loss: 0.05865002889186144
Epoch [10/10], Loss: 0.0451510539278388
```

Figura 40: Pérdidas del modelo Data Augmentation (10 épocas)

Fuente: Captura del log de las pérdidas de Data Augmentation, elaboración propia

Prueba de Kolmogorov-Smirnov

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.37	6.45e-08	0.27	0.0002	0.33	2.25e-06	0.33	2.25e-06

Tabla 14: Resultados Kolmogorov-Smirnov para Data Augmentation (10 épocas)

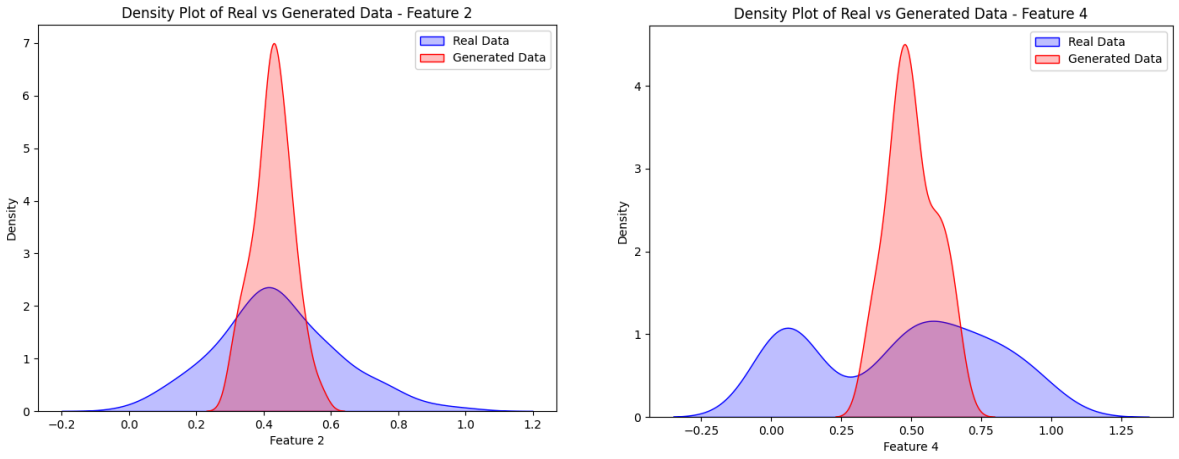
Fuente: Elaboración propia

Los resultados de la prueba de Kolmogorov-Smirnov para estas cuatro características muestran que en todas ellas existe una diferencia significativa entre las distribuciones de datos observados y las distribuciones teóricas de referencia. Los valores bajos de los estadísticos *D* y los valores p indican que estas características no siguen la distribución datos reales y que las diferencias son estadísticamente significativas como sucedía en el modelo VAE.

Visualización de los datos reales y de los datos artificiales

En los gráficos de densidad, se puede observar que las funciones de distribución detectan claramente la moda de los datos. Esto significa que gran parte de la generación de datos se concentra alrededor de estas modas, lo que se evidencia en la Figura 41. Este comportamiento es similar al que se observaba en el modelo VAE.

Sin embargo, al comparar los datos generados con solo 10 épocas en la Figura 42, notamos que el modelo encuentra dificultades para interpretar una distribución bimodal. Aunque es capaz de capturar un punto intermedio entre las dos modas presentes en los datos reales, le resulta complicado replicar de manera precisa una distribución con dos modas distintas con tan pocas épocas de entrenamiento. Esto subraya la importancia del número de épocas en el proceso de entrenamiento y cómo una mayor cantidad de épocas podría permitir al modelo reflejar con mayor precisión distribuciones más complejas y bimodales.

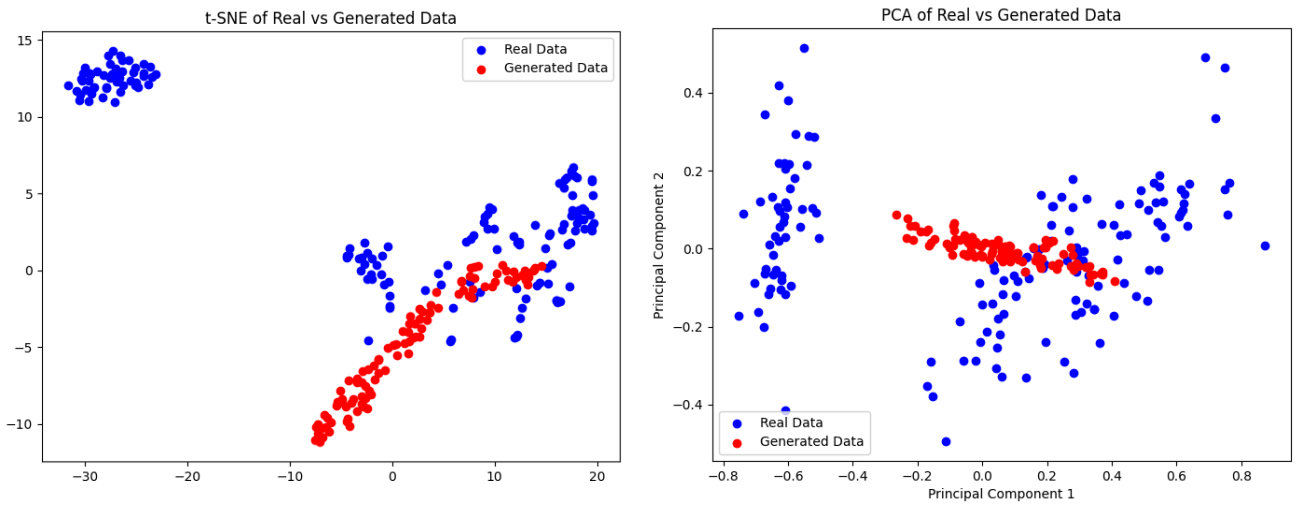


Figuras 41 y 42: Gráficos de densidad de las características 2 y 4 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

En la Figura 43, que representa el gráfico de t-SNE, se observa que los datos generados se dividen en dos grandes grupos, al igual que los datos reales. Sin embargo, el modelo de generación de datos solo parece ser capaz de interpretar y reflejar uno de estos grupos de manera adecuada. Aunque algunos puntos generados se encuentran dentro de los intervalos de los datos reales, la mayoría se desvía de manera significativa. Esto sugiere que el modelo puede estar experimentando dificultades para reproducir la variabilidad de ambos grupos de manera equitativa y podría estar enfocándose excesivamente en uno de ellos. Este comportamiento podría ser indicativo de un posible sobreajuste del modelo.

En el gráfico de PCA de la Figura 44, se observa una situación similar en términos de la división de los datos en dos grupos. Sin embargo, en este caso, los puntos generados muestran menos variabilidad y se agrupan principalmente en uno de los dos grupos, lo que sugiere una menor capacidad del modelo para capturar la variabilidad en comparación con el gráfico de t-SNE. Esta falta de variabilidad y la agrupación en un solo grupo pueden ser indicios de un posible sobreajuste del modelo a los datos.



Figuras 43 y 44: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

Generación de datos con 100 épocas

A continuación se ajustará el hiperparámetro de las épocas:

- Número de épocas (*EPOCHS*): 100

Pérdidas del modelo Data Augmentation

En la figura 45, se observa la convergencia en las pérdidas del *Data Augmentation*, sin embargo, no se observa una clara mejora con respecto al modelo entrenado con 10 épocas.

```
Epoch [94/100], Loss: 0.009146619122475386
Epoch [95/100], Loss: 0.006219145201612264
Epoch [96/100], Loss: 0.005114877945743501
Epoch [97/100], Loss: 0.005894306697882712
Epoch [98/100], Loss: 0.007904990576207638
Epoch [99/100], Loss: 0.0074084491934627295
Epoch [100/100], Loss: 0.005701722810044885
```

Figura 45: Pérdidas del modelo Data Augmentation (100 épocas)

Fuente: Captura del log de las pérdidas de Data Augmentation, elaboración propia

Prueba de Kolmogorov-Smirnov

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.37	6.45e-08	0.27	0.0002	0.33	2.25e-06	0.33	2.25e-06

Tabla 15: Resultados Kolmogorov-Smirnov para Data Augmentation (100 épocas)

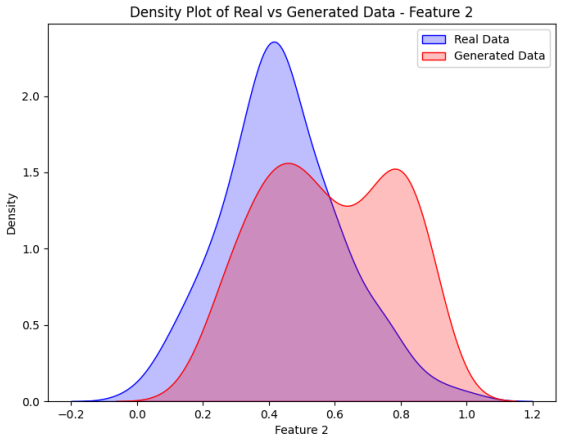
Fuente: Elaboración propia

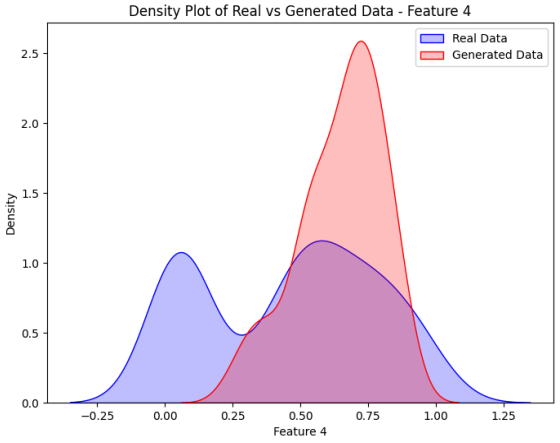
Los resultados de la prueba de *Kolmogorov-Smirnov* con 100 épocas de data augmentation muestran una mejora en comparación con los resultados obtenidos con solo 10 épocas. Sin embargo, aún persisten diferencias significativas entre los datos generados y los datos reales en todas las características. En particular, la primera característica muestra una diferencia aún más marcada, lo que sugiere que el modelo no puede replicar con precisión la variabilidad y distribución de los datos. Esto puede deberse a que el data augmentation, aunque mejora, sigue siendo una arquitectura simple que no captura completamente la complejidad de los datos reales en términos de variabilidad y distribución.

Visualización de los datos reales y de los datos artificiales

En la Figura 46, observamos que los datos generados producen una distribución bimodal, lo cual es inusual ya que los datos reales tienen una única moda. Esta discrepancia podría deberse a que las otras características en el conjunto de datos son bimodales, y la arquitectura simple utilizada para la generación de datos no es capaz de capturar la variabilidad de manera independiente. La influencia de las otras características bimodales podría estar afectando la distribución de esta característica en particular, generando una apariencia bimodal en lugar de una sola moda como se encuentra en los datos reales.

Por otro lado, en la Figura 47, la distribución generada no se parece en nada a la distribución bimodal de los datos reales. En lugar de replicar adecuadamente la bimodalidad presente en los datos reales, la generación de datos solo parece ser capaz de detectar y reflejar una de las modas, lo que resulta en una distribución unimodal que no se ajusta a la complejidad de los datos reales. Esto sugiere limitaciones en la capacidad del modelo para capturar y reproducir distribuciones de datos bimodales de manera precisa.





Figuras 46 y 47: Gráficos de densidad de las características 2 y 4 de los datos reales y generados.

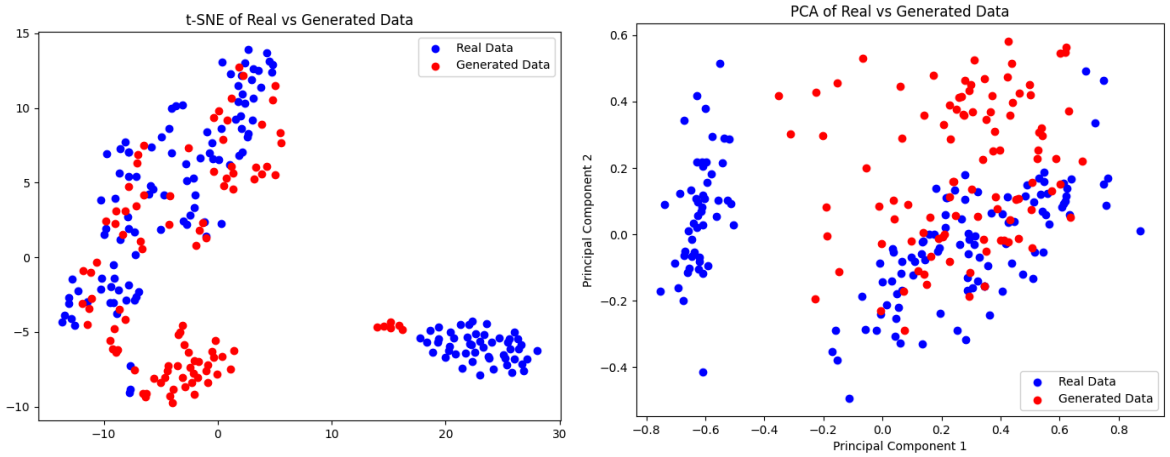
Fuente: Elaboración propia usando la librería seaborn.

En la Figura 48, representada por el gráfico de t-SNE, observamos que los datos reales se dividen en dos grupos claramente diferenciados. Los datos generados muestran una mejora en comparación con los resultados anteriores, ya que ahora son capaces de generar aproximadamente un 5% de los datos en el otro grupo. Aunque esta mejora es evidente, sigue siendo una diferenciación relativamente baja en comparación con la capacidad del modelo para generar datos en el grupo principal.

Esto sugiere que, si bien se ha logrado algún progreso en la capacidad de distinguir ambos grupos, todavía persisten desafíos en la generación de datos que se ajusten de manera precisa a la distribución y variabilidad de ambos grupos.

Por otro lado, en la Figura 49, que representa el gráfico de PCA, notamos que los datos reales se dividen claramente en dos grupos distintos. Los datos generados han aumentado su variabilidad y se asemejan más a la variabilidad presente en los datos reales. Sin embargo, el modelo todavía enfrenta dificultades para diferenciar adecuadamente ambos grupos, ya que los puntos generados siguen agrupándose principalmente en uno de los grupos.

A pesar de la mejora en la variabilidad, el modelo aún no puede reflejar completamente la complejidad de los datos reales en términos de separación de grupos.



Figuras 48 y 49: Gráficos que muestran el t-SNE y PCA de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

11.2. Conjuntos de datos grandes

En la siguiente sección, se explorará cómo se desempeñan los modelos en relación a conjuntos de datos de gran tamaño. Esta aproximación brinda la oportunidad de evaluar la habilidad de los modelos para capturar la diversidad de los datos en situaciones en las que se dispone de una gran cantidad de información. Este análisis es esencial para determinar su eficacia en entornos donde se requiere la identificación de las características fundamentales.

Para estos tests, ha sido configurado el siguiente hiperparámetro:

- Número de muestras generadas (*NUM_SAMPLES*): 5000

El *dataset* escogido para este estudio comprende un total de ocho variables, de las cuales algunas son categóricas y han sido codificadas mediante la técnica de *One Hot Encoding*. Para fines experimentales, se ha procedido a la reducción estratégica del número de variables a cuatro, alterando deliberadamente los datos para crear distribuciones atípicas. Este enfoque tiene como objetivo evaluar el rendimiento del modelo frente a escenarios complejos, caracterizados por relaciones inusuales entre los datos, y determinar su capacidad para manejar situaciones de alta complejidad.

Este ajuste se realiza con el objetivo de mantener una proporción cercana a la del conjunto de datos original, el cual consta de aproximadamente 15000 muestras. Esto proporciona una comparación equitativa entre los datos generados y los datos reales, permitiendo una evaluación más precisa de la eficiencia de los modelos en la generación de datos artificiales bajo restricciones de tamaño.

Las épocas para las pruebas se seleccionarán según los valores en los cuales los modelos demuestran un rendimiento óptimo tras las pruebas iniciales con conjuntos de datos reducidos. Dado que se ha observado que se obtienen los mejores resultados con 100 épocas, se procederá a ejecutar las pruebas directamente utilizando esta cantidad de épocas que es cuando tienden a converger las pérdidas de los modelos, también se activará la parada temprana por si convergen antes de las 100, por lo tanto:

- Número de épocas (*EPOCHS*): 100
- Parada temprana (*EARLY_STOPPING):* True

11.2.1. Test de GAN

Ahora se explorará el desempeño del modelo GAN en un escenario con un conjunto de datos considerablemente grande. Se mantendrán los mismos hiperparámetros que han sido utilizados previamente en conjuntos de datos más pequeños, ya que estos parámetros han demostrado proporcionar resultados óptimos.

Prueba de Kolmogorov-Smirnov

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.14	1.5e-61	0.16	7.5e-83	0.1	3.8e-30	0.17	2.4e-87

Tabla 16: Resultados Kolmogorov-Smirnov para GAN (conjunto grande)
Fuente: Elaboración propia

Los resultados de la Prueba de Kolmogorov-Smirnov presentados para el modelo GAN en la Tabla 16 son altamente indicativos de una similitud significativa entre los datos generados y los datos reales. Los valores *p* obtenidos son extremadamente bajos, lo que matemáticamente rechaza la hipótesis nula de que las dos distribuciones son iguales con un alto grado de confianza. Paradójicamente, en el contexto de las pruebas de generación de datos, un valor p extremadamente bajo puede indicar que el modelo ha aprendido a imitar con precisión las distribuciones de los datos reales.

El primer valor de estadística de 0.145, aunque representa una pequeña distancia entre las distribuciones comparadas, junto con un valor *p* cercano a cero, sugiere una excelente convergencia entre los datos reales y los generados. Similarmente, los otros tres valores de la estadística *D*, aunque ligeramente mayores, están acompañados de valores *p* aún más bajos, reforzando la evidencia de que la generación de datos ha sido altamente efectiva.

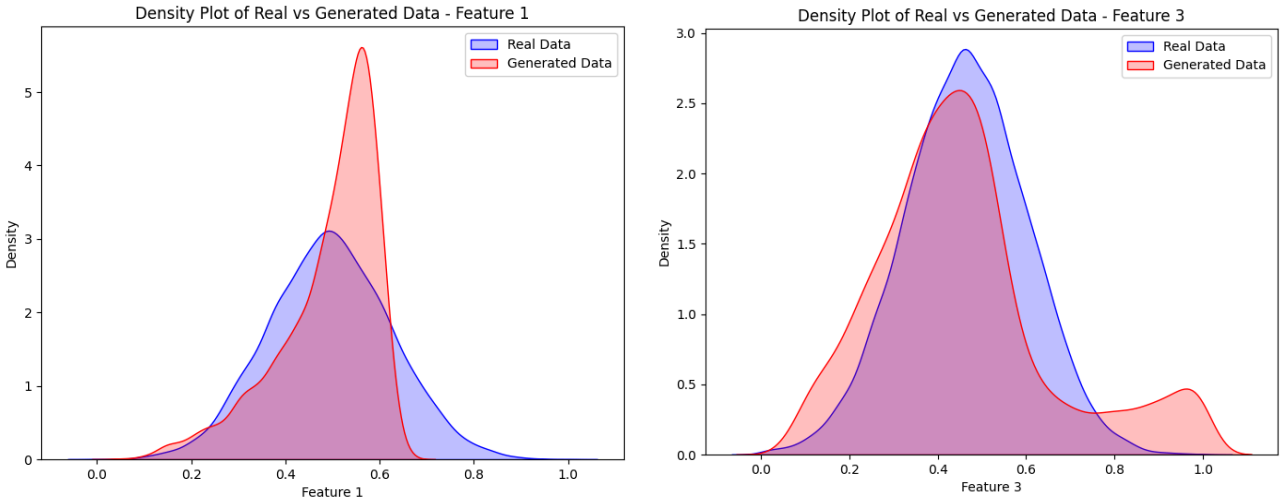
La consistencia en la proximidad de las distribuciones a través de las cuatro características evaluadas indica que el modelo GAN no sólo ha aprendido patrones generales, sino que ha capturado la esencia de los datos en múltiples dimensiones. Esto es un testimonio de la capacidad del modelo para generar datos sintéticos que podrían ser indistinguibles de los reales en muchas aplicaciones prácticas.

Visualización de los datos reales y de los datos artificiales

En la Figura 50, se percibe que el modelo logra identificar la moda de los datos con un grado notable de precisión. Sin embargo, se observa una sobreestimación en la concentración de las observaciones alrededor del pico central. Este fenómeno se manifiesta en un pico más pronunciado para los datos generados en comparación con los reales. A pesar de la discrepancia en altura, la consistencia en la forma de la distribución revela una interpretación adecuada de la variabilidad intrínseca de los datos por parte del modelo.

Por otro lado, la Figura 51 muestra una conformidad general en las distribuciones de ambos conjuntos de datos, con la excepción de la aparición de valores atípicos en la cola derecha de los datos generados. Estos valores atípicos dan lugar a una distribución bimodal, que puede interpretarse como una exploración por parte del modelo de segmentos del espacio de datos menos representados en el conjunto real. Este patrón bimodal podría sugerir la presencia de una diversidad latente no plenamente capturada en los datos originales o una inclinación del modelo hacia la generación de modalidades alternativas.

A pesar de esto, las distribuciones de las características de los datos generados parecen ser muy similares a las distribuciones de los datos reales.



Figuras 50 y 51: Gráficos de densidad de las características 1 y 3 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

Los conjuntos de datos reales exhiben una estructura de agrupación distintiva que el modelo ha identificado y mimetizado de manera efectiva en las regiones centrales del gráfico, esto se puede observar en la Figura 52.

La concordancia observada entre los datos reales y generados en las zonas de agrupación principal indica que el modelo ha logrado aprender las características subyacentes del conjunto de datos original con un alto grado de fidelidad.

La efectividad del modelo para capturar la variabilidad y las relaciones en los datos queda demostrada en el gráfico que representa t-SNE, lo que sugiere una generación de datos sintéticos competente y una comprensión estadística del conjunto de datos original. Estos resultados son prometedores y resaltan la potencialidad del modelo generativo como herramienta para la creación de conjuntos de datos sintéticos que puedan ser utilizados en diversas aplicaciones analíticas.

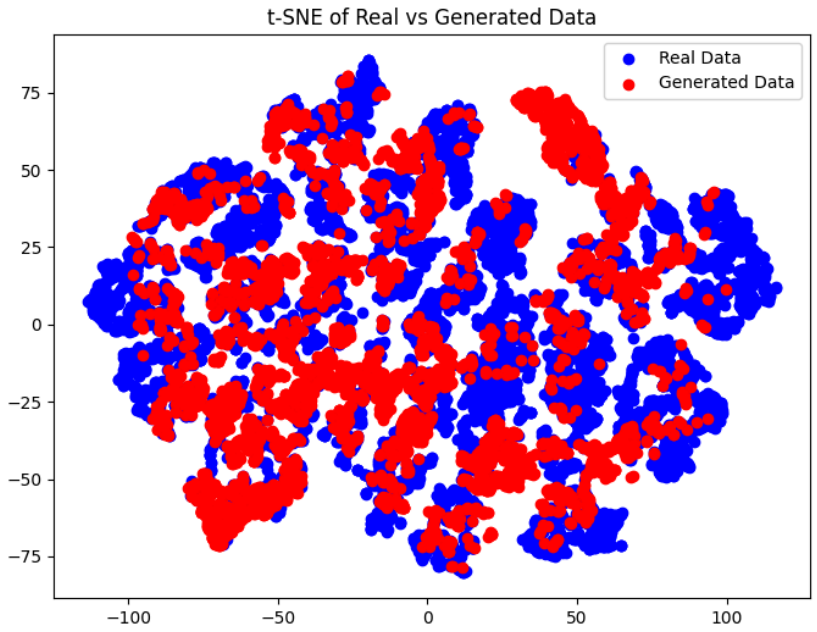


Figura 52: Gráfico que muestra el t-SNE de GAN de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

11.2.2. Test de VAE

Seguidamente se explorará el desempeño del modelo VAE en un escenario con un conjunto de datos grande. Se mantendrán los mismos hiperparámetros que han sido utilizados previamente en conjuntos de datos más pequeños, ya que estos parámetros han demostrado proporcionar resultados óptimos.

Prueba de Kolmogorov-Smirnov

Primera característica		Segunda característica		Tercera característica		Cuarta característica	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.39	4.48e-128	0.39	2.07e-126	0.43	3.5e-157	0.44	1.42e-161

Tabla 17: Resultados Kolmogorov-Smirnov para VAE (conjunto grande)
Fuente: Elaboración propia

Los resultados obtenidos de esta prueba para VAE presentan valores *D* significativamente altos en las cuatro características evaluadas, lo que sugiere una discrepancia notoria entre las distribuciones de los datos reales y los generados por el modelo.

Los valores de *p* asociados son extremadamente bajos, cayendo muy por debajo de cualquier umbral convencional de significancia estadística (como $p < 0.05$ o incluso $p < 0.01$). Esto indica que es altamente improbable que las diferencias observadas entre las distribuciones sean el resultado del azar; más bien, apuntan a diferencias sistemáticas en cómo el VAE está modelando los datos.

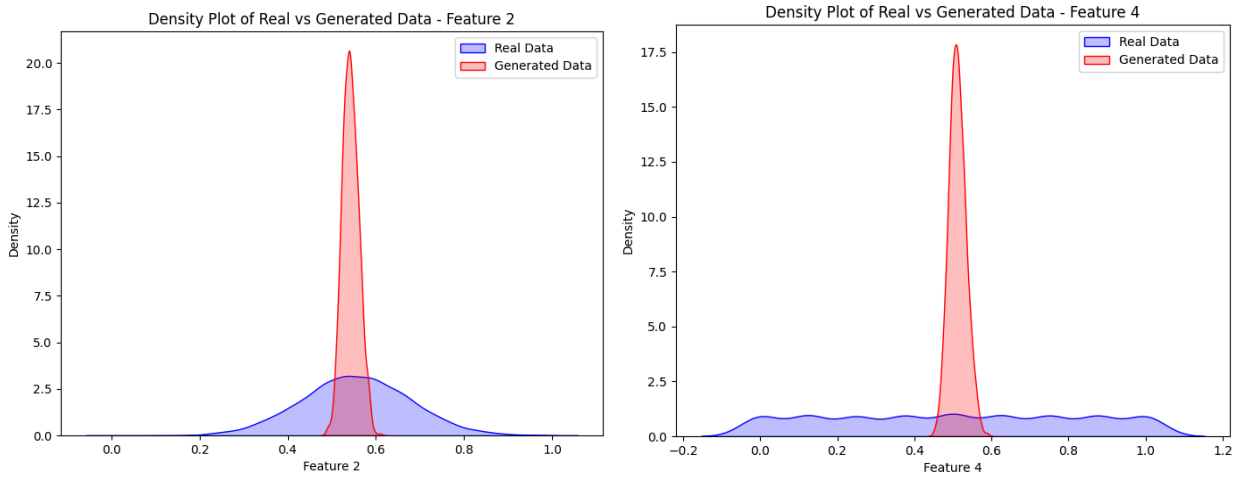
En particular, estos resultados sugieren que el VAE no está capturando adecuadamente la complejidad y variabilidad subyacente de los datos reales. La incapacidad del VAE para hacerlo, como se refleja en estos resultados, puede deberse a limitaciones en su estructura o en el proceso de entrenamiento que le impiden aprender con precisión la estructura de los datos de entrada.

Estos hallazgos sugieren que, aunque el VAE puede generar datos que parecen válidos a simple vista, si se realiza una inspección más profunda, se revela que los datos sintéticos no reflejan la variabilidad y complejidad de los reales.

Visualización de los datos reales y de los datos artificiales

En los gráficos de densidad de las Figuras 53 y 54, se puede observar que las funciones de distribución detectan claramente la moda de los datos. Esto significa que gran parte de la generación de datos se concentra alrededor de estas modas como sucedía en VAE cuando el conjunto de datos era pequeño

Esto quiere decir que VAE es incapaz de aprender distribuciones de datos complejas, ya que prioriza la reconstrucción de los datos antes que la variabilidad, como ya pasaba anteriormente.



Figuras 53 y 54: Gráficos de densidad de las características 2 y 4 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

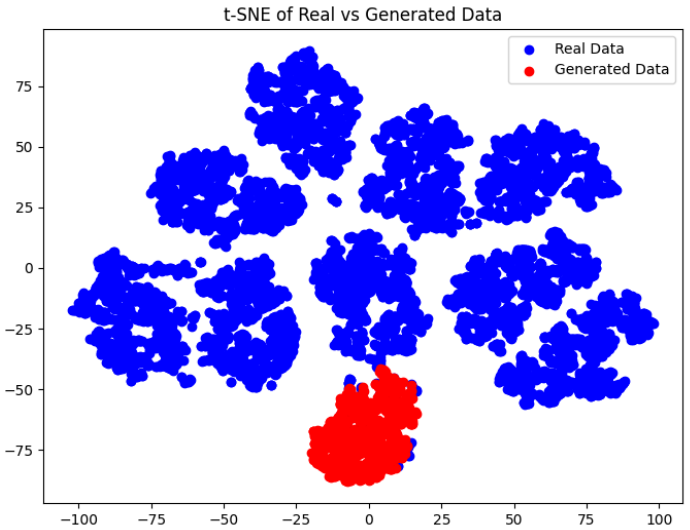


Figura 55: Gráfico que muestra el t-SNE de VAE de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

Por otro lado, en el t-SNE de la Figura 55, se observa que VAE sólo reconoce un conjunto de los 10 que se nos presentan en los datos. El modelo es incapaz de reconocer relaciones complejas entre datos pero sí es eficiente reconstruyendo conjuntos concretos de los datos reales.

11.2.3. Test de Bit Diffusion

A continuación se explorará el desempeño del modelo *Bit Diffusion* en un escenario con un conjunto de datos considerable. Se mantendrán los mismos hiperparámetros que han sido utilizados previamente en conjuntos de datos más pequeños, ya que estos parámetros han demostrado proporcionar resultados óptimos.

Prueba de Kolmogorov-Smirnov

Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.15	2.86e-106	0.22	9.19e-226	0.21	2.04e-195	0.11	1.17e-53

Tabla 18: Resultados Kolmogorov-Smirnov para Bit Diffusion (conjunto grande)
Fuente: Elaboración propia

Las estadísticas de prueba *D* varían entre 0.1106 y 0.2268, lo que indica diferencias entre las distribuciones de los datos reales y los generados. Sin embargo, en comparación con los resultados previos del modelo VAE, estas estadísticas son generalmente más bajas, lo que sugiere una mejor correspondencia entre las distribuciones de los datos reales y generados.

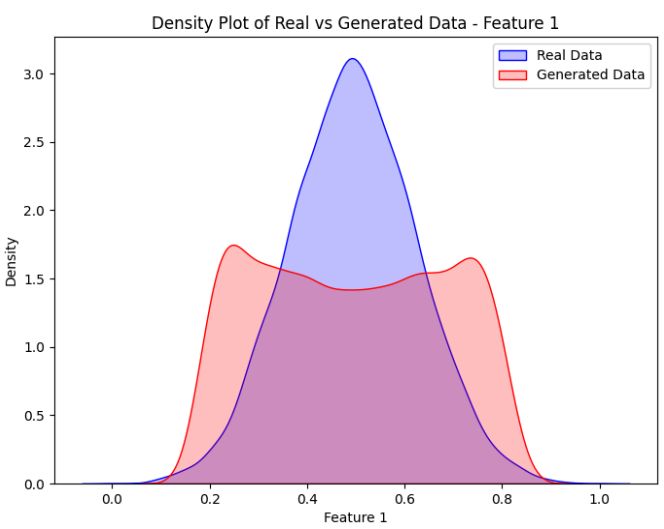
Los valores de *p* asociados con estas estadísticas son extremadamente bajos, lo que confirma la significancia estadística de los resultados. A pesar de ello, la magnitud más baja de las estadísticas *D* en comparación con los resultados del VAE implica que el modelo de *Bit Diffusion* está capturando la variabilidad de los datos con mayor fidelidad.

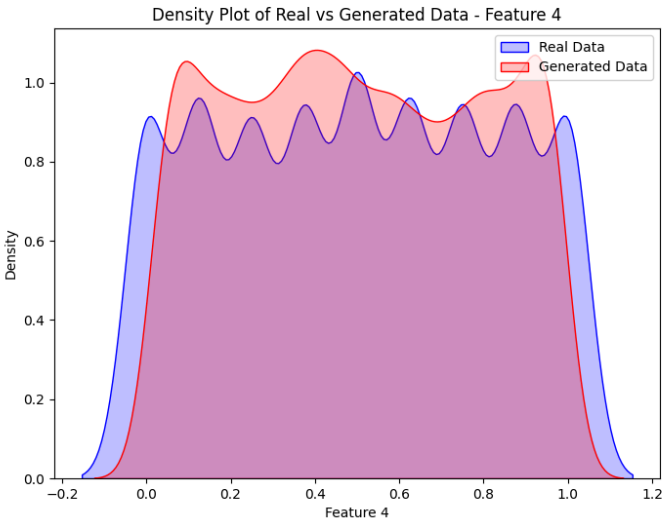
La menor estadística *D* para la característica 4, con un valor de *p* notablemente más alto que para las otras características, sugiere que para esta dimensión en particular, el modelo de Bit Diffusion ha logrado una simulación más precisa de la distribución real de los datos. Esto es un indicativo de que el modelo es capaz de aprender y reproducir eficazmente las distribuciones subyacentes en ciertas dimensiones del espacio de datos.

Visualización de los datos reales y de los datos artificiales

En la Figura 56, correspondiente a la característica 1, se observa que la densidad de los datos generados posee un pico más pronunciado y estrecho en comparación con los datos reales, que presentan una distribución más aplanada y extendida. A pesar de que ambos picos están alineados alrededor del mismo valor, la discrepancia en la forma de las distribuciones sugiere que el modelo generativo puede estar sobreajustando alrededor de la moda de los datos reales y no captura adecuadamente la variabilidad en las colas de la distribución.

Por otro lado, en la Figura 57, correspondiente a la característica 4, muestra un patrón distinto. Aquí, la distribución de los datos generados sigue de cerca la forma de la distribución real, con una pequeña divergencia en las colas. Los datos generados reflejan con mayor fidelidad la variabilidad de los datos reales, aunque todavía se percibe una ligera sobreestimación de la densidad en las regiones de menor frecuencia.





Figuras 56 y 57: Gráficos de densidad de las características 1 y 4 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

En la Figura 58, se nota que los puntos rojos están dispersos de manera similar a los puntos azules, lo que indica que el modelo generativo ha aprendido con éxito las relaciones y agrupaciones complejas presentes en los datos reales. Las áreas donde los puntos rojos y azules se intercalan sugieren que el modelo es capaz de simular las características subyacentes de los datos reales, replicando las agrupaciones sin aislarse en regiones específicas del espacio proyectado.

Este nivel de mezcla y solapamiento es un indicador positivo de que el modelo generativo no solo ha capturado la variabilidad de los datos reales sino que también ha aprendido las complejas relaciones multidimensionales. En consecuencia, el modelo demuestra ser una herramienta eficaz en la generación de nuevos datos que mantienen las propiedades estadísticas del conjunto original, lo que es un aspecto crucial en la validación de modelos generativos para la creación de datos sintéticos.

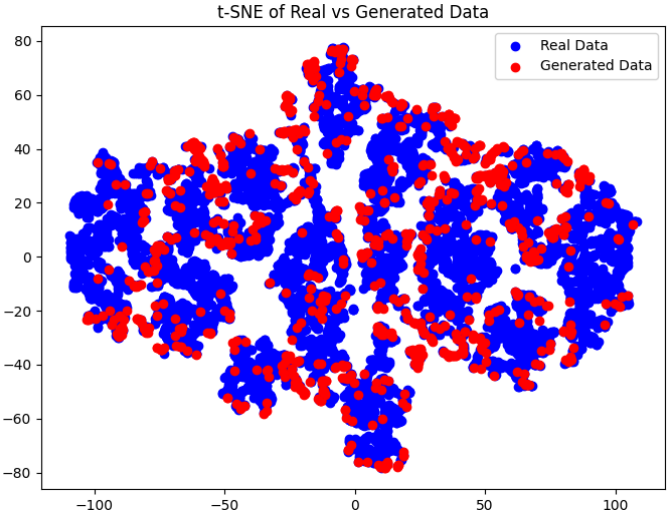


Figura 58: Gráfico que muestra el t-SNE de Bit Diffusion de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

11.2.4. Test de Data Augmentation

Finalmente se explorará el desempeño del modelo *Data Augmentation* en un escenario con un conjunto de datos considerable. Se mantendrán los mismos hiperparámetros que han sido utilizados previamente en conjuntos de datos más pequeños, ya que estos parámetros han demostrado proporcionar resultados óptimos.

Prueba de Kolmogorov-Smirnov

Primera característica		Segunda característica		Tercera característica		Cuarta característica	
D	***p***	***D***	***p***	***D***	***p***	***D***	***p***
0.22	5.04e-145	0.17	4.89e-84	0.24	3.43e-169	0.26	5.02e-121

Tabla 19: Resultados Kolmogorov-Smirnov para Data Augmentation (conjunto grande)

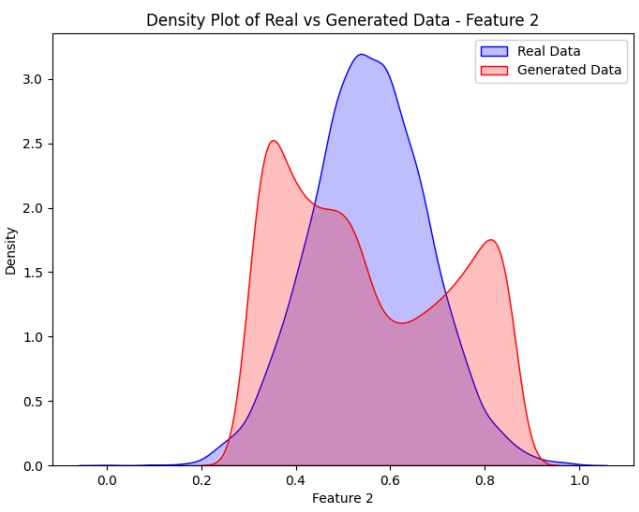
Fuente: Elaboración propia

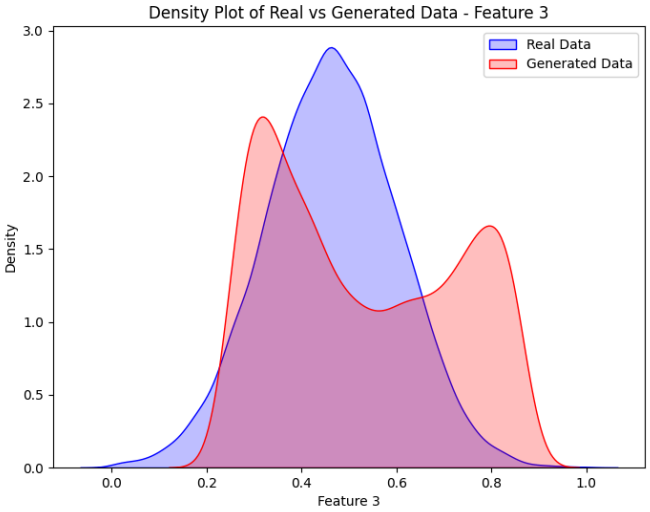
Los resultados muestran que hay diferencias significativas entre las distribuciones de datos reales y generados, con valores estadísticos que varían de 0.1695 a 0.2402 y *p* extremadamente bajos, pasa algo muy similar a lo que pasaba con VAE. Esto indica que, aunque el modelo logra cierta aproximación a los datos reales, no consigue replicar con exactitud sus distribuciones.

Visualización de los datos reales y de los datos artificiales

Las Figuras 59 y 60 reflejan una tendencia similar a la observada en el modelo VAE: se detecta una aproximación en la captación de la moda de los datos reales, pero no se logra una representación precisa de la distribución completa.

Específicamente, el modelo *Data Augmentation*, aunque es una técnica más sencilla, muestra limitaciones en la captación de la variabilidad y complejidad inherente a los datos reales. A pesar de ser entrenado en conjuntos de datos grandes, el modelo tiende a generar datos centrados alrededor de puntos pico más que en reproducir la gama completa de variabilidad y distribución, lo que sugiere una necesidad de métodos más avanzados o complejos para lograr una representación más fidedigna de los datos originales.





Figuras 59 y 60: Gráficos de densidad de las características 2 y 4 de los datos reales y generados.

Fuente: Elaboración propia usando la librería seaborn.

Al observar la disposición de los puntos, es notable que, aunque el modelo ha logrado identificar y generar puntos que se alinean en gran medida con los datos reales, hay una falta de distinción en lo que parece ser dos conjuntos o agrupaciones distintas en los datos reales.

Los datos generados parecen concentrarse en una sola agrupación más densa, sin reconocer o reproducir la separación entre los diversos conjuntos. Este comportamiento podría sugerir una limitación en la capacidad del modelo para captar la totalidad de la estructura de los datos, enfocándose en la tendencia predominante y omitiendo la presencia de posibles subgrupos o características distintivas en los datos.

Este fenómeno podría ser indicativo de la necesidad de revisar la complejidad del modelo para capturar con mayor precisión la diversidad inherente a los datos originales, como ya ha sido mencionado, *Data Augmentation* es uno de los modelos más sencillos existente y, por lo tanto, su potencial para capturar y aprender datos con distribuciones complejas es insuficiente.

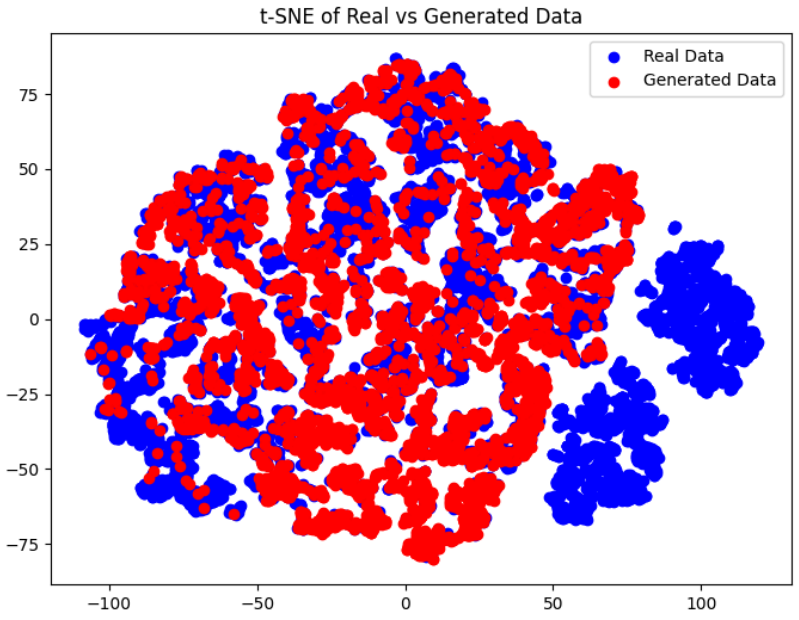


Figura 61: Gráfico que muestra el t-SNE de Data Augmentation de los datos reales comparándolos con los generados.

Fuente: Elaboración propia usando la librería seaborn

11.3. Conclusiones de los resultados de los modelos

La evaluación de los distintos modelos implementados en este proyecto ofrece una visión comprensiva sobre sus fortalezas y limitaciones al enfrentarse a la tarea de generar datos artificiales tabulares. En la tabla 20 podemos observar una tabla resumen de los resultados de la prueba de Kolmogorov-Smirnov.

	Primera característica		**Segunda característica**		**Tercera característica**		**Cuarta característica**	
Modelos	***D***	***p***	***D***	***p***	***D***	***p***	***D***	***p***
GAN	0.14	1.5e-61	0.16	7.5e-83	0.1	3.8e-30	0.17	2.4e-87
VAE	0.39	4.48e-128	0.39	2.07e-126	0.43	3.5e-157	0.44	1.42e-161
Bit Diffusion	0.15	2.86e-106	0.22	9.19e-226	0.21	2.04e-195	0.11	1.17e-53
Data Augmentation	0.22	5.04e-145	0.17	4.89e-84	0.24	3.43e-169	0.26	5.02e-121

Tabla 20: Resumen de los mejores resultados Kolmogorov-Smirnov
Fuente: Elaboración propia

El *Variational Autoencoder* (VAE) ha demostrado ser competente en la reconstrucción de datos a partir de sus representaciones latentes. Sin embargo, ha mostrado limitaciones para aprender y reproducir la variabilidad y las relaciones complejas entre los datos. Su tendencia a centrarse en el promedio de las distribuciones hace que sea más adecuado para aplicaciones donde la fidelidad en la reconstrucción es más crítica que la captura de la diversidad de datos.

El modelo llamado *Data Augmentation* que utiliza técnicas de *denoising autoencoders*, con su simplicidad inherente, se revela como una herramienta eficaz para la rápida generación de datos no complejos. Aunque no es el más adecuado para capturar las sutilezas de los conjuntos de datos con estructuras intrincadas, su facilidad de uso y eficiencia lo convierten en una opción valiosa para escenarios donde la complejidad de los datos no es una preocupación principal.

Bit Diffusion emerge como una solución robusta para capturar la complejidad y la variabilidad de los datos. Aunque no es tan eficiente en la imitación de la densidad de población de los datos reales, su habilidad para generar datos que preservan las relaciones complejas lo hace idóneo para situaciones donde el volumen y la diversidad de datos generados son de mayor importancia que la proporción exacta de subgrupos dentro del conjunto de datos.

Por último, las *Generative Adversarial Networks* (GAN) se presentan como una opción sólida para imitar las distribuciones de los datos y capturar tanto la variabilidad como las relaciones complejas entre ellos. Aunque no superan a *Bit Diffusion* en la captura de complejidades, su capacidad para generar datos que siguen fielmente las distribuciones de los datos reales sin perder de vista la variabilidad los posiciona como una elección equilibrada para una amplia gama de aplicaciones.

12. Relación entre el proyecto y la carrera

En esta sección se destacarán las competencias técnicas aplicadas a partir de las asignaturas cursadas a lo largo de la carrera:

- **IES y PROP**: Estas asignaturas proporcionaron las bases para una gestión efectiva de proyectos de larga duración y la elaboración de casos de uso relevantes para este trabajo.

- **MD, APA, IA, CAIM**: Las competencias adquiridas en estas asignaturas fueron fundamentales para llevar a cabo un análisis exhaustivo de los datos previos al proyecto. Los conocimientos aplicados permitieron procesar y evaluar diversos modelos, ajustando los hiper parámetros de manera adecuada para distintos escenarios.

- **Algoritmia y EDA**: Los conocimientos adquiridos en estas asignaturas posibilitaron la descomposición eficiente de problemas complejos en subproblemas más manejables (divide y vencerás), aplicando los algoritmos pertinentes.

- **PE**: Los conceptos de probabilidad y estadística de esta asignatura resultaron clave para evaluar los resultados de los modelos. Se llevaron a cabo análisis estadísticos de las distribuciones de datos, aplicando conceptos relevantes para optimizar los resultados.

Estas asignaturas han proporcionado las herramientas y habilidades necesarias para llevar a cabo este proyecto de inteligencia artificial, siendo la mayoría de estas parte de mi especialidad en computación.

13. Futuros proyectos

Este trabajo representa solo el inicio del proyecto de la generación de datos tabulares artificiales. El proyecto, disponible en GitHub [13], está abierto a futuras contribuciones y desarrollos en áreas clave:

- **Integración de nuevas técnicas generativas**: Explorar y añadir más métodos para la creación de datos puede enriquecer las capacidades de la librería, haciéndola más versátil y efectiva. Esto puede ayudar a que sean comparados muchísimos más modelos y poder obtener resultados más exactos.

- **Optimización de los métodos de evaluación**: Implementar métricas más avanzadas, detalladas y específicas para cada caso mejorará la capacidad de evaluar con precisión la calidad y el realismo de los datos generados.

- **Perfeccionamiento de modelos actuales**: Al ser una librería con distintos modelos todo el trabajo no se ha centrado en refinar un único modelo, por lo tanto, se puede continuar optimizando y ajustando los modelos existentes para obtener un rendimiento y una eficiencia mejorados.

El proyecto presenta un amplio margen para el crecimiento y la mejora continua. Una de sus grandes fortalezas es la estructura en módulos en la que se basa. Al estar dividido en diferentes módulos especializados, los desarrolladores pueden enfocarse en áreas específicas sin necesidad de un conocimiento exhaustivo sobre el resto de componentes.

Esta modularidad facilita las contribuciones dirigidas y eficientes, permitiendo que distintos expertos aporten a cada segmento del proyecto. Esto implica que el potencial de enriquecimiento y perfeccionamiento por parte de la comunidad de desarrolladores e investigadores es significativo, abriendo una puerta a la mejora continua del proyecto.

14. Conclusiones

La realización de este trabajo representa un hito significativo en el ámbito de la inteligencia artificial aplicada a la generación de datos tabulares. Se ha logrado con éxito desarrollar una librería accesible y funcional que ofrece a investigadores y entusiastas del área la posibilidad de experimentar con diversos modelos generativos. Este proyecto no solo provee una base sólida para futuras investigaciones y desarrollos en el sector sino que también invita a la mejora y expansión de los métodos ya implementados.

A pesar de las dificultades encontradas, especialmente debido a la escasez de recursos y estudios previos en el nicho específico de la inteligencia artificial generativa para datos tabulares, el trabajo cumplió con los objetivos propuestos, proporcionando un aprendizaje profundo y una experiencia valiosa en el campo del *deep learning*. El desarrollo de este proyecto ha sido un desafío tanto a nivel personal como profesional, impulsando un crecimiento significativo en el conocimiento y aplicación de la inteligencia artificial.

La satisfacción con el progreso y los resultados obtenidos es considerable, ya que se ha conseguido consolidar una comprensión profunda de distintos modelos generativos y su potencial en el tratamiento y análisis de datos tabulares, un área que representa una parte fundamental de mi interés y ambición profesional en el *deep learning*.

Finalmente, deseo expresar mi sincero agradecimiento a Javier Béjar, cuya propuesta de este tema desafiante fue el catalizador de este proyecto. Su orientación y los recursos proporcionados han sido esenciales para navegar por el complejo terreno de la inteligencia artificial generativa. Este trabajo es testimonio de la intersección entre la curiosidad intelectual y la colaboración efectiva en la búsqueda del conocimiento y la innovación.

Referencias

[1] Universidad de Granada. *Diseño Estadístico de Experimentos*. *https://www.ugr.es/~bioestad/_private/cpfund3.pdf* [Online; 24-Sep-2023]. 2019.

[2] Talent.com. Talent. https://es.talent.com. [Online; 05-Oct-2023]. 2023.

[3] Boletín Oficial del Estado. General Data Protection and Regulation, Reglamento del Parlamento Europeo y del Consejo. https://www.boe.es/doue/2016/119/L00001-00088.pdf. [Online; 08-Dic-2023]. 2016.

[4] Boletín Oficial del Estado. Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales. https://www.boe.es/eli/es/lo/2018/12/05/3/con. [Online; 08-Dic-2023]. 2018.

[5] Secretaría de Estado de Digitalización e Inteligencia Artificial. Estrategia Nacional de Inteligencia Artificial. Estrategia Nacional de Inteligencia Artificial. https://www.lamoncloa.gob.es/presidente/actividades/Documents/2020/ENIA2B.pdf. [Online; 08-Dic-2023]. 2020.

[6] YData. Librería de generación de datos artificiales ydata-synthetic en Github. https://github.com/ydataai/ydata-synthetic. [Online; 12-Sep-2023]. 2020.

[7] SDV-Developers. Librería de generación de datos artificiales SDV en Github. https://github.com/sdv-dev/SDV [Online; 12-Sep-2023]. 2019.

[8] Jordi de la Torre. Decodificadores Variacionales (VAE) Fundamentos Teóricos y Aplicaciones. https://arxiv.org/ftp/arxiv/papers/2302/2302.09363.pdf. [Online; 20-Nov-2023]. 2023.

[9] Aparna Dhinakaran. Divergencia de Kullback-Leibler aplicada al Machine Learning. https://towardsdatascience.com/understanding-kl-divergence-f3ddc8dff254. [Online; 20-Nov-2023]. 2023.

[10] Ting Chen, Ruixiang Zhang and Geoffrey Hinton. Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning. https://arxiv.org/abs/2208.04202 [Online; 12-Sep-2023]. 2022.

[11] Hinton's Group. Implementación del modelo Bit Diffusion para la generación de imágenes. https://github.com/lucidrains/bit-diffusion. [Online; 13-Sep-2023]. 2022.

[12] Universidad Complutense de Madrid. Contrastes no paramétricos. https://www.ucm.es/data/cont/docs/518-2013-11-13-noparam.pdf. [Online; 21-Nov-2023]. 2013.

[13] Marc Nebot i Moyano. Proyecto de final de grado *Métodos para la generación de datos artificiales tabulares.* https://github.com/marcnebotmoyano/TFG-FIB-GenerativeArtificialTabularDataMethods . 2024