NayaOne Data Science Project - Improve GAN Model
BUSM130 Group Project

George Karseras, Mateusz Milczarek, Dev Upadhyay, Hussain Aftab, Anurag Sinha
180098895 210822119 200644879 200596198 210028618

NayaOne Queen Mary University of London

Data Science Project – Improve GAN Model

Supervisors: Professor Linus Wunderlich and Professor Natalia Efremova

MSc Business Analytics, Queen Mary University of London

Year: 2021/22

BUSM130 Group Project – Team 4

STUDENTS' DECLARATION

Students' Declaration

We,

Name	Student ID	Email
George Karseras	180098895	bs211529@qmul.ac.uk
Mateusz Milczarek	210822119	bs211411@qmul.ac.uk
Dev Upadhyay	200644879	bs21084@qmul.ac.uk
Hussain Aftab	200596198	bs21016@qmul.ac.uk
Anurag Sinha	210028618	bs21105@qmul.ac.uk

hereby declare that the work in this Group Project Report is our original work. We have not copied from any other students' work, work of ours submitted elsewhere, or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for us by another person. The individual sections of the report have been written by the author in the section title with no external assistance.

Referenced text has been flagged by:

- Using italic fonts, and
- using quotation marks " ", and
- explicitly mentioning the source in the text.

TABLE OF CONTENTS

S. No.	Topic	Page Number
1.	EXECUTIVE SUMMARY – 1860 Words	3
2.	INTRODUCTION – MOTIVATION	6
3.	STRUCTURE	6
4.	BUSINESS CASE	7
5.	WHAT ARE GANS	8
6.	GAN SYNTHETIC DATA AND ITS APPLICATIONS	11
7.	THE MATHEMATICS OF GENERATIVE ADVERSARIAL NETWORKS	15
8.	LITERATURE REVIEW	20
9.	DATASET DESCRIPTION – LOAN DATASET	24
10.	EXPLORATORY DATA ANALYSIS	25
11.	CTGAN FITTING	28
12.	EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES	32
13.	1st DATASET CONCLUSION AND EVALUATION LIMITATIONS	35
14.	DATASET DESCRIPTION – COMPLAINT DATASET	36
15.	DATA PRE-PROCESSING	37
16.	EXPLORATORY DATA ANALYSIS	37
17.	SENTIMENT ANALYSIS	40
18.	CTGAN SYNTHESIZER MODEL FITTING	47
19.	EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES	48
20.	2nd DATASET CONCLUSION AND EVALUATION LIMITATIONS	52
21.	REFERNECES	53
22.	APPENDIX	57
23.	INPUT DATA LINK	59
24.	2 DATASETS CODES ATTACHED – TWO FILES	

1. EXECUTIVE SUMMARY – Group Work 1860 Words

Data collection, management and analysis is now an essential component of managing a fast-paced dynamic business. As on one hand technological advancements have made it a very simple process for businesses of all sizes and sectors to start doing the same, on the other hand privacy, regulatory and legal guidelines make it difficult to play around with data on a real time basis. Most of largest corporations in the world based on market value are tech firms (MAAAN – Meta, Alphabet, Apple, Amazon, Netflix) that have historically acquired and analysed large sets of intrinsic data for business use. Businesses are using data strategically to better understand their users, customers, and market sectors in a variety of industries, including finance, insurance, media, and hospitality.

The motivation for this group project comes from the innovation-hampering effect that General Data Protection Regulation (EU GDPR) policies have on technology providers nowadays. For one to design and develop new, innovative solutions for a broad range of financial institutions, they need to have access to representative, relevant data. Such data, however, very often contains lots of personally identifiable information (PII in short) and individuals' protection imposed by GDPR prevents technology developments from working on such data without tedious and complicated legal processes. As such, this is where synthetic data comes extremely useful, as it would be statistically accurate and, hence, as representative as the real data, but since it would not contain any PII (simply because it would be artificially created), their use would not be legally challenging.

NayaOne is a new age fin-tech organisation that is revolutionising innovation in financial services. It provides banks with a one stop shop and access of actionable data, insights to hundreds of fintechs, large investment and commercial banks including regulators, through a specialised Digital Sandbox and Fintech-as-a Service offering. Organisations in a regulated market space like banking and healthcare, are able to discover, build, evaluate and scale with data in a matter of weeks instead of months. The business case at hand, refers to the process of producing synthetic data free of PIIs that the above entities can use freely, without any legal implications. For this project, our aim is to try to improve the open-source CTGAN, producing synthetic data that matches the original data, while at the same time keeping its original statistical properties intact.

Two datasets namely a compiled loan dataset and a customer complaint dataset have been utilized on this report that has been made available through the support of LendingClub. LendingClub operates as a peer-to-peer (p2p) personal loan lending marketplace website. The company assesses an applicant's (borrower) risk and allows end-investors (lenders) to make direct loans to individuals or spread funds across multiple loans to diversify risks. The borrower gets charged a processing fee of 1% to 5% (depending on credit risk) and the lender gets charged a 1% processing fee of the loan amount. Typically, the margin on the rates is beneficial to the all the three parties involved as compared to a traditional bank. Since its launch in 2007, the platform has facilitated over $5 billion in loans.

Capturing and modelling real world data has several issues especially related to the regulations on privacy. Collecting huge sets of PII hinders organizations' development towards long term strategic growth based on existing customers data. These issues can be resolved by creating synthetic data that accurately captures the key statistical characteristics of the underlying real-world data. In comparison to gathering huge datasets, it is less expensive and can enable software testing or the creation of AI/deep learning models without endangering the privacy of the users. By 2030, it is predicted that 65% of the data required to create AI and analytics projects would be created artificially. The applications and the level of customization GANs can offer is endless. Its usage is evident across

Automotive and Robotics, Manufacturing, Food and Beverages, Marketing, Banking, Financial Services, Cyber-Security, Machine Learning, Healthcare, Social Media, and Human Resources management among others.

Research paper analysis on variations of GAN (MaliGAN, MedGAN, EHR-GAN, PATE-GAN, TimeGAN, Quant GAN, LSTM GAN) highlights the importance of variable types in model fitting. With mixed modal data that contains categorical and discrete tabular values we see GANs inherently failing due to its limitations of not being able to analyse both together. Conditional GANs on the other hand has been able to successfully able to cater to both the limitations of modality and multi nominal distribution. CT GAN is able to learn better simulations than a Bayesian network whereby using a mode specific normalization to convert continuous columns of data into a bounded vector that is usable for a neural network. The conditional generator can help generate data with specific discrete values that can be used for the fake data creation.

The mathematics behind conventional GANs are thoroughly discussed in the report, emphasizing on the infrastructure of the adversarial game involving both the discriminator and the generator. Emphasis is also given to specific procedures of CTGAN, value functions, adversarial game, and optimization of loss functions through stochastic process. Among these procedures, both information and implementation of Mode-Specific Normalization and training by sampling are discussed, along with an assessment of the improvements they offer, contrary to conventional GAN methods.

The dataset we worked with, initially contained 887,379 observations and 73 columns that held information on loans issued in the US in the years from 2007 to 2015. The information present provided insights on the loan's specifications but also on the customer the loan was issued to. The dataset was mostly numerical in nature but did contain several categorical columns as well.

The pre-processing done in the report, mostly emphasized on dropping columns with high number of missing values and imputing columns that were possible to impute, without affecting the initial dataset's statistical originality. The need for this pre-processing procedure, arises by examining the limitations of CTGAN, that have to do with the inability of the algorithm to create synthetic data, by using a dataset with any number of null values. The final form of the dataset, we were left with included 773,968 observations and 42 columns, of which none contained any null entries.

By comparing distributions and correlations of several variables from both the original sample, used in the fitment of CTGAN, and the synthetic sample generated, we were able to determine that the synthetic data was able to capture original correlations and relationships, that were initially present. One of the limitations observed by visually and statistically inspecting the two datasets was the inability of the CTGAN to efficiently depict outliers present in the original sample, on the synthetic data generated. This limitation was observed by observing the statistics of annual income in the two datasets.

Through the introduction of certain constraints in the modelling part of the report and efficient pre-processing, the CTGAN's performance, was satisfactory, in terms of the degree of originality present in the synthetic data. This is also evident by observing Figure 3 in the Appendix, where a 0.89 score of correlation is obtained for the column correlations in the two datasets.

The second dataset i.e., the complaint dataset dimensions are 555,957 rows by 18 columns. In the customer complaint data collection, we have a feature called complaint id, which has the data type int64. The all other 17 features, however, have object data types. Following an initial review of the information contained within the dataset's data, certain features were eliminated. These included sub-product, sub-issue, consumer complaint narrative, company public response, tags, consumer

consent, complaint id, zip code, and others. Most of these features contained maximum null values and data that was deemed to be of little or no use for our analysis. Subsequently there were just 5 characteristics left.

In our initial exploratory data analysis, we created bar graphs and pie charts for product and company responses to consumer features, respectively, to acquire glimpses of the features and the frequencies of data values. This was done after the pre-processing was done in the report. Even after deleting features that had null values, there were still a few variables that had null values; hence, we decided to eliminate all rows from the dataset that had null-valued columns. Following that, the dimensions of our dataset were altered to be 551,070 by 5.

As the next step in our investigation, we chose to analyze sentiment, and while doing so, we found that the concerns had contractions like "can't" as well as punctuation like "," etc. and "." These will be dealt with by performing the operations such as lowercasing the reviews, expanding the contractions, and removing the punctuations. In addition, to obtain a more in-depth understanding, we counted the most common stop words, words, bigrams, and trigrams. To clean up our data and prepare it for future examination, we performed stop words removal, lemmatization, and created a document term matrix for sentiment analysis. These processes helped us gain a deeper understanding about customer issues, which is necessary for sanitizing our data. After developing the Document Term Matrix, we developed custom word clouds for each product, which ultimately assisted in better comprehending the general public's opinion surrounding each product. Additionally, we developed graphs depicting product polarity, which show that the vast majority of products contain a negative charge.

We trained GANs to produce synthetic data from the tabular data. However, the provided dataset is too large for our local system, so we randomly subset it using a sample function whose fraction parameter determined the subset size. 55000 rows from the dataset equals 0.1 frac or ~10% of dataset. Further we defined a CTGAN Synthesizer instance and called fit with the data frame and categorical variables. After 100 cycles, discriminator and generator loss became negligible. After training, the model's sample function is called to create samples. Eventually, the model was able to generate 55000 synthetic data samples.

The sample subset and synthetic sample both have more than 17,500 mortgage records. Synthetic data collection has 14,000 mortgages, fewer than the original subset. Word cloud displayed column data well. Size and color indicate a word's frequency or importance. Loan, modification, collection, and foreclosure dominate the issue column word cloud for both the original subset sample and the synthetic data sample. In the pie chart, the web option occurs most often in the submitted via column, yet there is a 19% difference between the original subset and the synthetic sample, with 65% in the original subset and 46% in the synthetic sample. A horizontal bar graph shows that firm response to consumer columns is highest when closed with explanation. Some responses are positioned differently in both samples.

Our evaluations consistently yielded an accuracy of 60% or higher which is a fair outcome through GAN. We were unable to handle additional data because of technical constraints, however, even with this reduced dataset, we can achieve satisfactory accuracy. This gives us high hopes that, with more data processing and possible hyperparameter tuning, we could improve accuracy even further. Because the model is not designed to work with relational datasets by default, we had to get rid of the complaint id before we started modeling. This modeling approach cannot handle limitations because that is not its default setting.

2. INTRODUCTION - MOTIVATION

The motivation for this group project comes from the innovation-hampering effect that General Data Protection Regulation (EU GDPR) policies have on technology providers nowadays. For one to design and develop new, innovative solutions for a broad range of financial institutions, they need to have access to representative, relevant data. Such data, however, very often contains lots of personally identifiable information (PII in short) and individuals' protection imposed by GDPR prevents technology developments from working on such data without tedious and complicated legal processes. As such, this is where synthetic data comes extremely useful, as it would be statistically accurate and, hence, as representative as the real data, but since it would not contain any PII (simply because it would be artificially created), their use would not be legally challenging.

There seems to be a few methods to produce synthetic data, but Generative Adversarial Networks (GAN) models are particularly interesting. Although researchers keep emphasizing GANs' outstanding capabilities on data augmentation, dealing with missing data problems and other feasible application areas, the case that led to this project is the capability of Generative Adversarial Networks to generate artificial data that would maintain the (statistical) attributes of the original data. As such, usage of such generated data would allow for exploration of attributes of private datasets with no breach to privacy regulations and no risk of sharing any pieces of sensitive information.

3. STRUCTURE

The report begins with the motivation and the business case we have targeted for this report. Further, GAN background and the structural mathematics is illustrated. The relevance of Synthetic Data through GAN and its applications across five industries are covered. Over twelve research papers are referred in the literature review section to highlight the continuous development of the GAN framework across industries. For the first dataset Exploratory Data Analysis is performed along with the parsimonious cleaning, evaluation of different hyperparameter infused sampled models. Then Exploratory Data Analysis on the second dataset (text string) is performed with subsequent model evaluation of generated samples with the original dataset. Towards the end, a summarization, and the impact/results of the generated data of this report are concluded.

4. BUSINESS CASE

NayaOne acts as a single point of access to many FinTechs and datasets, enabling banks, regulators, and other financial institutions to develop new innovative practices and solutions in the financial industry. The business case at hand, refers to the process of producing synthetic data free of PIIs that the above entities can use freely, without any legal implications. For this project, our aim is to try to improve the open-source CTGAN, producing synthetic data that matches the original data, while at the same time keeping its originality intact.

The data we will try to successfully replicate below, consists of two different datasets. The first dataset consists of loan and customer information given out by Lending Club in the US from 2007 to 2015 and is mostly numerical in nature. The second dataset is more string oriented containing information on the complaints by customers in different financial institutions in the US, sourced from the Consumer Financial Protection Bureau website. Our ultimate goal is to create a GAN, or two different GANs, through adequate data pre-processing and hyper-parameter tuning to efficiently replicate both datasets, stated above.

Over both the data sets the GAN performance was fairly well achieved, despite of the limitations in our GPU/system. Through the introduction of certain constraints in the modelling part of the report and efficient pre-processing, the CTGAN's performance on the first loan dataset, was satisfactory, in terms of the degree of originality present in the synthetic data. This is also evident by observing Figure 3 in the Appendix, where a 0.89 score of correlation is obtained for the column correlations in the two datasets. For the second complaint dataset our evaluations consistently yielded an accuracy of 60% or higher which is a fair outcome through GAN.

Project deliverables include mathematical model and code for the technical aspects of the project. Report on method used to perform the data generation and the results, including the level of accuracy obtained.

Goal 1 - Improve on the CTGAN for numeric and categorical data – **Fair results with accuracy of 63+%**

Goal 2 - Enable the same GAN to work with string data – **Fair results with accuracy of 60+%**

5. WHAT ARE GANs?

Generative Adversarial Networks or GAN, as an unsupervised deep learning technique, has first arisen in 2014 when Ian Goodfellow published an article on two separate neural networks working together to generate artificial data that has attributes alike to the real data (Mohammed Alhamid, 2020) - they learn about objects that exist and then they can create new versions of those objects that never existed in reality (Udemy Computer Vision, n.d.). The two neural networks are the Generator and the Discriminator, and they work in tandem - as Ian Goodfellow himself described it, the generator is like a counterfeiter, trying to produce fake money without getting caught, while the discriminator is like a police officer, trying to detect the fake money and stop its circulation (Patrick L, 2015). What's worth mentioning, both models start "from scratch" – i.e., they are not trained before they start competing with each other, and in a way, they learn together. The most used inputs to the model are random noise (for the generator to serve as a starting point) and real objects/images (for discriminator to learn and provide feedback, via backpropagation, to the generator) (Piacentino, 2019).

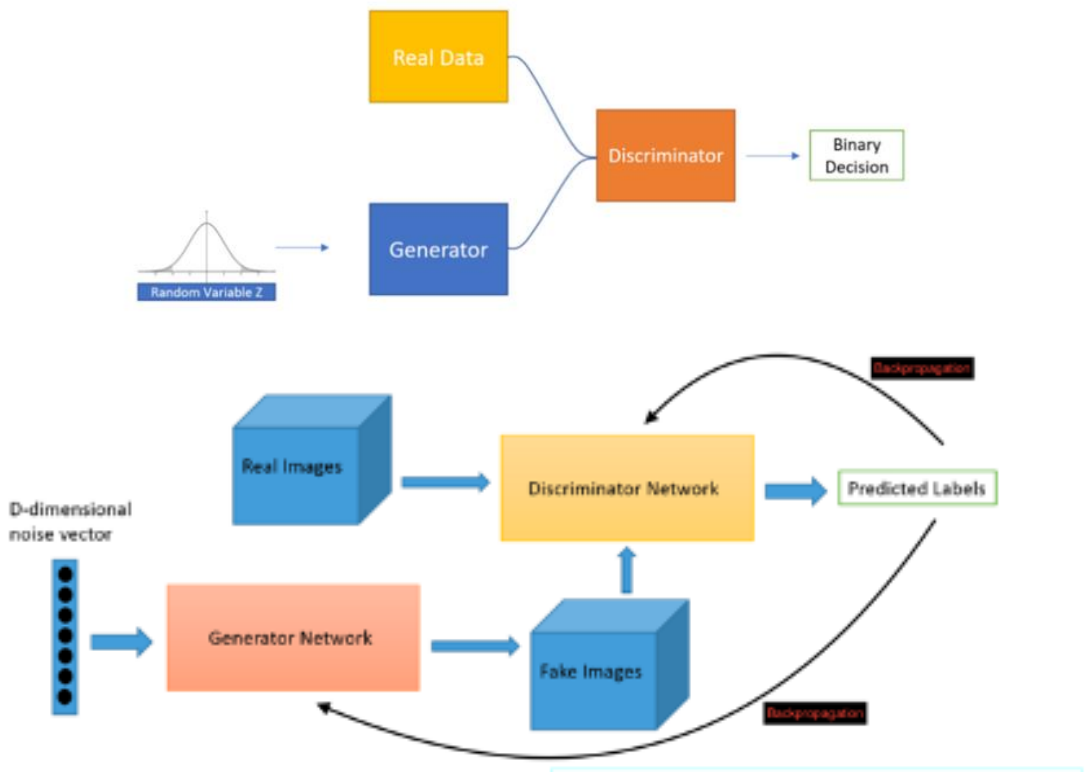


Figure 1: GAN structure (Piacentino, 2019)

The training process consists of loops called epochs. Each epoch includes the following steps (Piacentino, 2019):

- Feeding real images - a batch of real images is fed into the discriminator, and, from the discriminator output, the errors are calculated, having in mind that all images were real, and as such the output should be all ones.
- Generating fake images - a batch of noise vectors is used by the generator to create a batch of fake images, which are then fed into the discriminator. Again, from the discriminator output, the errors are calculated, having in mind that all images were fake, and as such the output should be all zeros.
- Discriminator training - errors from both calculations are backpropagated to the discriminator network so that its weights are updated.
- Generating fake images – step 2 is repeated using a new batch of noise vectors. Then, however, the errors are calculated, having in mind that we aim at fooling the discriminator and the images should be good imitations of the real ones, hence the output should be all ones.
- Generator training – the calculated errors are backpropagated to the generator network so that its weights are updated.
- Repeat all steps with another batch of real images for a given epoch. After each batch processed, both generator and discriminator models are trained by means of backpropagation of errors calculated from differences between both models' outputs and the expected values, as per the steps outlined above.
- The generator aims to maximize the discriminator error, whilst the discriminator aims to minimize it, and the process should ideally end when the discriminator attains the error of 0.5 (Piacentino, 2019). Since discriminator is essentially engaged in binary classification, this corresponds to a case in which discriminator's decision making is no better than a random process.

The basic structure of the GAN, as presented in Figure 1, can be commonly enhanced by another network feeding the generator, such that the images that it generates are based on another source of information, like a text or another image (Piacentino, 2019).

GAN WITH TABULAR DATA

GAN can generate images and work well for image generation. However, as specified in the objective of this report, generating synthetic tabular data for regression or classification related purpose is of prime focus. GANs can help to generate categorical or continuous data values for new observations.

GAN does this through using a typical discriminator and generator in neural networks. One neural network generates the data and the other tries to determine its quality and validation for real and fake data. The data generated is checked and compared for consistency in the range of values and root mean square values on predicted regression values vs the real-world root mean square values. In GAN training, the two networks have opposing objectives.

- While the generator's aim is to deceive the discriminator, the discriminator seeks to optimize its classification accuracy by properly recognizing which pictures came from it.

- By the time it has finished learning, the generator should be able to create data that resemble the original dataset so closely that analysis would mistake them for the genuine thing (Jeff Heaton , 2021) (Ashrapov, 2020).

Categorical and Discrete data are easy to represent in vectors through one hot encoding. The act of dividing each category in a discrete variable into its own dimension is known as one-hot encoding (Unzueta, 2021).

	Monday	Tuesday	Wednesday	Thursday	Friday
1	1	0	0	0	0
2	0	1	0	0	0
..					

Table 1: One hot encoding for discrete data values

For continuous data which are more complex, a mode-specific normalization which transforms the continuous variable into a vector containing the information is recommended. To begin with, each continuous variable is fitted with a VGM (variational Gaussian mixture model). A gaussian mixture model only uses expectation maximization to try to choose the optimal k Gaussians to represent the data. Through a weights threshold, VGMs may determine the optimal number of Gaussians (k) to fit onto the data. More details around the mathematical expressions and working can be followed in the mathematical section of this report (Unzueta, 2021) (Lei Xu, 2019).

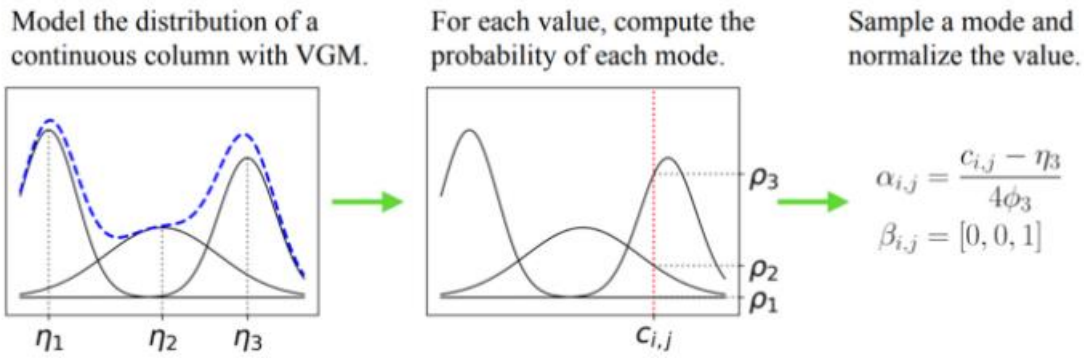


Figure 2: k Gaussian distributions

Despite of challenges like simultaneously processing of discrete and continuous features, multi-modal non gaussian values, and imbalance within categorical features, GANs still offer greater flexibility in modeling distributions than their statistical counterparts.

6. GAN SYNTHETIC DATA AND ITS APPLICATIONS

Capturing and modeling real world data has several issues especially related to the regulations on privacy. Collecting huge sets of PII hinders organizations' development towards long term strategic growth based on existing customers data. These issues can be resolved by creating synthetic data that accurately captures the key statistical characteristics of the underlying real-world data. In comparison to gathering huge datasets, it is less expensive and can enable software testing or the creation of AI/deep learning models without endangering the privacy of the users. By 2030, it is predicted that 65% of the data required to create AI and analytics projects would be created artificially. (Cem Dilmegani, 2022)

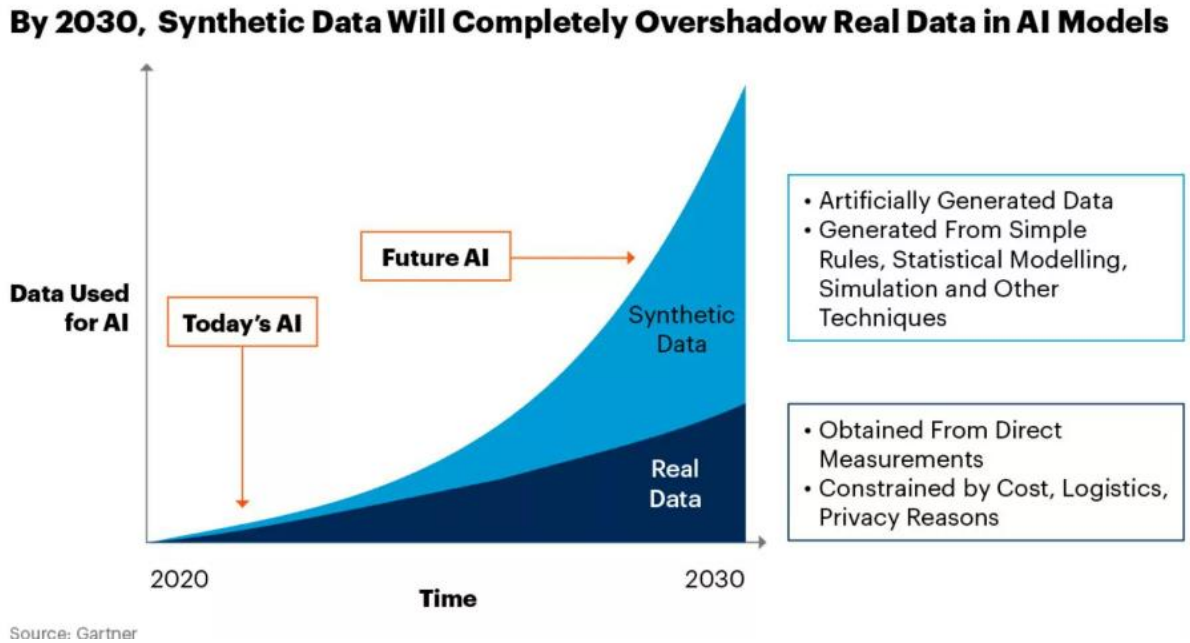


Figure 3: Synthetic data growth

Industries and Business Functions that can benefit from synthetic data:

Automotive and Robotics	Manufacturing	Marketing
Financial services	Cyber-Security	Machine Learning
Healthcare	Social Media	Agile development and DevOps
Internet	Packaged Goods	HR

Banking, Financial Services and Insurance

As the focus of this report's dataset, GAN applications are interestingly useful in assessing customer demographic and behavioural details pertaining to promoting wealth management financial products, optimization and churning revenue, launching/issuing new category of credit cards, disbursement of loans and scaling customer service teams for issues resolution. The dataset available for this report is from LendingClub, which is a P2P marketplace lending platform. GANs can help institutions to better

design their offerings, test new processes, scale customer service team and channelise focussed marketing solutions to real world after testing on synthetic data.

PII and other intrinsic details of a customer/transaction like customer behaviours, demographics, buying, and payment patterns are often blocked and subject to high level of privacy, security, and regulations. For example, customers can be divided into segments based on median income, past median credit card balance, payment history etc categorized as Prosperous and content, Deal chasers, Financially stressed etc and cover facts such as income, age, and geography (McKinsey, 2021) (McKinsey - Behavioral insights, 2018).

	Prosperous and content	Deal chasers	Financially stressed	Recovering credit users	Self-aware avoiders
Percent of U.S. credit card holders in segment	23%	18%	19%	22%	18%
Median annual household income	$85,000	$65,000	$45,000	$45,000	$55,000
Percent of segment with revolving credit card balance	29%	81%	93%	64%	64%
Mean credit card revolving balance per household	$890	$3,802	$7,453	$1,726	$1,969
Most-used instrument for POS payments	Credit	Debit	Debit	Debit	Cash
Share of credit card in POS spending	59%	24%	20%	11%	19%
Most-used credit card does not earn rewards	8%	37%	52%	45%	43%

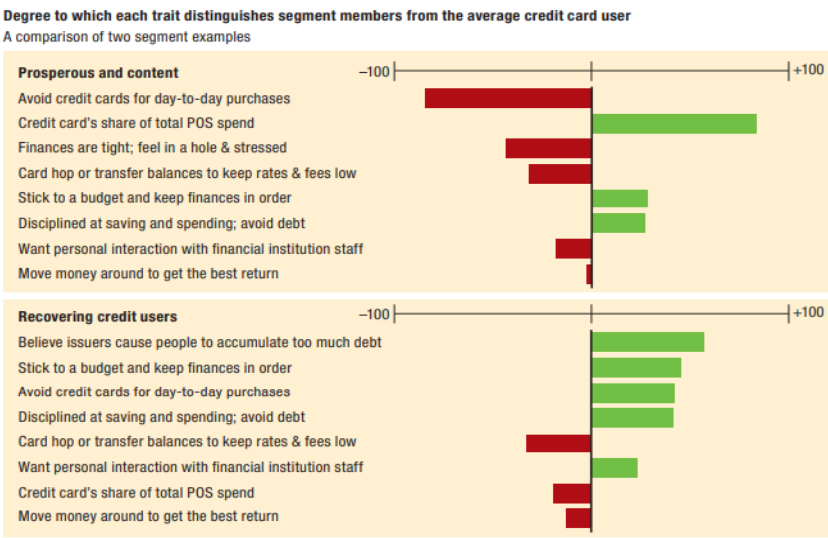


Figure 3.1: Typical Behavioral customer segments
(McKinsey, 2021) and (McKinsey - Behavioral insights, 2018)

Further, in the fraud detection industry, the financial sector is reducing card fraud by implementing research in high-tech security measures, which include keeping an eye on real-time transaction data and behavioural biometrics on devices as well as creating fraud screening and detection machine learning technologies that are accessible to retailers. Another illustration is the usage of generating dynamic codes in place of static cryptograms to reduce merchant liability for fraud connected to card data theft (Nilson Report, 2021) (Thales Group, 2022) (Experian, 2021). However, for any ML model, data is a fundamental aspect and through GAN new systems can be improved further and trained quicker.

UK's Financial Conduct Authority (regulator) developed a testing environment (sandbox) for financial concepts, goods, and services to encourage innovation. It gives an example of a scalable, comprehensive development environment for testing applications with a focus on simulating real-

world situations and perform stress testing. Over 20 groups including established players in the banking industry actively participated in this initiative's test run (Florian Eckerli, 2021).

American Express, a global financial services and credit card company, has an AI lab that creates synthetic data using combination of Conditional and Deep Regret Analytic GANs to enhance its fraud detection algorithms in an effort to find answers to several problems. The nitty-gritty of the code has not been disclosed publicly.

Some of the other major developments, are in market and customer prediction, stress testing, tuning of trading models, portfolio and wealth management, fraud detection and other types of optimizations.

Field	Application	Method
Time Series Forecasting	Market Prediction	GAN-FD [9], ST-GAN [19], MTSGAN [20]
	Fine-Tuning of trading models	C-GAN [10], MAS-GAN[21]
Portfolio Management	Porfolio Optimization	PAGAN[11], GAN-MP[22], DAT-CGAN[23], CorrGAN[12]
Time Series Generation	Synthetic time series generation and Finance Data Augmentation	TimeGAN[24], WGAN-GP[25], FIN-GAN[3], Quant GAN[14], RA-GAN[26], CDRAGAN[27], SigCWGAN[28], ST-GAN[19]
Fraud Detection	Detection of market manipulation	LSTM-GAN[13]
	Detection of Credit Card Fraud	RWGAN[29], LSTM-GAN-2[30]

Figure 4: GAN in finance research (Florian Eckerli, 2021)

Internet

Alphabet's, Google's parent company, reported an annual revenue of USD 258 billion in the year 2021 and more than 80% of this revenue is through ads. Despite of the pandemic, general slowness in business and consumer spending and Apple's App Tracking Transparency privacy changes in iOS in 2020, requiring users to opt in if they wanted to allow apps to track them across other apps and on websites, Google crossed the 200-billion-mark figures. Earlier in the year, the Financial Times reported that Facebook (now Meta), Twitter, Snapchat and YouTube lost forego USD 10 billion in revenue in 2021 because of the new iOS settings (The Verge, 2022).

To report such high figures, one of the essences of effective advertising reach is having the identifiable users' data which helps organizations capture behavioral targeting components. However, with GDPR and other strict regulations advertisers are forced to let go of the cookie-based tracking promotions. This has created an opportunity for the use of synthetic data through GAN.

More restrictions on the use of third party cookies, which are the most common mechanism for cross-site tracking today have made Google build the Privacy Sandbox on its existing efforts on the web, providing a clear path forward to improve user privacy without putting access to free content and services at risk.

GANs help to deliver interest-based advertising without needing to collect a particular individual's browsing history. Browsers would require a method to create artificial and private clusters: Utilizable by gathering individuals with sufficiently comparable interests and creating labels appropriate for

machine learning, and private by establishing sizable synthetic clusters that do not reveal too sensitive information. The combination of synthetic data and clusters is made available to websites via a new JavaScript API. Browsers use machine learning algorithms to create cohorts based on the websites users visit. The central idea is that these input characteristics of algorithms, including web logs, are stored locally in the browser, and not uploaded elsewhere. The browser only provides the generated synthetic cohort. The browser ensures that the cohort is properly distributed, and each represents thousands of people. Browsers can also use other anonymization methods, such as differential privacy. As people's browsing behaviour changes, their cohort changes too, but the algorithm that turns input features into cohort assignments should remain stable (FLoC alog, 2022) (Google User Trust Chetna Bindra, 2021).

In the same industry, Meta (formerly Facebook), which has been suspected of hearing users' conversations for target-based ads, uses several machine learning algorithms including GAN to experiment and evaluate user classifications. The lack of transparency about what is really going on behind the Facebook curtain is what can lead users to jump to technically inaccurate but emotionally on-point explanations for creepy ad phenomena (EFF Gennie Gebhart, 2018).

Digital Marketing

In another application, a renowned US-based media agency worked with GAN to generate more targeted advertisements. With growing competition and choices available for consumers, a personalization approach on social media brand messaging becomes essentially helpful, and therefore to automate customer behavior analysis, the agency used Random Forest Regressor Shapley Additive explanations model for text and impact diagnosis and GAN to generate unique personalized ad text. It helped them achieve a 3X increase in return on advertising spend (Algoscale, 2021).

Food and Beverages - Consumer Packaged Goods

Food and fast-moving consumer goods company often experiment or complete end-to-end lengthy research and development efforts in new product development and branding. For example, to launch a new flavor of cookies, the company often works for months with the food regulators, suppliers, data scientists, chemical engineers, health practitioners, professional tasters, product managers, psychologist, behavior marketing team etc. to create an exciting new flavor and launch in the market. With the help of GAN ML, most of the steps including the availability of raw materials, ingredient combination and recipe compatibility could be automated for undesired results. The ML system can shortlist 2 or 3 from hundreds of permutation and combination possibilities, generate genre specific flavors which can then be implemented in lab in a quicker time. One of the results of such GAN work has been successfully done by a top tier consulting firm analytics division for their client Mondelez (Cadbury Oreo), in Denmark market.

Healthcare

GANs are also used in generating healthcare records to understand personal demographic details, family health history for hereditary diseases, wellbeing, medical history, trends, correlations of drug efficacy and reaction and drug discovery. To create aberrant MRI (i.e., magnetic resonance imaging) images with GANs that could be used to train a deep learning model, technology firm Nvidia collaborated with the Mayo Clinic and the MGH & BWH Center for Clinical Data Science in 2018. It helped the project's researchers in the development of low-cost, diversified data that was easy to share across institutions and had no privacy concerns. Additionally, it enabled them to create millions of unique combinations that would be difficult to produce with biological photos, such as altering the size or position of a tumour. (Hazel Tang, 2020)

7. THE MATHEMATICS OF GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks are a machine learning algorithm that is made up by two distinct components: a generator denoted with the letter G and a discriminator denoted with the letter D. These two components are trained at the same time enabling them to learn and challenge each other simultaneously, subsequently generating results of higher quality. Predictive models in ML are typically discriminative in nature where model learns the conditional probability of target variable given the input variable P(Y|X=x). E.g.: Logistic and Linear regressions.

To mathematically break-down GANs we should first understand the outcome we want to achieve when employing them. When employing a GAN to generate synthetic data our ultimate objective is to be able to recognize and identify the probability distribution of the original data and generate synthetic data derived from an approximately identical distribution. Two data sets therefore can be considered similar if they are the products of identical or almost identical probability distributions (Wang, 2020). Generative models in ML learns the joint probability distribution of input variable and output variable P(X,Y) = P(X|Y)P(Y) = P(Y|X)P(X). To predict, the model uses Bayes theorem P(Y|X)=P(X,Y)/P(X)

Generative models can be used to create new instances of data as the models learns the distribution function of data itself.

The Generative Adversarial Network that was proposed in the research paper published by Goodfellow et al. is an extension of the GANs that were first introduced but essentially follow the same procedure of formulating an adversarial game. Made up of two components as stated above D and G, undertake the adversarial game, where the generative model G predicts a probability distribution of the data, and generates data of a given dimension, while the discriminative model D predicts the probability that the sample generated can be considered to belong to the distribution of the training data at hand (Lei Xu, 2019) (Wang, 2020).

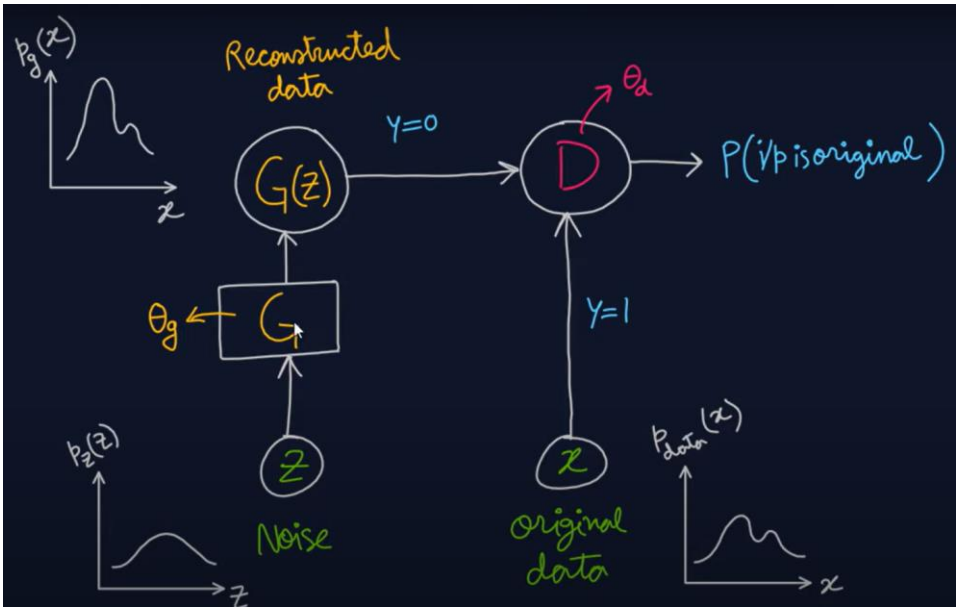


Figure 5: GAN Mathematics

- To simply for univariate target data, G&D are multi-layered neural networks.
- θg θd are weights. Noise gets passed through generator to get G(z) and distribution of G(x) is with same x as for original data as domain of original data is same as range of G(z).

- Pdata(x) is probability distribution of original data and Pz(Z) is distribution of noise, and Pg(x) is distribution for output of generator.
- When reconstructed along with original data is passed through discriminator (binary classifier) it gives a probability number of the input belonging to original data. G&D plays a minimax game (tic-tac game) to minimize or maximize value function.

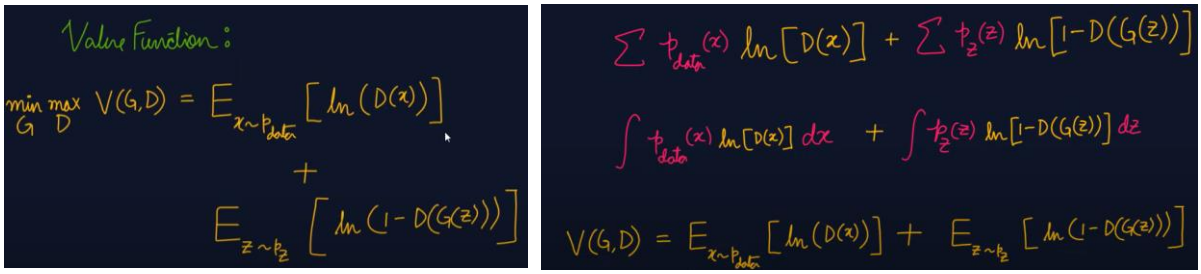


Figure 6: Value Function in GAN Mathematics

- Value function is just the expectation (weighted mean) of loss functions of G&D.
- Just as in *Binary Cross entropy function* -
- When original data y=1 is passed (Discriminator), ŷ=D(x), Loss Function = ln[D(x)]
- When fake data y=0 is passed (Generator), ŷ=D(G(x)), Loss Function = ln[1- D(G(x))]
- Adding these two and applying expectation gets the total value function. In the above image, Σ is for discrete and ∫ is for continuous distribution (Σ+∫ becomes Expectation(E)).
- **V(G,D) = E $_{Pdata(x)}$ {ln[D(x)]} + E $_{Pz(Z)}$ {ln[1- D(G(x))]}**

The adversarial game can be composed using mathematical notations as follows:

$$V(D,G) = E_{X\sim\mu}[\log D(x)]+E_{u\sim\gamma}[\log(1-D(G(u)))],$$

The above equation consists of two terms. The first term represents the discriminators predictions on the real data thus the average probability given by the discriminator that the samples obtained by the real data belong to the real distribution. While the second represents the discriminator predictions on the fake data, thus the average of the predictions when you pull a sample of noises from the generator function (Wang, 2020).

Next, optimization of loss function using stochastic process is performed in training loop.

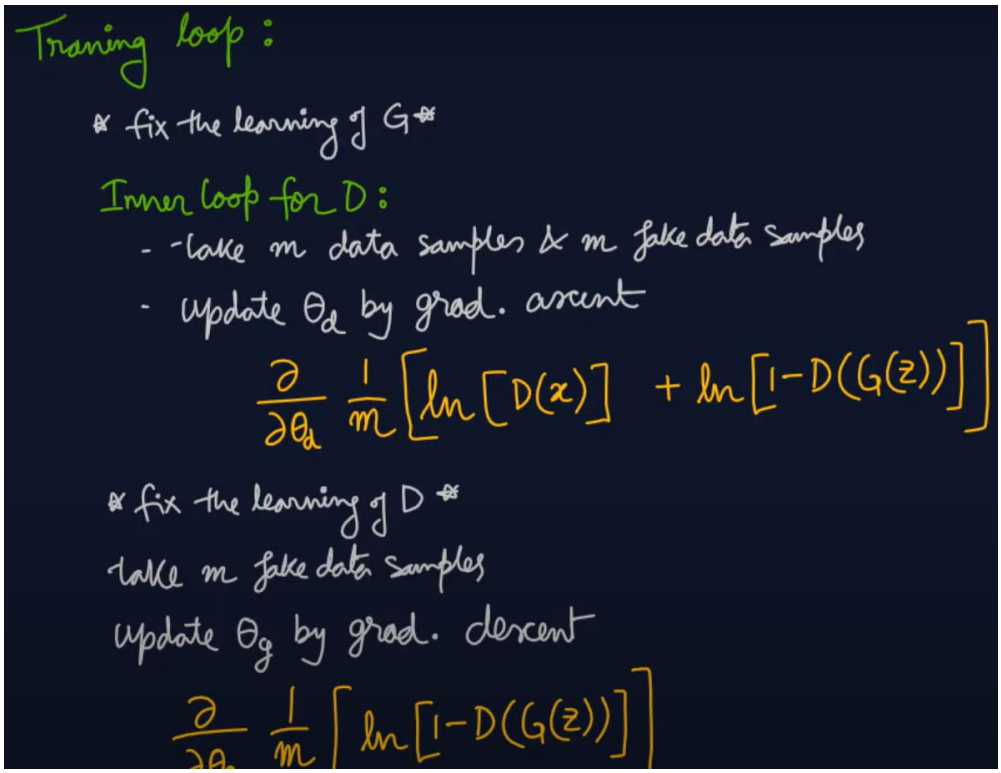


Figure 7: Optimization of loss function in GAN Mathematics

- The learning of G is fixed then the inner loop for D. This loop will continue for k steps. In this loop m data points from original and m from fake are considered.
- Then parameters of discriminator (θd) are updated by gradient ascent because discriminator is trying to maximise value function after k update of d. After it gets outside the loop, learning of d is fixed.
- Then the generator is trained through m data samples of fake data the parameters of G (θg) is updated by gradient descent. At this stage, G tries to minimize value function.
- As partial derivative of {ln[D(x)]} wrt (θg) is Zero, for every k update of discriminator, the generator is updated only once (Ian J. Goodfellow, 2014).

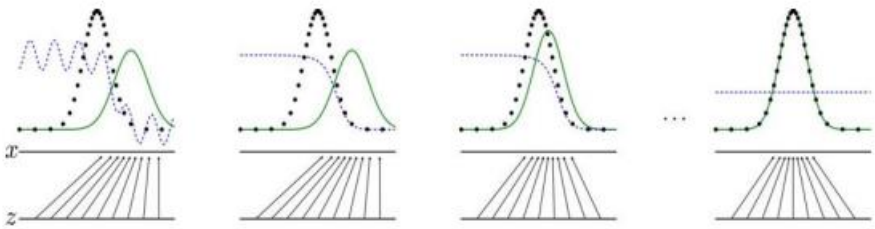


Figure 7.1: Theoretical evolution of GAN training, as the Generative distribution (green line) approaches the real data distribution, the Discriminative distribution (dotted blue line) will be unable to distinguish them and stabilizes at D(x) = ½

By nature, the discriminator is intended to maximize the first term since is the probability of a real sample being indeed real and minimize the second term since it has as a goal to correctly classify fake samples generated by 'G' as fake. However, since the second term contains the equation 1-D(G(u)), then the discriminator wants to maximize both, and the generator which wants to fool the discriminator, minimize both terms, thus the adversarial game:

$$Min_G, Max_D\ V(D,G) = E_{x\sim\mu}[logD(x)]+E_{u\sim\gamma}[log(1-D(G(u)))]$$

The discriminator role is to distinguish between the actual data and the generated(fake) data while the generator works in such way such that it can fool the discriminator (Wang, 2020). Through the repeated iterations of the above adversarial game an equilibrium is reached, where the samples generated from G will have similar or even identical distribution to the original training samples.

MATHEMATICS OF CTGAN:

During the past years, different variations of GANs have been constructed to serve different needs of the industry. In the case of tabular data, CTGan as discussed in a recent study by Lie Xu, is the most prominent way of synthesizing 'fake' data (Lei Xu, 2019).

As we know in real-life, data is rarely originating from a fixed-state distribution. CTGAN differs from conventional GANs as it uses a technique called mode-specific normalization to bypass the problem of multimodal and non-normal distributions in continuous variables. CTGAN also employs a conditional generator along with a technique referred to as 'training by sampling' to overcome the problem of disproportionate categorical variables (Lei Xu, 2019).

Mode-Specific Normalization:
The conventional approach of GANs, when it comes to modelling continuous variables, is to use min-max normalization to normalize values of a given variable between -1 to 1. When using CTGAN, each continuous variable is being processed and normalized separately using Mode-Specific Normalization, which consists of the following 3 steps:

- Use VGM (Variational Gaussian Mixture) model to make an estimation on how many different modes exist in each continuous variable and then proceed to fit a Gaussian mixture
- Compute the probability of each entry in the column originating from each of the given modes
- Sample a single mode from the Gaussian Mixture and use it to normalize each associated entry

Training by Sampling and Conditional Generator:
The traditional GAN architecture has as input a vector sampled from a Multivariate Normal Distribution, which after simultaneous training with the discriminator, a mapping function is generated mapping the Multivariate Normal Distribution to the distribution of the original data. The CTGAN however uses a conditional generator to account for imbalanced categorical variables, which are overlooked by the conventional sampling method of MVN.

Conditional Vector and Generator Loss:
Along with the implementation of a conditional generator, the discriminator is also upgraded as it is fed a conditional vector instead, which results from a technique labelled as 'training by sampling', and is described in the following steps:

- Create several zero-filled masked vectors equal to the number of discrete variables in training the dataset.
- Select a discrete variable (column) at random for each entry.
- Create a probability mass function for the entries present in the selected column, so that the probability mass of each entry is the log(frequency) of that entry in the selected column.
- Select a value from the above Probability mass function.
- Set the associated mask vector equal to 1.
- For each entry, merge the mask vectors to construct a conditional vector for the associated column (Lei Xu, 2019).

The above operation used to adequate sample the conditional vector and the training data, helps the CTGAN to properly investigate all values present in a categorical column (Lei Xu, 2019).

8. LITERATURE REVIEW

It is widely known that a large amount of data plays a pivotal role in training model. Creating a model that can churn out synthetic data that represents the actual data as close to possible brings itself with major hurdles. Banking data in our case face sever hurdles to be shared in the public for experimentation and design due to privacy policies of HIPAA and GDPR that restrict sharing personal data of anyone to be used for corroboration and analysis.

We take inspiration from the Preferred Reporting Items for Systematic Review and Meta Analysis (PRISMA) to state the literature associated with the research to define the synthetic data for banking use. Our review attempts to obtain reliable performance improvements and accurate ways of manufacturing synthetic data through data augmentation. Several existing works reported gains in such concepts where different authors had attempted through experiments. Our attempt to improve made us go back to the literature since we could not achieve high accuracy depiction in our synthetic data. This forced us to grow our review database outside the scope of synthetic data itself so that we could come up with a better alternative as well as finalize the best possible method of augmenting the data.

We first investigated all GAN related publications in the ISIC workshops after which extensive research was done under the google scholar database running the query "GAN and synthetic data for banking". This search gave us 18,700 search results with various scholarly articles. We first delve into a research publication by Sanket Kamthe, Samuel Assefa, Marc Deisenroth who use copula flows for synthetic data generation. This gives us the first insight into the copula flows and its importance in generating high fidelity synthetic data. It divides the density estimation task into two parts, estimation of univariate marginals and estimation of multivariate copula density over the univariate marginals (Sanket Kamthe, 2021). Since Copulas is beyond the state of this research, we move back to the literature to find a more relevant source of research that could improve and help us with our research.

We look into Generative Adversarial Nets (GAN) itself using the papers by Yang Wang (Wang, 2020) and Clair Little (Claire Little, 2021) where they explain due to works by Goodfellow (Ian J. Goodfellow, 2014) that introduced the concepts of generative models using simultaneous usage of a discriminator and a generator. This led us to evaluate the mathematics of the GAN's and the evolution of CT GAN's from it.

GAN's works by moving the data on the gradients back from the discriminator through the generated samples to the generator. In continuous data, it is simple but in data that is discrete, it is first encoded using one hot representation, sampled through a multinomial distribution then the output received through softmax is used. Thus, creating a distribution of discrete variables that could be used. This gives us the first insight into the challenges that our experiment will face due to having mixed data types. Since we use continuous and discrete variables both, it will have to apply Activation Functions: Softmax and Tanh on the output simultaneously in turn increasing the complexity. It also hints towards the challenge that will rise due to sparse discrete variables that have been one hot encoded and hence would lead to accuracy decreasing due to having a trivial discriminator being present that would be easily identifiable between real and fake data due to the spread of the data being vast in the real one.

Since the medical and finance field both utilize GAN's to generate synthetic data due to data privacy laws and GDPR regulations restrictions in both industries, we come across an interesting find of a research publication by Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. (Edward Choi, 2018). It explains the use of GAN in the medical industry due to the

advancement of EHR (Electronic Health Record) along with an explosion of computational health due to high quantity and quality of data being collected. However due to privacy reasons, the access to such data remains limited yet the need to utilize such data by hospitals and medical outlets remain a pressing need. It focuses on mitigating the risk of reidentification of patients through residual distinguishable patterns of various features.

The finance and insurance industry has seen various improvements in using GAN for developing synthetic data for predictive modelling discerning churn and different risk factors for de facto companies in the industry. (Tong Che, 2017) designed a new GAN named MaliGAN to generate discrete variables while the (Matt J. Kusner, 2016) used gumbel-softmax synthesizer on a Bayesian network to model correlation between different features. (Ramiro Camino, 2018) used GAN's to generate multilabel discrete records along with an auto encoder to create binary data. This evolved into MedGAN (Lei Xu, 2019) proposed using conditional GAN such as CTGAN to address problems of several modes in continuous while unbalanced modes in discrete columns that make modelling complex.

All this literature when summarized provides us with the specific use cases whereby Med GAN combines auto encoder for binary, while EHR-GAN generates augmented medical record. Table GAN aims towards generating synthetic data using neural networks whereby the generated data could be used to train classifiers while PATE-GAN generates differentially private synthetic data. The Past decade has seen the evolution of treatment of each column in a table as a random variable, modelling a joint probability distribution and then sampling from that distribution. Spatial data being modeled using decomposition trees while continuous variable using the expose of copulas. As such all such ways limit severely the fidelity of the data and hence leads us to combine the best of all among them to be used in the CT Gan approach to generate our synthetic data.

We explore MED GAN for literature whereby it could provide more insight for us towards synthetic data. Medical image analysis's "new frontier" of picture-to-image translation is thought to have a wide range of potential applications. A significant fraction of contemporary techniques, however, provide customized solutions based on task-specific structures or call for improvement through non-end-to-end training. It offers a novel framework for medical image-to-image translation called MedGAN, which works end-to-end at the picture level. By combining the adversarial framework with a novel assortment of non-adversarial losses, MedGAN expands on recent developments in the field of generative adversarial networks (GANs). To penalize the difference between the translated medical images and the required modalities, it employs a discriminator network as a trainable feature extractor. The textures and fine structures of the desired target images are also matched to the translated images using style-transfer losses.

Next, we look at H-GAN or HandGAN which is a multi-scale perceptual discriminator-based cycle-consistent adversarial learning technique. It is intended to convert synthetic hand images to the actual world. Though synthetic hands offer accurate ground truth annotations, they do not accurately reflect the target distribution of actual data (Sergiu Oprea, 2021).

Without jeopardizing the privacy of the original dataset, we use PATE-GAN to create the resulting model which can be used to generate synthetic data on algorithms that can be trained and verified. The Private Aggregation of Teacher Ensembles (PATE) framework is modified for use with GANs by our methodology. It can strongly limit the impact of every one sample on the model using our modified framework, which is referred to as PATE-GAN. This leads to tight differential privacy guarantees and, consequently, an improvement in performance over models with the same guarantees. We also take a fresh look at how to evaluate the quality of synthetic data. It uses two algorithms' relative

performance on the synthetic dataset that should be the same as that of the algorithms' relative performance on the original dataset.

Table-GAN creates synthetic tables that are statistically identical to the real table while avoiding information leakage by using generative adversarial networks (GANs). It demonstrates that the performance of machine learning models trained using synthetic tables is comparable to models learned using the genuine table for unidentified testing situations. This characteristic is known as model compatibility. Methods for anonymization, perturbation, and synthesis without model compatibility are not extremely useful (Noseong Park, 2018).

Our analysis of literature leads us to conditional Generative Adversarial Nets from a paper of Mehdi and Simon (Mehdi Mirza, 2014) that constructed the generator and discriminator by feeding the data into a variable for the generation of MNIST digits. They Illustrated a model that could be used for in a multi modal model system as well as provided image tagging applications that could tag images through training labels. The use of Multimodal and unimodal data sets from MNIST digit generation to use of images from Flickr in this experiment provided insight for this report in tackling our issue of the variety of types of labels that we had to use in both of our datasets.

Since our data set faced challenges such as a mix of discrete and continuous variables along with generation of tabular data that was realistic, the paper by Lei Xu was one of the most reviewed by us to tackle. The paper highlights the challenges and states the CTGAN as a synthetic tabular data generator that can address several issues mentioned above. It can outperform all methods to date and surpasses Bayesian networks. They further challenge it on auto encoder for mixed tabular data synthesis whereby using VAE's to directly build the generator which showed exemplary evidence into success of CTGAN (Lei Xu, 2019).

One of the main literatures that we used was the modality bank (Qi Chang, 2022) whereby their main aim remained creation of synthetic data of model images in the medical data on healthcare data centers without any sharing of medical data. It focused on creating a set of effective domain specific modulation parameters that could be used across the domain. They also focused on the missing modality of data which in our datasets are the missing values in the tabular data. They were also able to synthesize multi modal samples that could achieve high performance and accuracy by learning from one data center and achieve close to actual images across the others.

Their research iterated creation of one domain specific modulation bank, a central generator, and multiple discriminators. The domain specific bank would treat all incoming data for privacy concerns by setting up layers of treatment above it such that privacy laws could be upheld. The convolutional kernel being used in this bank would also be used in the central generator whereby it could decode the data for synthesizing, and this would act as a key for the data. They experimented with three different datacenters that were set up on different settings along with three different datasets. This experiment shown they were able to learn the distribution and generate multi modal images across the datasets. They were also able to deal with missing data in the modality datasets using their generators (Qi Chang, 2022).

Traditionally a GAN is fed with a sample that trains together with the discriminator or a critic neural network that maps the sample vector into a distribution of data. This method however falls short in our case since it does not account for imbalance between the categorical data. Equal representation of the whole data into the sample fails for us and hence a simple GAN could no longer be sufficient for us. This class imbalance problem among the data is corrected by using a conditional GAN such that the sample would contain all the categories from discrete data evenly. This method is built upon the

basic GAN's by following three key elements into its structure of adoption such that building a conditional vector, creating a generator loss as well as training by sampling method. Thus, the CTGAN model is created that can deal with discrete and continuous categories both all the while generating close to accurate synthetic rows.

Our Literature review also led us to understand that the evaluation of the GAN's cannot be treated in a straightforward manner and would require precision analysis and adoption of some new metrics to realize the accuracy of our synthetic data. The use of likelihood fitness metric as well as machine learning efficacy to test accuracy was to be adopted as the main approach in this report.

9. DATASET DESCRIPTION – LOAN DATASET

The original dataset at hand provided by NayaOne consists of 887,349 observations, and 79 unique columns. The dataset provides information on the Loans given out by Lending Club in the US from 2007 to 2015. The variables present in the dataset are both numerical and categorical in nature, providing information on the loan issued itself but also on the customer who obtained that loan.

DATA PRE-PROCESSING:

Dropping Irrelevant and invaluable Columns:

By inspecting the original form of the dataset which can be found in figure 1 in the appendix, we deemed appropriate to drop columns irrelevant to the business problem. These columns included the id, member id, loan description, URL, employee title and title. For example, URL will not provide NayaOne with intuitive information regarding the loan issued and will not help the company's employees in formulating results that will be of use to its customers. One more reason that some columns were dropped was due to the large number of distinct values present. Since a discrete variable is a categorical variable, it was not possible to have it present as the CTGAN's efficiency is decreased when working with columns containing a vast number of categories. In this instance, the employee title contained 299271 unique entries. This was done to decrease the dataset's complexity and allow for a smooth fitment of the CTGAN. One more reason for dropping this specific column was because employment length and annual income will be included in the resulting dataset, which is a metric explaining variations in ability to repay a loan almost as efficiently as employee title. Employment length and Annual income shows your ability to repay a loan because it underlines a stable income and sustainability.

Missing Values and Data Imputation:

After an initial inspection of the dataset, we found out that many columns had missing values. We proceeded by imputing the columns that were missing a reasonable proportion of their data (Threshold of missing values: 100000 observations need to be present) and dropped the columns missing a very large portion of their data. Examples of such columns were: 'open_acc_6m' and 'open_il_6m' which only had approximately 20000 entries. The reason for this initial modification in the dataset is that the machine learning algorithm, CTGAN, we will fit later the data, does not work when null values are present. This can be considered one of the limiting factors of the CTGAN algorithm.

After dropping the above columns, the dataset consisted of 43 columns with only 3 exhibiting many missing values. The columns in question were, 'mths_since_last_delinq', 'mths_since_last_record' and 'mths_since_last_major_derog'. After a closer inspection in the given variables, it was understood that the null values present were because the observation in question did not have the associated entry present. For example, if an observation's 'months since last record' column was null, then we imputed the column based on the assumption that no past record was present.

The result, was to create 3 categorical variables based on the three columns above consisting of 6 distinct categories:

- *No_Associated_Entry_Present*
- <6 Months
- <12 Months
- 1-2 Years
- 2-4 Years

- 4-8 Years
- 8+ Years

In addition to the above categorical variables, we created another categorical variable consisting of the address state column. After inspecting the address state column, we concluded that it contained 52 distinct values. We proceeded by grouping the associated state entries by using greater regions in the United States. The result was to create one more categorical variable consisting of 9 distinct categories:

- East North Central
- West North Central
- East South Central
- West South Central
- Pacific
- Middle Atlantic
- South Atlantic
- New England
- Mountain

The remaining null values were dropped, and the final form of the dataset contains 773,968 entries and be seen in Figure 2 of the appendix.

10. EXPLORATORY DATA ANALYSIS

1) Key Numerical Variables Summary Statistics:

	count	mean	std	min	25%	50%	75%	max
loan_amnt	887379.0	14755.264605	8435.455601	500.00	8000.000	13000.000000	20000.000000	3.500000e+04
installment	887379.0	436.717127	244.186593	15.67	260.705	382.550000	572.600000	1.445460e+03
int_rate	887379.0	13.246740	4.381867	5.32	9.990	12.990000	16.200000	2.899000e+01
annual_inc	887375.0	75027.587761	64698.300142	0.00	45000.000	65000.000000	90000.000000	9.500000e+06
total_pymnt	887379.0	7558.826684	7871.243336	0.00	1914.590	4894.999117	10616.814231	5.777758e+04

Table 2: Numerical Summary Statistics

2) Key Categorical Variables Summary Statistics:

	count	unique	top	freq
emp_length	842554	11	10+ years	291569
verification_status	887379	3	Source Verified	329558
loan_status	887379	10	Current	601779
US Area	887379	9	South Atlantic	181994
home_ownership	887379	6	MORTGAGE	443557

Table 3: Categorical Summary Statistics

3) Numerical and Categorical Variables distributions

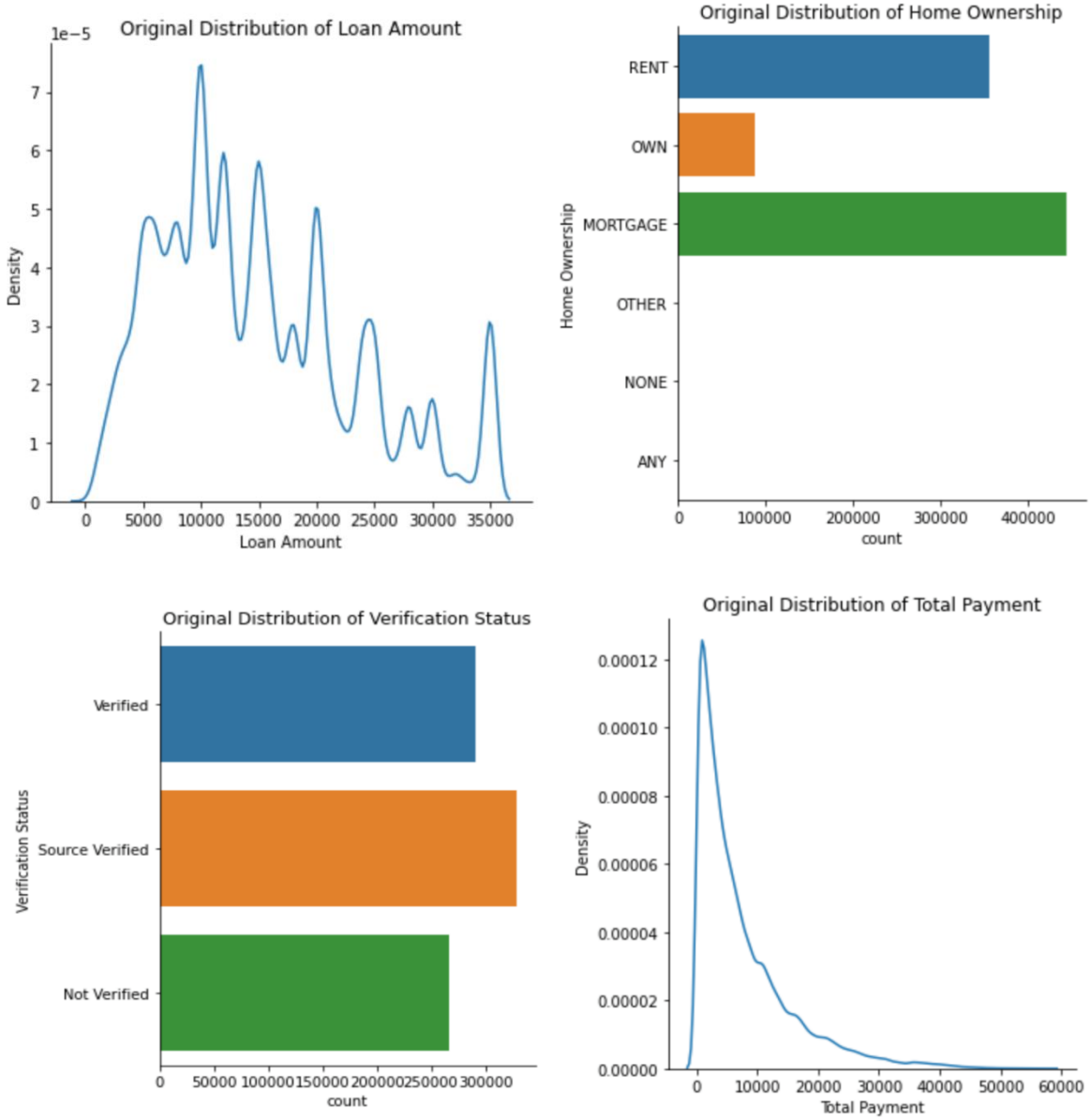
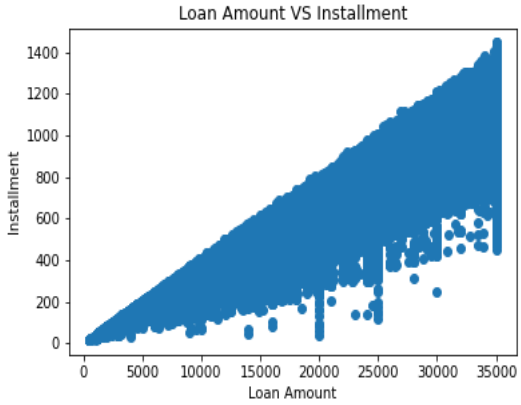


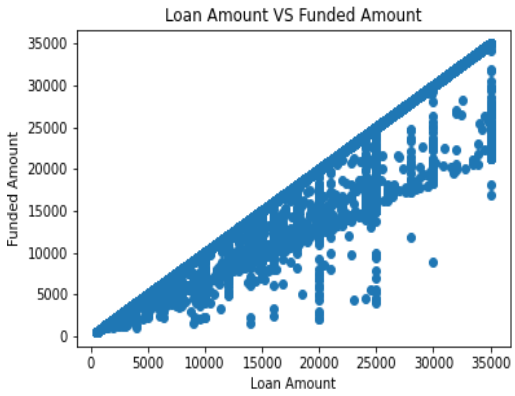
Figure 8: Numerical and Categorical Variables distributions

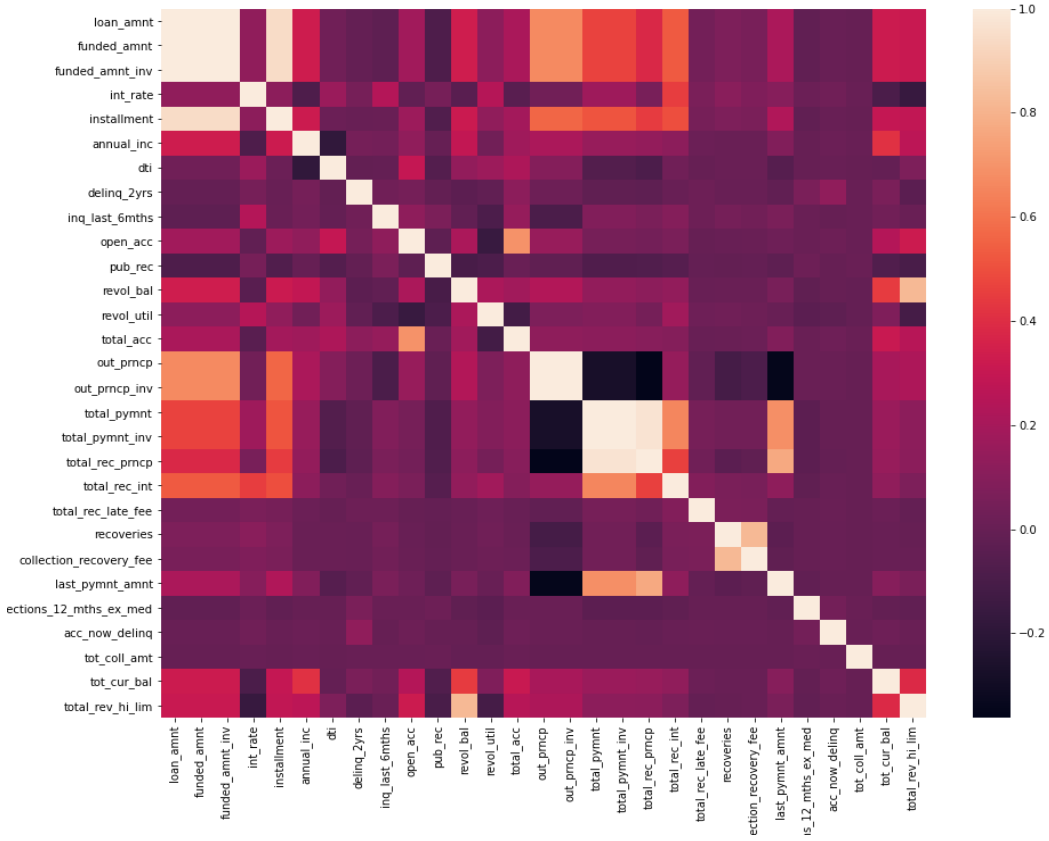
Description of above graphs and summary statistics:

- It can be seen in table 1 of summary statistics that the mean loan amount of the original distribution is 14755, while the mean of total payment is 7558. It can also be seen that the mean interest rate is equal to 13.2%.
- From table 2, it can be observed that most loans are currently being paid, thus the 'current' loan status and were issued in the South Atlantic area of the United States. It can also be observed that most of the individuals for which the loans are issued are either renting or have a pending mortgage on their place of living.
- In addition, by examining the distribution of total payment in figure 1, a positive skewness can be observed.

4) <u>Relationships and Correlation Matrix of Numerical Variables:</u>







Figures 9,10: Correlation matrix and relationships of original dataset

<u>Correlation Remarks:</u>

- Loan amount, funded amount invested, and funded amount are all positively and highly correlated with each other, exhibiting a positive correlation coefficient of 1. The same can be said for instalment and the Loan amount variables listed above. The relationships exhibited align with the framework of issuing Loans, as instalment are naturally higher for larger loan amounts.
- Total payment, total payment invested, and total_received_pmcp are all highly correlated with each other, exhibiting a positive correlation coefficient of 1.

11. CTGAN FITTING

SAMPLE CREATION:

For our data to be processed using CTGAN we needed to efficiently decrease its size without affecting its originality and initial correlations. The reason we needed to do that was due to the limited processing power available to the team during the time of the project. For this purpose, we used a method known as Simple Random Sample to sample two datasets: one being 0.1 fraction of the original one post cleaning (77384 entries) and the other being a 0.01 fraction of the original one post cleaning (7738 entries). The resulting samples were then evaluated using the summary statistics to ensure their representativeness as compared to the original, complete Loans dataset after cleaning exercises.

PERFORMANCE METRIC:

To be able to promptly assess the performance of the models to be fitted and to make informed conclusions about the fitting process, we decided to use the `evaluate()` function from `sdv.evaluation` library. This function takes two datasets as inputs and returns an accuracy score; a number between 0 and 1 that quantifies the similarity of the two tables provided (MIT Data To AI Lab, 2018). From every model that we train, we would generate a sample of 1000 entries of synthetic data and we would compare the 1000-row generated sample to the data that the model was trained on. Although the final evaluation of the synthetic data that we create will include a detailed comparison and discussion of distributions of specific variables, the `evaluate()` function will provide us a prompt insight for reasoning how the fitting process is progressing and whether the alterations to the model that we make (i.e., introduction of constraints or hyper parameters tuning) improve the expected output of the model (i.e., 1000-row sample of synthetic data).

TRAINING THE MODEL:

Having cleaned the dataset as explained above, we made the first attempt on fitting the model. To start with, we imported CTGAN from `sdv.tabular` and fitted the model to the smaller sample of data (7738 entries) on default tuned hyper parameters, i.e., `epochs = 300, batch_size = 500, generator_dim = (256, 256), discriminator_dim = (256, 256)`. In this report, this model will be referred to as Model 1. Having trained the model and generated a 1000-row synthetic data sample, the accuracy score computed by the `evaluate()` function was approximated to 0.676. As such, we considered it to be a fairly good result, indicating a 67.6% similarity of the generated dataset to the original one. Below figure is an extract from the generated sample.

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	sub_grade	emp_length	home_ownership	annual_inc
0	6479.0	5407.0	4918.0	36 months	20.18	211.08	A4	1 year	RENT	36038.65
1	1000.0	5988.0	3164.0	36 months	13.78	271.96	C1	8 years	RENT	71066.75
2	18606.0	22080.0	14355.0	60 months	11.25	908.64	B4	< 1 year	OWN	115161.24
3	15369.0	21478.0	10352.0	60 months	13.25	481.53	C1	10+ years	MORTGAGE	55868.26
4	16816.0	9909.0	9875.0	36 months	24.46	183.42	C2	6 years	RENT	118808.78

Figure 11 : 5-rows x 10-columns extract of data generated by Model 1

Despite good accuracy obtained in the first run of the Model 1, one significant flaw of the generated data was noted. Namely, Model 1 failed to reflect the relationship between variables `loan_amnt`, `funded_amnt` and`funded_amnt_inv` (see Correlation Remarks). As seen in the above table, in 4 out of 5 instances displayed `the`funded_amnt is larger than `loan_amnt`, which is not reasonable. Furthermore, in one `instance`funded_amnt_inv is larger than `funded_amnt`, which is, again, not reasonable. Hidden(Un-reflected) relationship also means that correlations between variables in the new dataset were far from original ones. Below table compares the obtained and desired correlations (in brackets).

Model 1	loan_amnt	funded_amnt	funded_amnt_inv
loan_amnt	1.00 (1.00)	0.503 (1.00)	0.535 (0.997)
funded_amnt	0.503 (1.00)	1.00 (1.00)	0.524 (0.998)
funded_amnt_inv	0.535 (0.997)	0.524 (0.998)	1.00 (1.00)

Table 4: Comparison of realized and expected (in brackets) correlation scores between variables `loan_amnt, funded_amnt` and `funded_amnt_inv` in the 1000-row sample generated by Model 1

As per the above argument, we found the usability of the synthetic data created by Model 1 highly limited. To improve it, we resorted to adding constraints to account for the desired relationship between `loan_amnt`, `funded_amnt` and `funded_amnt_inv`. From `sdv_constraints` we imported Inequality, function, which allowed us to formulate and add two constraints to the CTGAN setup:

1) `loan_amnt > funded_amnt`
2) `loan_amnt > funded_amnt_inv`

Having added the above constraints, we ran the model (referred to as Model 2) again on the same, default tuned hyper parameters, and sampled 1000 rows of generated data. The generated dataset reported an accuracy score of 0.610, which is lower than accuracy of the dataset of the same size generated by Model 1, which is unexpected. Despite decreased accuracy, the model seems to have captured the desired relationship between `loan_amnt`, `funded_amnt` and `funded_amnt_inv` quite well, and the correlation scores do reflect the same. In the extract below, it can be clearly seen that in all five instances `loan_amnt` and `funded_amnt` are in alignment, whilst `funded_amnt_inv` is slightly smaller, which is expected.

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	sub_grade	emp_length	home_ownership	annual_inc
0	25462.001070	25462.0	21875.0	60 months	20.04	664.75	D4	5 years	MORTGAGE	213007.23
1	18854.001667	18854.0	9586.0	36 months	14.66	580.23	D1	3 years	MORTGAGE	146405.11
2	33916.000605	33916.0	12169.0	36 months	10.98	291.96	C4	< 1 year	MORTGAGE	119925.69
3	3623.000477	3623.0	1061.0	36 months	10.62	39.03	B4	10+ years	RENT	48472.76
4	26914.000770	26914.0	16270.0	60 months	13.00	539.71	E1	6 years	MORTGAGE	171111.88

Figure 12: 5-row x 10-columns extract of data generated by Model 2

Model 2	loan_amnt	funded_amnt	funded_amnt_inv
loan_amnt	1.00 (1.00)	1.00 (1.00)	0.768 (0.997)
funded_amnt	1.00 (1.00)	1.00 (1.00)	0.768 (0.998)
funded_amnt_inv	0.768 (0.997)	0.768 (0.998)	1.00 (1.00)

Table 5: Comparison of realized and expected correlation scores between variables `loan_amnt, funded_amnt and funded_amnt_inv` in the 1000-row sample generated by Model 2

After careful consideration, we decided that there is potential for the model to be further improved by adding a third constraint to the set of constraints: `funded_amnt > funded_amnt_inv`. Having added the same, we ran the model (hereby referred to as Model 3) on the same, default tuned parameters, and sampled 1000 rows of generated data. The generated dataset reported an accuracy score of 0.636, which is noticeably higher than the score for Model 2 data, however, lower than the score from Model 1 data. Nonetheless, the desired correlations seem to have been achieved, being in almost full alignment with the expectations.

Model 3	loan_amnt	funded_amnt	funded_amnt_inv
loan_amnt	1.00 (1.00)	0.999 (1.00)	0.999 (0.997)
funded_amnt	0.999 (1.00)	1.00 (1.00)	1.00 (0.998)
funded_amnt_inv	0.999 (0.997)	1.00 (0.998)	1.00 (1.00)

Table 6: Comparison of realized and expected correlation scores between variables `loan_amnt, funded_amnt and funded_amnt_inv` in the 1000-row sample generated by Model 3

As it can be seen in the below table, the aligned correlations may have resulted from the fact that differences between values for the three variables for each row are very small. However, all the conditions stated are satisfied; funded_amntis smaller than loan_amnt, as well as funded_amnt_inv is smaller than both funded_amnt and loan_amnt. This stands in correspondence to the original dataset and allows us to deem the result as satisfactory.

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	sub_grade	emp_length	home_ownership	annual_inc
0	10600.019770	10600.002481	10600.0	36 months	26.12	236.51	B3	10+ years	RENT	196255.86
1	10765.012578	10765.001721	10765.0	60 months	19.67	177.88	A3	10+ years	RENT	132106.80
2	21164.026003	21164.004027	21164.0	60 months	26.53	336.10	E1	10+ years	RENT	128227.23
3	1000.001762	1000.000475	1000.0	36 months	15.53	98.06	C3	10+ years	RENT	48085.13
4	21095.009384	21095.006604	21095.0	60 months	26.47	270.66	A5	< 1 year	RENT	54496.93

Figure 13: 5-rows x 10-columns extract of data generated by Model 3

To further increase the accuracy of the generated samples, we resorted to *hyperparameters* tuning, and ran Model 4, Model 5, and Model 6 with below specifications. Since results achieved with default

number of epochs and batch size were relatively satisfactory, however, time consuming and computationally demanding, we preferred trying to decrease the hyperparameters to optimize on computation time and power required for fitting the data. As such, the following results were obtained.

	Model 3	**Model 4**	**Model 5**	**Model 6**
Parameters	`epochs = 300` (default), `batch_size = 500` (default)	`epochs = 200, batch_size = 500` (default)	`epochs = 200, batch_size = 100`	`epochs = 100, batch_size = 100`
Accuracy score	0.636	0.585	0.603	0.575

Table 7: Comparison of accuracy scores reported by 1000-rows samples generated by Models 3-6.

As seen in the above table, data samples generated by Models 4, 5 and 6 reported noticeably lower accuracy scores than Model 3. Given the highest accuracy score is obtained by a 1000-row sample generated by Model 3, which is the model with default hyper parameters (`epochs = 300 and batch_size = 500`), we have concluded that this is the optimal specification for the CTGAN model to generate synthetic data from Loans dataset in question. It is worth emphasizing, however, that the accuracy reported by sample generated by Model 3, did not increase; rather slightly decreased, as compared to sample generated by Model 1, despite a crucial correction that was the establishment of the three constraints.

However, it is worth noting that Models 1-6 were calibrated on a small extract of data which constituted a 0.01 fraction of the original dataset. The original dataset contained 887,349 observations, out of which Models 1-6 were trained only on 7738 observations. Having found Model 3 to be best among the ones we trained using the small extract of the original dataset, we decided to retrain it on the second, larger extract of the original dataset (77384 observations) and generate a 1000-row sample to compare the results that the same model returns when being trained on datasets of different size. As reported in the table below, Model 3 trained on a larger extract of original data did generate synthetic data of a higher accuracy score. This leads us to believe that if we had been able to train the model on the cleaned version of the entire original dataset, the accuracy score achieved could have yet been higher, hence the generated data could have been much more like the original observations. However, the computational burden of running the model on larger data extract did not justify the benefit of increased accuracy scores.

	Model 3 trained on 7738 observations	Model 3 trained on 77384 observations
Accuracy Score	0.636	0.648

Table 8: Comparison of accuracy scores reported by 1000-rows samples generated by Model 3, calibrated on smaller and larger data extracts

12. EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES

Comparison and Evaluation of Synthetic and Original Sample

To evaluate how well the synthetic sample generated from model 3 matches the original sample used to fit the CTGAN, we decided to look at key summary statistics of both, along with some key correlations of numerical variables.

	Original Sample				**Synthetic Sample**			
Variables	Mean	Min	Max	St.Dev	Mean	Min	Max	St.Dev
Loan Amount	15149	1000	35000	8436	16014	1000	35158	8724
Instalment	447	30.1	1424.5	243.6	490	30	1338	278.2
Interest Rate	13.3	5.3	28.9	4.41	14.3	5.32	26.2	4.1
Annual Income	77145	8400	9500000	77993	75246	12173	467379	52503
Total Payment	7146	0	53851	7586	10621	0	49027	10450

Table 9: Original and Synthetic Sample Numerical Summary Statistics

	Original Sample			**Synthetic Sample**		
Variables	Mode	Frequency	Mode%	Mode	Frequency	Mode%
Emp_Length	10+ Years	27499	35.5%	10+ Years	1072	26.8%
Ver_Status	Verified	30702	39.6%	Verified	1780	44.5%
Loan_Status	Current	56885	73.5%	Current	2403	60.0%
US_Area	S.Atlantic	15807	20.4%	M.Atlantic	1464	36.6%
Home_Ownership	Mortgage	39289	50.7%	Mortgage	2251	56.2%

Table 10: Original and Synthetic Sample Categorical Summary Statistics

As it can be seen in table x above, the CTGAN was successful in capturing the distributions of most variables, with means in both samples being close to each other. One of the limitations observed in the above statistics, is the inability of CTGAN to capture outliers in the annual income variable. In addition, the total payment mean value calculated in the synthetic sample is 50% higher than the mean total payment in the original Sample. This is a result of the abnormally skewed distribution of the total payment variable, which was discussed in the exploratory analysis carried out above.

This is evident by comparing the maximum value of annual income in the original sample (=9500000) and in the synthetic sample (=467379). The maximum value of annual income is much smaller in the synthetic sample than in the original sample.

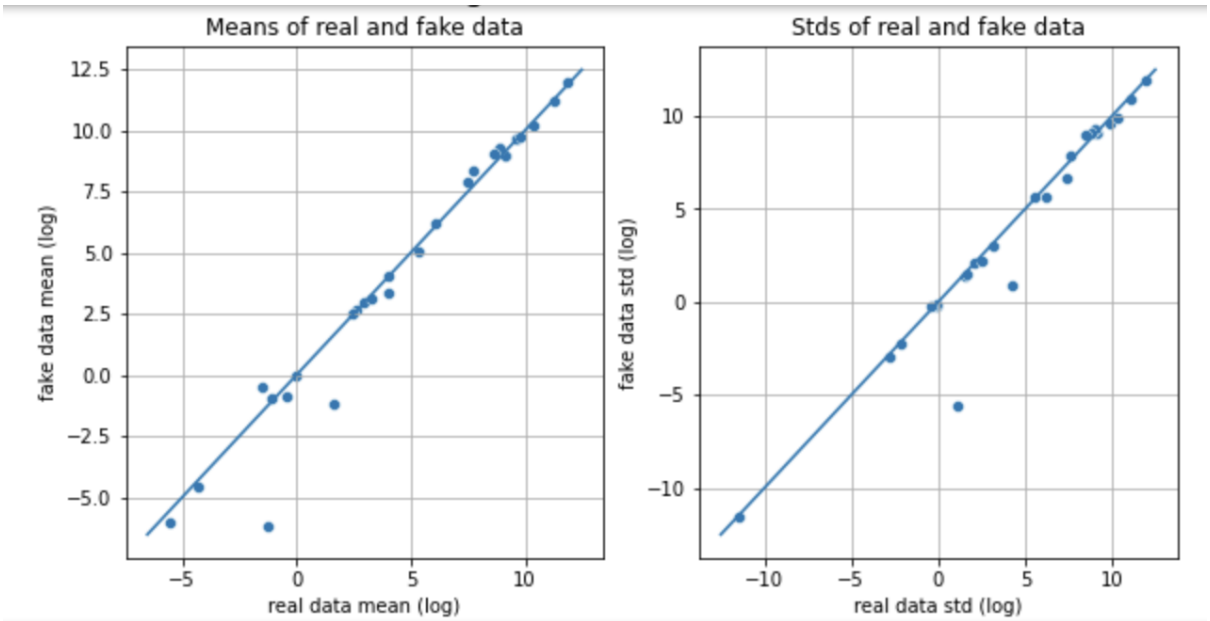


Figure 14: Absolute Log Mean and STDs of Numeric Data

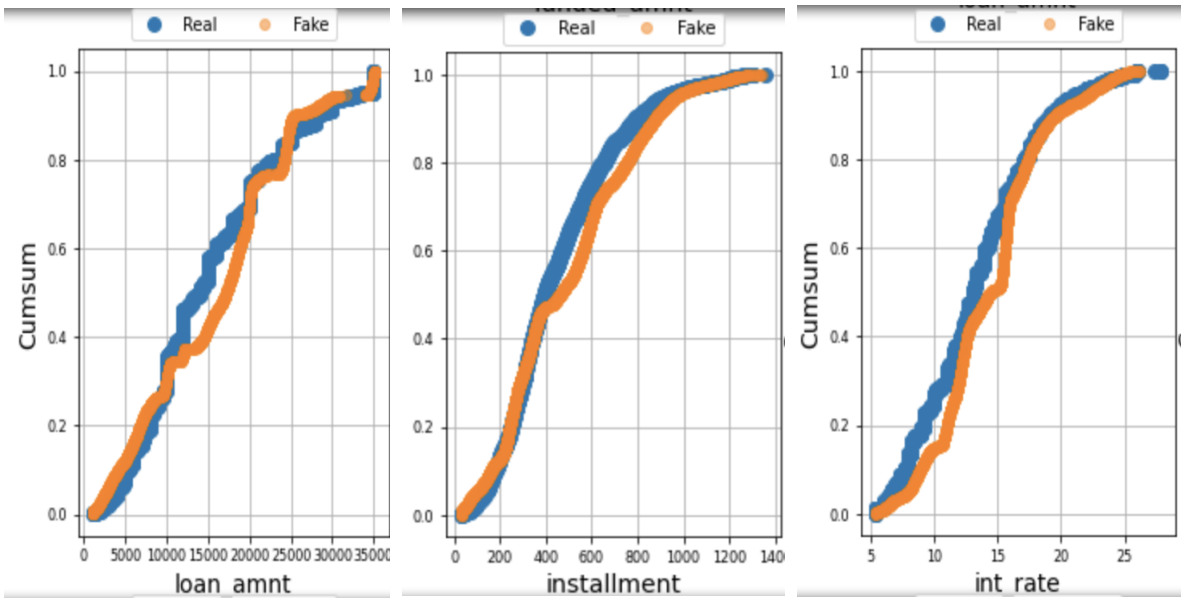


Figure 15: Cum-Sum of Real and Fake Samples: Real/Fake

As it can be seen from the above figures the fake sample was able to capture the numerical variable distributions of the original sample. It can be argued that by increasing the size of the dataset used in the fitting of CTGAN, the smoothness of the fake data cumulative sum lines will increase and be able to depict real distributions more accurately.

Comparison of Correlation Matrices:

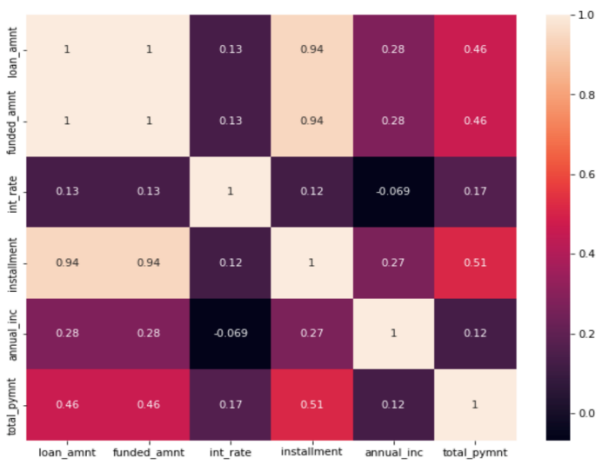


Figure 16: Original Sample Correlation

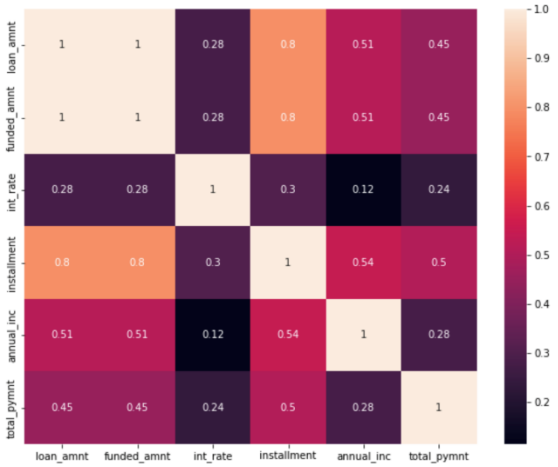


Figure 17: Synthetic Sample Correlation

The correlations of key numerical variables, seem to hold in the synthetic data generated. Through the above correlation, it can be deduced that the introduction of adequate constraints in model 3, not only helped capture the correlation of Loan amount and Funded amount but were also useful in capturing the correlation between instalment and loan amount. (Original Correlation = 0.94, Synthetic Correlation= 0.8).

How well can synthetic data generated be used to classify status of Loans:

By using the table-evaluator package in python, we were able to determine the accuracy of classification of Loan Status, when using the synthetic data generated, in comparison to the original sample at hand. Loan status was chosen between other variables since is the most important and insightful variable given the business case at hand. By comparing the F-test scores of both synthetic and original samples, we were able to determine that the synthetic data can classify loan status, almost as efficiently as the original data. The results of the classifier can be seen in figure x in the appendix. Another metric used to compare the two samples was the Jaccard Similarity index. Jaccard Similarity is a statistic varying from 0 to 1, used to compare the similarity of two datasets. The higher it is, the more similar the two samples are. In this case the Jaccard Similarity obtained, underlined a high similarity between the two datasets. (as high as 0.8980)

13. 1st DATASET CONCLUSION AND EVALUATION LIMITATIONS

- Despite introduction of the constraints, which factually made the generated observations for variables loan_amnt, funded_amnt and funded_amnt_inv appropriate, data generated by Model 3 did report a lower accuracy score as compared to data generated by Model 1. Hence, by looking only at accuracy scores, Model 1 could have been mistakenly considered as good as Model 3, whilst Model 1 in fact generated not representative data. Further to this argument, introduction of two constraints in Model 2 substantially decreased the accuracy score of the generated data as compared to Model 1, which is not reasonable. Therefore, the accuracy score cannot be used as a standalone metric to assess the performance of the models.

- For a model to be capable of generating samples of high similarity to the original data, it should be trained on as many data points as possible. However, CTGAN, and the neural networks within it, is a very computationally demanding structure, and as discussed above, we must have sampled an extract of the whole dataset to be able to efficiently fit the model. This compromises the similarity of the generated dataset as compared to the original one. As seen in table 5-7, increased number of observations for training does increase the accuracy score reported by the generated data, which leads us to believe that if we had been able to train the model on the entire loan's dataset, we would have been able to achieve a higher score. Nevertheless, given technical constraints, this was infeasible.

- The accuracy score of 0.636 reported by data generated by Model 3 may be considered relatively satisfactory, but it still indicates at most a moderate similarity of the two datasets. As such, real life useability of a synthetic dataset with such an accuracy score is limited since the targeted user must account for potential differences and trends that have not been captured by the model.

- It could be argued that low accuracy scores obtained by the generated samples were because of their small size (1000 rows). To validate this claim, we generated a sample of 4000 rows from retrained Model 3 and compared it to the original data. This was done to see how the accuracy score changes with sample size of synthetic data. Surprisingly, we found the accuracy score of the larger sample to be 0.610, noticeably lower than for the smaller 1000-row sample. This finding made us conclude that the effect of the synthetic sample size on the accuracy score is nontrivial and should have been considered more carefully throughout this project. This is certainly another limitation to the usage of the fitted models.

- As part of data cleaning exercises, we dropped the column containing employee titles. This was a simplification we introduced, as it contained thousands of unique values, and an appropriate generation of synthetic data for this variable would be computationally challenging. It is reasonable, since the titles would differ between employers, regions, seniority etc. For further model fitting, we could group the values for this variable such that, for instance, similar employee titles, or employee titles of similar salaries would be grouped together. This way the model would have to generate only the category for each observation and such an alteration could potentially allow for relevant and appropriate generation of synthetic data containing information on the employee title.

14. DATASET DESCRIPTION – COMPLAINT DATASET

We loaded the Necessary Libraries and the read csv () function to import the consumer complaints dataset. Using the info () function, we extracted information about the dataset such as its dimensions, 555957 * 18, and the total number of columns and entries, 555,957, for the consumer complaint data set.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 555957 entries, 0 to 555956
Data columns (total 18 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   date_received                 555957 non-null  object
 1   product                       555957 non-null  object
 2   sub_product                   397635 non-null  object
 3   issue                         555957 non-null  object
 4   sub_issue                     212622 non-null  object
 5   consumer_complaint_narrative  66806 non-null   object
 6   company_public_response       85124 non-null   object
 7   company                       555957 non-null  object
 8   state                         551070 non-null  object
 9   zipcode                       551452 non-null  object
 10  tags                          77959 non-null   object
 11  consumer_consent_provided     123458 non-null  object
 12  submitted_via                 555957 non-null  object
 13  date_sent_to_company          555957 non-null  object
 14  company_response_to_consumer  555957 non-null  object
 15  timely_response               555957 non-null  object
 16  consumer_disputed?            555957 non-null  object
 17  complaint_id                  555957 non-null  int64
dtypes: int64(1), object(17)
memory usage: 76.3+ MB
```

Figure 18: Complaint data frame information

	date_received	product	sub_product	issue	sub_issue	consumer_complaint_narrative	company_public_response	company	state	zipcode	tags	consumer_consent_p
0	08/30/2013	Mortgage	Other mortgage	Loan modification,collection,foreclosure	NaN	NaN	NaN	U.S. Bancorp	CA	95993	NaN	
1	08/30/2013	Mortgage	Other mortgage	Loan servicing, payments, escrow account	NaN	NaN	NaN	Wells Fargo & Company	CA	91104	NaN	
2	08/30/2013	Credit reporting	NaN	Incorrect information on credit report	Account status	NaN	NaN	Wells Fargo & Company	NY	11764	NaN	
3	08/30/2013	Student loan	Non-federal student loan	Repaying your loan	Repaying your loan	NaN	NaN	Navient Solutions, Inc.	MD	21402	NaN	
4	08/30/2013	Debt collection	Credit card	False statements or representation	Attempted to collect wrong amount	NaN	NaN	Resurgent Capital Services L.P.	GA	30106	NaN	

Figure 19: Complaint dataset head

15. DATA PRE-PROCESSING

Except for complaint id, which has int64 data, the remaining 17 features have object datatypes. Following an examination of the dataset's data, the following modifications were made:

- From the information table, characteristics such as sub product, sub issue, consumer complaint narrative, company public response, tags, and consumer consent provided contain few non-null values, and therefore missing data imputation is not the option for dealing with these missing values. Therefore, we decided to eliminate these features entirely.
- In addition to these attributes, complaint id is a unique value column that is not relevant when exploring data, training a model, or producing synthetic data, so we eliminated it. Similarly, Zip code can be eliminated because it will not provide any value. Date columns such as date received and date sent to company can also be omitted, as date values are not required for our synthetic data generation.
- The features timely response and consumer disputed are binary and give little training support. After this, just five characteristics remained.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 555957 entries, 0 to 555956
Data columns (total 5 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   product                       555957 non-null  object
 1   issue                         555957 non-null  object
 2   state                         551070 non-null  object
 3   submitted_via                 555957 non-null  object
 4   company_response_to_consumer  555957 non-null  object
dtypes: object(5)
memory usage: 21.2+ MB
```

Figure 20: Complaint data frame information after dropping features

	product	issue	state	submitted_via	company_response_to_consumer
0	Mortgage	Loan modification,collection,foreclosure	CA	Referral	Closed with explanation
1	Mortgage	Loan servicing, payments, escrow account	CA	Referral	Closed with explanation
2	Credit reporting	Incorrect information on credit report	NY	Postal mail	Closed with explanation
3	Student loan	Repaying your loan	MD	Email	Closed with explanation
4	Debt collection	False statements or representation	GA	Web	Closed with explanation

Figure 21: Complaint dataset head after dropping features

16. EXPLORATORY ANALYSIS

<u>Analysing text statistics:</u>
EDA was utilized to familiarize ourselves with our newly censored dataset using descriptive statistics and data visualizations.

First, we selected the product attribute that has no null values, and based on the data, the bulk of products with consumer complaints are mortgages (Fig 22), which are nearly double the second-

placed product, debt collection. This is followed by satisfaction with credit reporting, credit cards, and bank account servicing, indicating that consumers are typically dissatisfied with these goods.

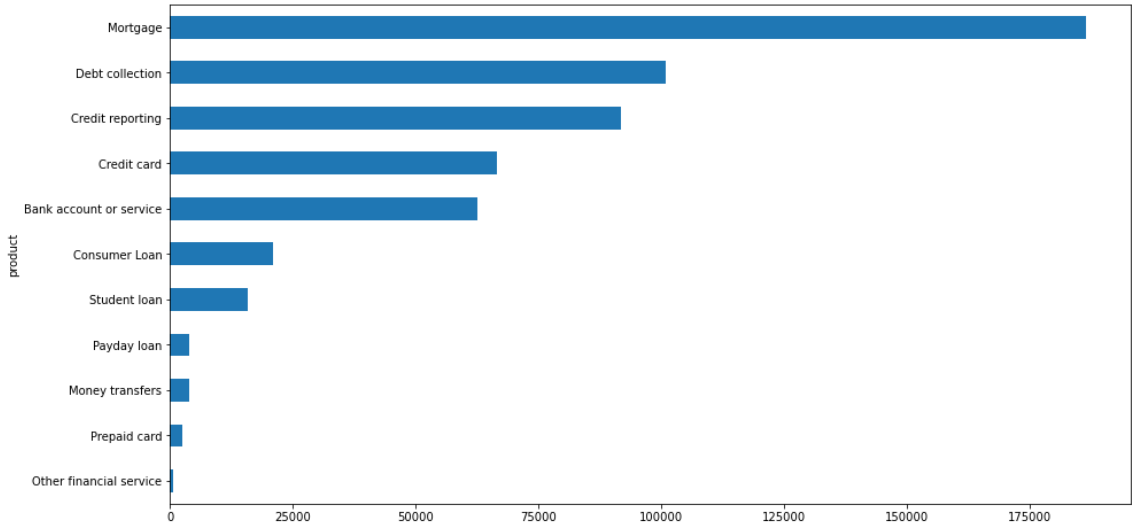


Figure 22: Horizontal bar plot of product frequency in complaints

Second, the company response to consumer function may inform us how the business is responding to all its complaints. Using a pie chart, we determined that 72.7% of the complaints were "Closed with an explanation" and 12.6% were "Closed with monetary relief".

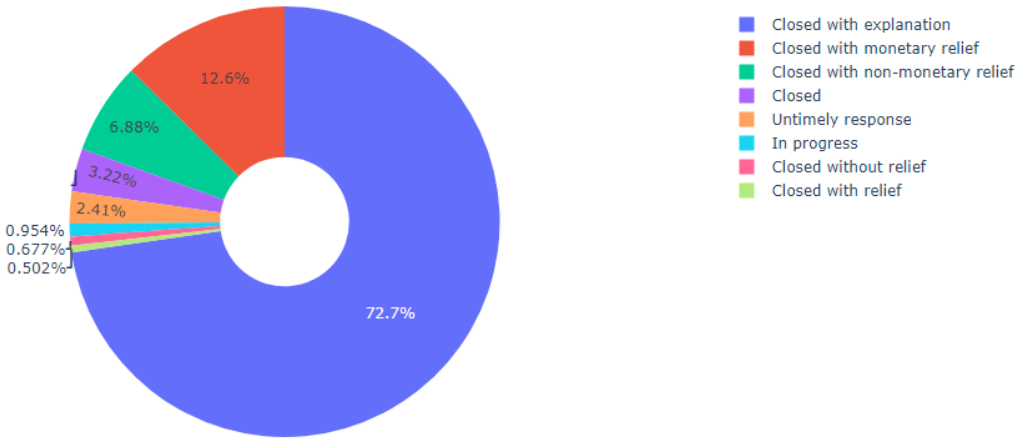


Figure 23: Pie chart of company_response_to_consumer

Even after removing features with null values, there remained a few variables with null values, therefore we chose to remove all null-valued rows from the dataset. After that, the dimensions of our dataset changed to 551070 by 5. (Fig 24)

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 551070 entries, 0 to 553096
Data columns (total 5 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   product                       551070 non-null  object
 1   issue                         551070 non-null  object
 2   state                         551070 non-null  object
 3   submitted_via                 551070 non-null  object
 4   company_response_to_consumer  551070 non-null  object
dtypes: object(5)
memory usage: 25.2+ MB
```

Figure 24: Complaint data frame information after dropping null value rows

<u>Python Text Data Cleaning:</u>

Typically, text data contains a great deal of noise in the form of symbols, punctuation, and stop words. Therefore, it is vital to clean up the text, not only to make it more comprehensible but also to gain deeper insights. Here, our dataset has six columns, one of which ('problem') contains textual data. After examining the data closely, we detect contractions such as "can't" and punctuation such as "," etc. and "." in the issues. These will be handled by completing the following operations:

- Expand contractions
- Lowercase the reviews
- Remove punctuations

Expand contractions: In the dataset, we find contractions such as 'can't' and 'aren't'; as these are abbreviated versions of words, we extend them to 'cannot' and 'are not', respectively. We've constructed a dictionary of typical English contractions to map the contractions so that they can be expanded for a better analysis.

Lowercase the reviews: Dataset contains 'loan' and 'Loan', which are considered differently; consequently, we lowercase the words to circumvent this issue. Therefore, the lower() method was used to transform text to lowercase.

Remove punctuations: There are punctuation marks such as commas, hyphens, and periods in the dataset. They must be eliminated, even though they are essential for English grammar but not for text analysis. Here, the function'string.punctuations' holds all punctuation marks, and we utilise regular expressions to look for and delete them from the text.

Checking the head of the dataset reveals that a new column has been generated because of the cleaning of columns issue. In the new column, we can see that all punctuation marks have been removed, all words have been changed to lowercase, and contractions have been extended.

	product	issue	state	submitted_via	company_response_to_consumer	cleaned
0	Mortgage	Loan modification,collection,foreclosure	CA	Referral	Closed with explanation	loan modificationcollectionforeclosure
1	Mortgage	Loan servicing, payments, escrow account	CA	Referral	Closed with explanation	loan servicing payments escrow account
2	Credit reporting	Incorrect information on credit report	NY	Postal mail	Closed with explanation	incorrect information on credit report
3	Student loan	Repaying your loan	MD	Email	Closed with explanation	repaying your loan
4	Debt collection	False statements or representation	GA	Web	Closed with explanation	false statements or representation

Figure 25: Complaint dataset head with new column ”cleaned”

17. SENTIMENT ANALYSIS

To examine the complaint's tone, we must divide the data sentences into word levels and determine which concern is most prevalent. First, we can eliminate the English stop words, which are the most frequently used words in any language, such as "the," "a," and "an." Analyzing the quantity and variety of stop words can provide valuable insights into the data. Using the nltk package, you may obtain a corpus containing stop words.(Fig 26)

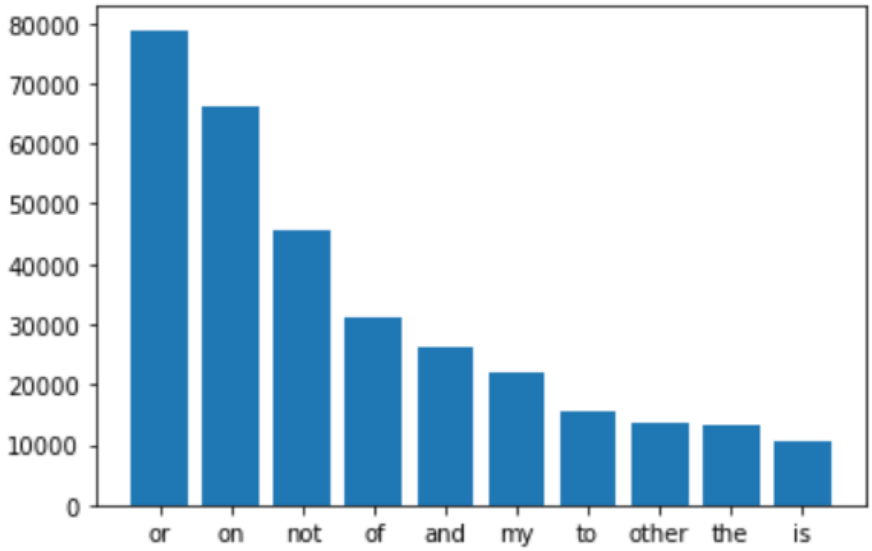


Figure 26: Bar plot for top stop words

Stop words such as "or," "on," and "not" are prevalent in the issue. Now that we know which stop words appear frequently in our text, let us examine which additional words appear frequently. We utilized the collections library's counter function to count and record the occurrences of each word in a list of tuples. This is an especially important function in natural language processing when dealing with word-level analysis. (Fig 27)

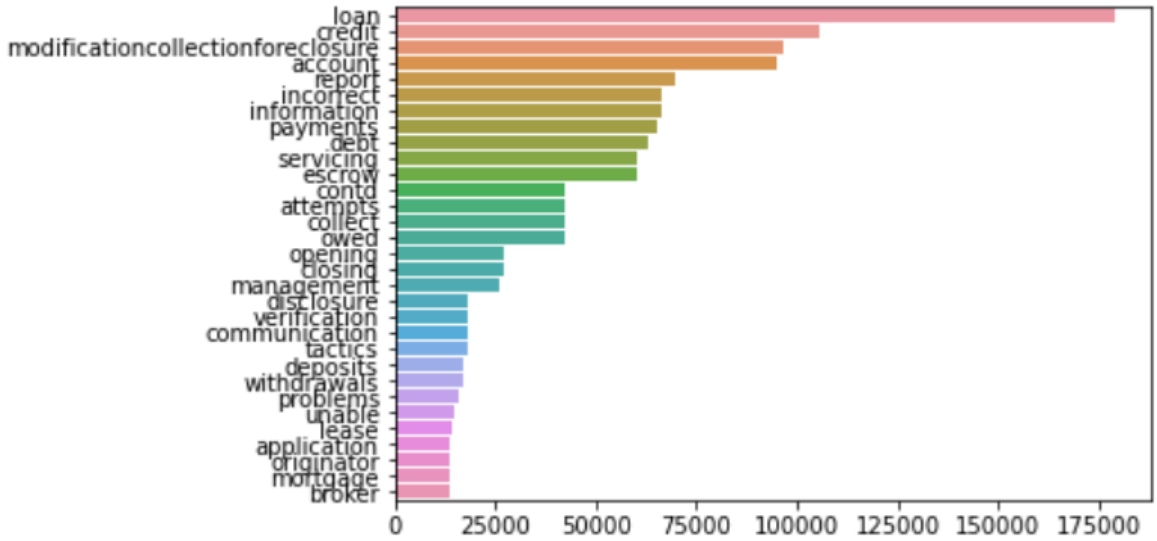


Figure 27: Vertical Bar plot for top words

The terms "loan", "credit", "modification collection foreclosure", and "account" predominate among the customer's concerns. After performing a word-level analysis, we wanted to analyze contiguous sequences of n words. To accomplish this, we used the ngrams function from the nltk.util package to perform Ngram exploration. (Fig 28)

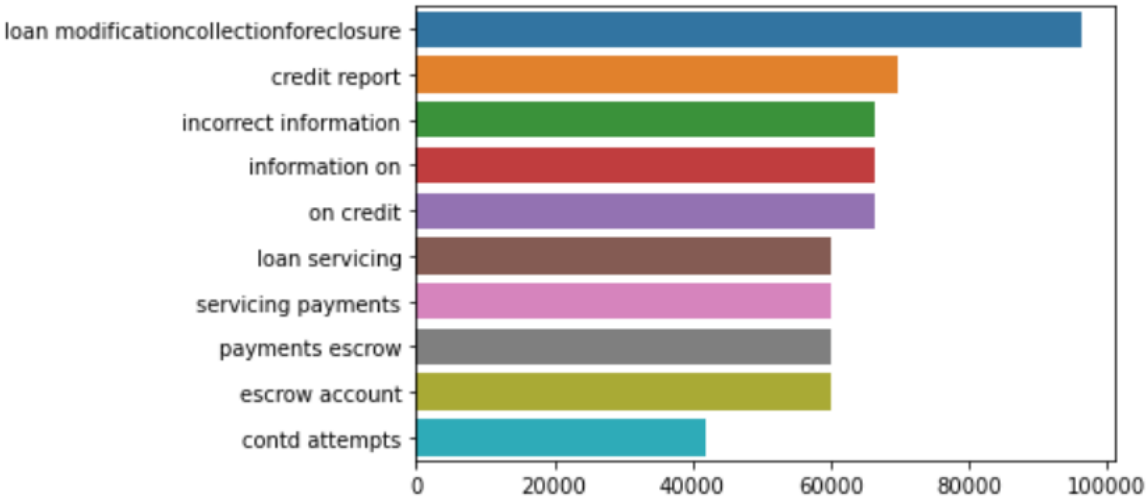


Figure 28: Vertical Bar plot for top bigrams

From the bigram bar plot, maximum consumer raises complaints for a loan modification, collection, and foreclosure after which its credit report.(Fig 29)

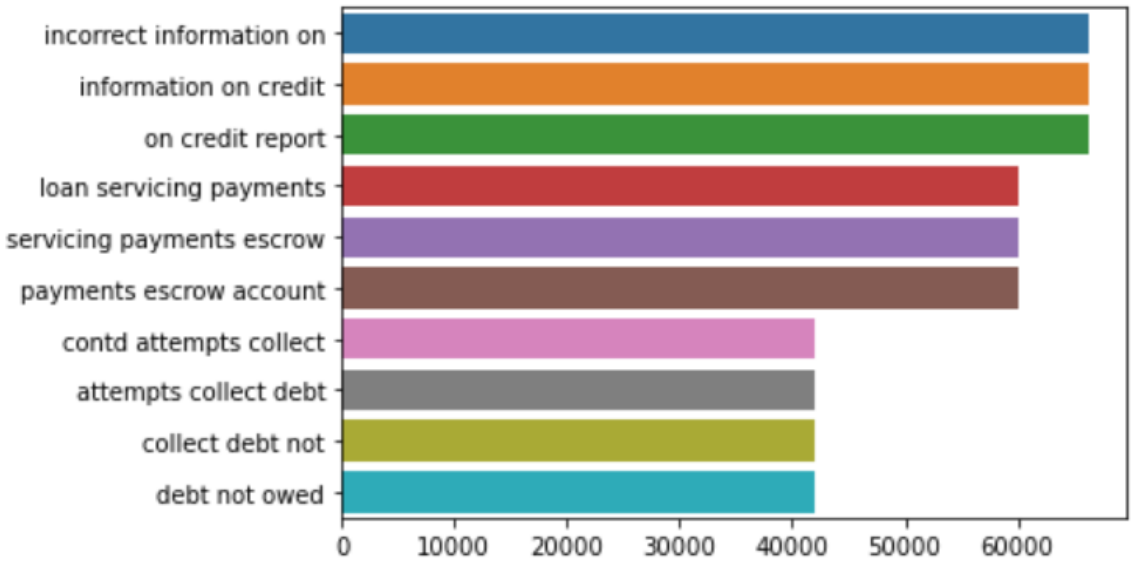


Figure 29: Vertical Bar plot for top trigram

Many of these trigrams consist of combinations or subparts of "credit information", "servicing payments escrow", and "debt collection efforts". It implies that we should try to sanitize our data. So, for further investigation, we will conduct the following:

- Stop words Removal
- Lemmatization
- Create Document Term Matrix

As stop words provide little to no value, these values are deleted to reduce the size of the dataset and enhance the emphasis on significant words. Following this, we performed lemmatization, which consisted of converting a word to its root form. Here, SaCy was employed for stop word removal and lemmatization.

product	lemmatized
Bank account or service	deposit withdrawal deposit withdrawal makingre...
Consumer Loan	manage loan lease manage loan lease manage loa...
Credit card	application processing delay credit line incre...
Credit reporting	incorrect information credit report incorrect ...
Debt collection	false statement representation contd attempt c...

Figure 30: Product with lemmatized column

It can be seen(Fig 30), a new lemmatized column is formed, with which we created a Document Term Matrix. Here, Document Term Matrix provides the frequency of a word in a corpus, and from the (Fig 31), we can determine the frequency of various terms.

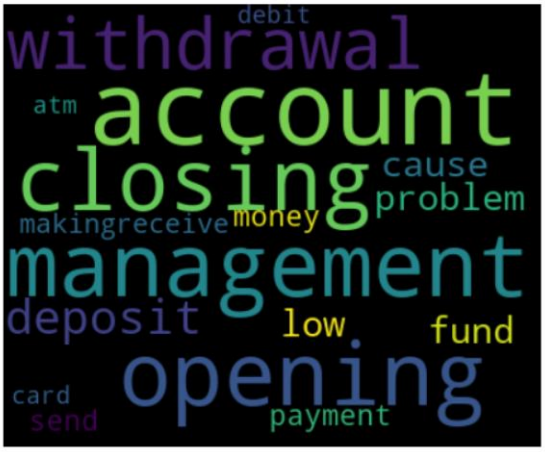
product	account	acct	action	add	advance	advertising	amt	application	apply	apr	...	unauthorized	underwritin
Bank account or service	26046	0	0	0	0	0	0	0	0	0	...	0	
Consumer Loan	439	47	0	0	0	0	4	0	34	0	...	0	
Credit card	7323	0	0	0	348	2179	0	387	0	4581	...	0	

3 rows × 154 columns

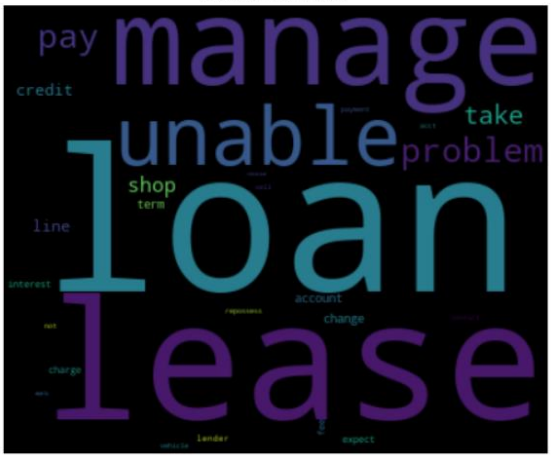
Figure 31: Document Term Matrix of dataset

In this phase, we uncovered hidden patterns in the data and drew conclusions from it, thus we examined the most often occurring words in each product's complaints. This was accomplished by employing a document term matrix with word clouds to plot these terms. When the size of a word in the word cloud corresponds to its frequency for each product. (Fig32)

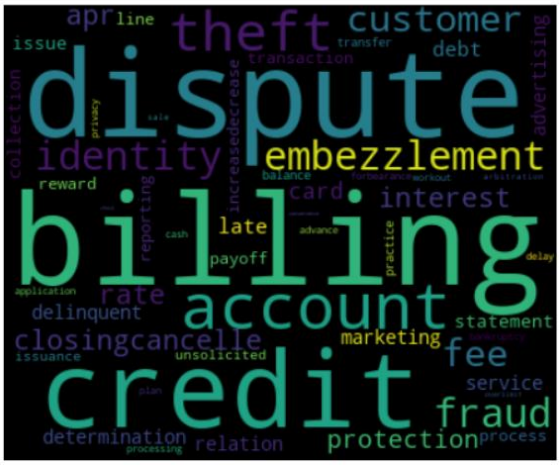
Bank account or service



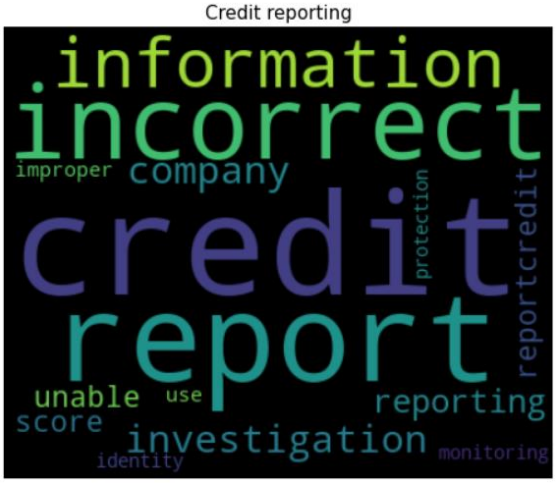
Consumer Loan



Credit card



Credit reporting



Debt collection

Money transfers

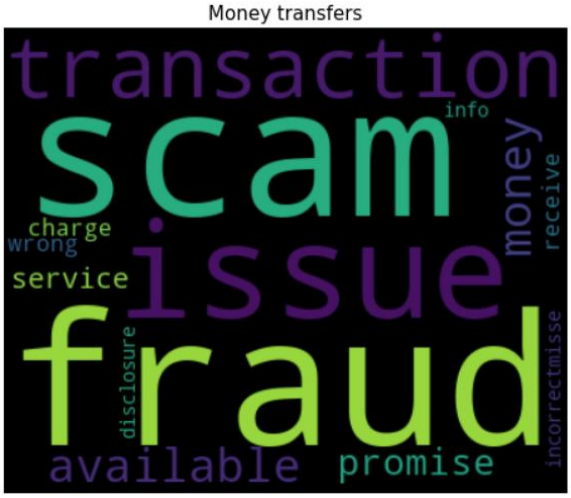


Figure 32: Word cloud product wise

For items such as credit cards, the most common terms include dispute, billing, credit, etc. for Credit reporting terms such as inaccurate, information, credit report, and for money transfer and other financial services terms such as scam, fraud, etc. This technique can extract the core of all difficulties and identify the consumer's concern.

Now that we were more inquisitive, we wanted to know the feelings behind every product issue. For this analysis, we employed the polarity technique to determine the number of negative complaints received, which indicates how well a product is working.

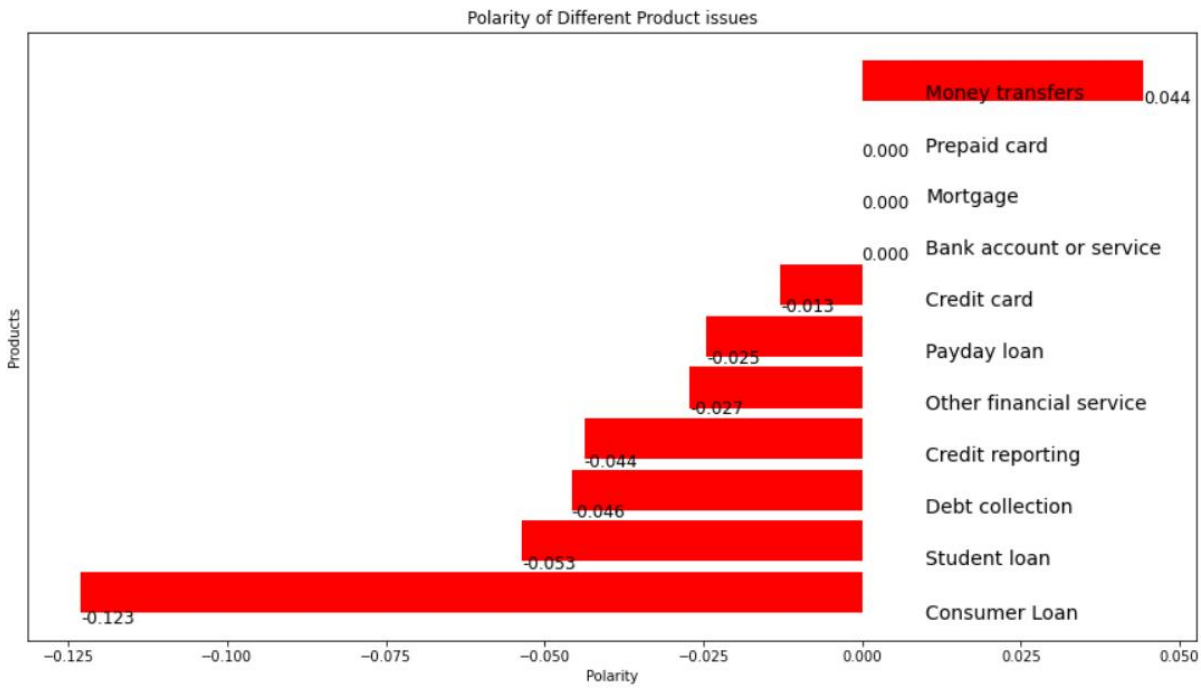


Figure 33: Product polarities vertical bar plot

As expected, the polarity plot reveals that most items have negative polarity, with the consumer loan having the highest negative polarity value of -0.123. In contrast, only Money transfers have a positive polarity value of 0.044, whilst Prepaid cards, mortgages, and bank accounts or services have a neutral polarity rating of 0.0.(Fig 33)

As soon as we completed data cleaning and examination of the original data to gain a knowledge of the dataset, we began training the Generative adversarial networks (GANs) model to generate synthetic data from tabular data. First, we observe the dataset using the head function, which reveals that there are three characteristics developed — cleaned, lemmatized, and polarity — that we no longer require for synthetic data generation (Fig 34). Therefore, we eliminated extraneous columns and encoded the entire dataset. We define a column-name list for categorical variables. This list will be supplied to the model so it can determine how to handle these fields.

issue	state	submitted_via	company_response_to_consumer	cleaned	lemmatized	polarity
Loan modification,collection,foreclosure	CA	Referral	Closed with explanation	loan modificationcollectionforeclosure	loan modificationcollectionforeclosure	0.0
Loan servicing, payments, escrow account	CA	Referral	Closed with explanation	loan servicing payments escrow account	loan servicing payment escrow account	0.0
Incorrect information on credit report	NY	Postal mail	Closed with explanation	incorrect information on credit report	incorrect information credit report	0.0
Repaying your loan	MD	Email	Closed with explanation	repaying your loan	repay loan	0.0
False statements or representation	GA	Web	Closed with explanation	false statements or representation	false statement representation	-0.4

Figure 34: Column cleaned, lemmatized, and polarity

18. CTGAN SYNTHESIZER MODEL FITTING

The amount of provided data is too large for our system to comprehend, so we chose to randomly subset it using a sample function whose frac parameter determined the size of the subset. Therefore, we are taking 550000 rows from the dataset, which corresponds to a 0.1 frac value (Fig 2.18). Next, we simply define an instance of CTGAN Synthesizer and invoke its fit method with the data frame and list of categorical variables. We only train the model for 100 epochs because the discriminator and generator loss become fairly small after this number of iterations. After training, the model's sample function is simply called to generate samples based on the learnt model. In this example, we generate 550000 samples of synthetic data. Consequently, to determine the accuracy of the generated data relative to the original data, we utilise the evaluate function; after executing this function, our model's accuracy is 60+%.

	product	issue	state	submitted_via	company_response_to_consumer
0	Debt collection	Cont'd attempts collect debt not owed	MN	Web	Closed
1	Debt collection	Disclosure verification of debt	VA	Web	Closed
2	Mortgage	Loan modification,collection,foreclosure	FL	Referral	Closed with explanation
3	Bank account or service	Account opening, closing, or management	TX	Web	Closed with non-monetary relief
4	Student loan	Can't repay my loan	TX	Web	Closed with explanation

Figure 35: Head of original data subset

19. EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES

Comparison and Evaluation of Synthetic and Original Sample

Mortgage is the highest-numbered product in both the sample subset, which has more than 17500 records, and the synthetic sample, which likewise contains more than 17500 records. However, the number of mortgages in the synthetic data set is 14,000, which is less than the original subset. Second highest is debt collection, which is greater in the synthetic data set than in the original subset. In the original subset sample, other financial services rank lowest. However, in the synthetic sample, they rank second lowest, behind money transfer. (Fig 36, 37)

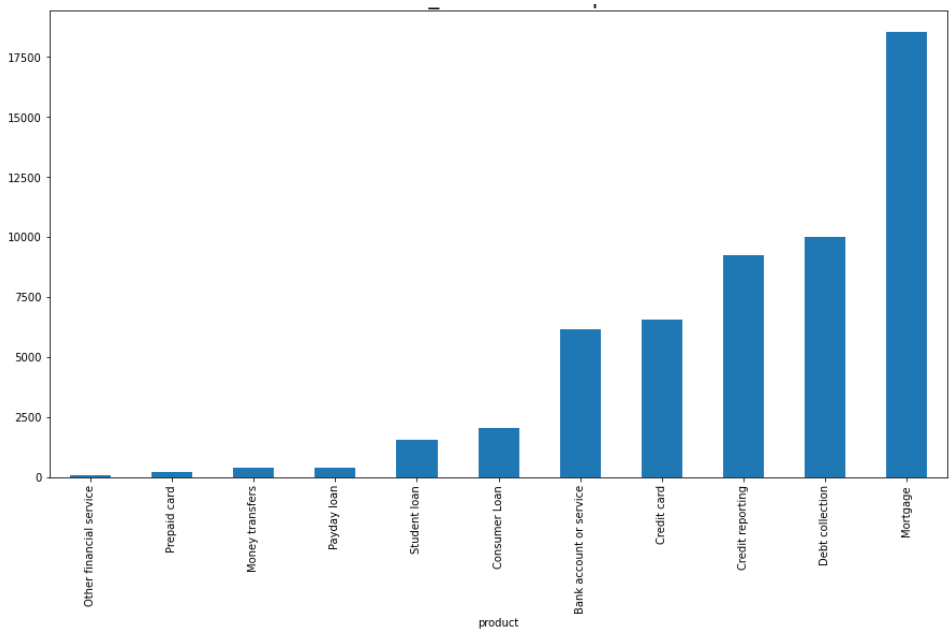


Figure 36: Bar plot of product frequency for original subset

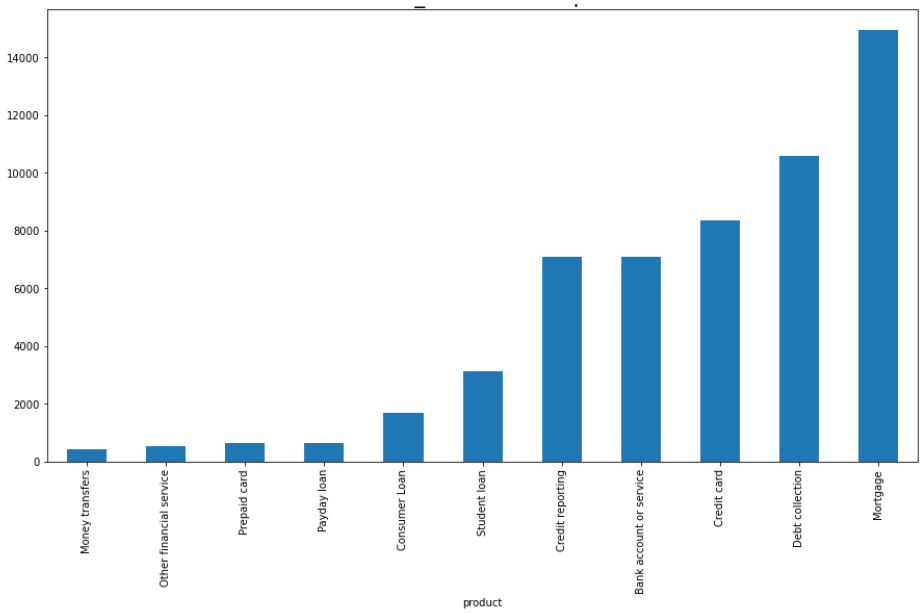


Figure 37: Bar plot of product frequency for synthetic sample

Word cloud is an excellent method for displaying issue column data. The size and color of each word in the word cloud represent its frequency or significance. Most of the words in the word cloud of the issue column for both the original subset sample and the synthetic data sample are Loan, modification, collection, and foreclosure as they are of comparable size and color. However, both samples have different words in the word cloud with different frequencies, but they are exceedingly small in size, indicating that the difference is minimal. (Fig 38,39)

Figure 38: Word cloud issue words frequency for original subset

Figure 39: Word cloud issue words frequency for synthetic sample

In the pie chart, we can see that the web option occurrence is the largest in the submitted via column, while there is a 19% difference between the original subset and the synthetic sample, as the web option occurrence is 65% in the original subset and 46% in the synthetic sample. The remaining selections are presented in the same order, but with varying percentages. (Fig 40,41)

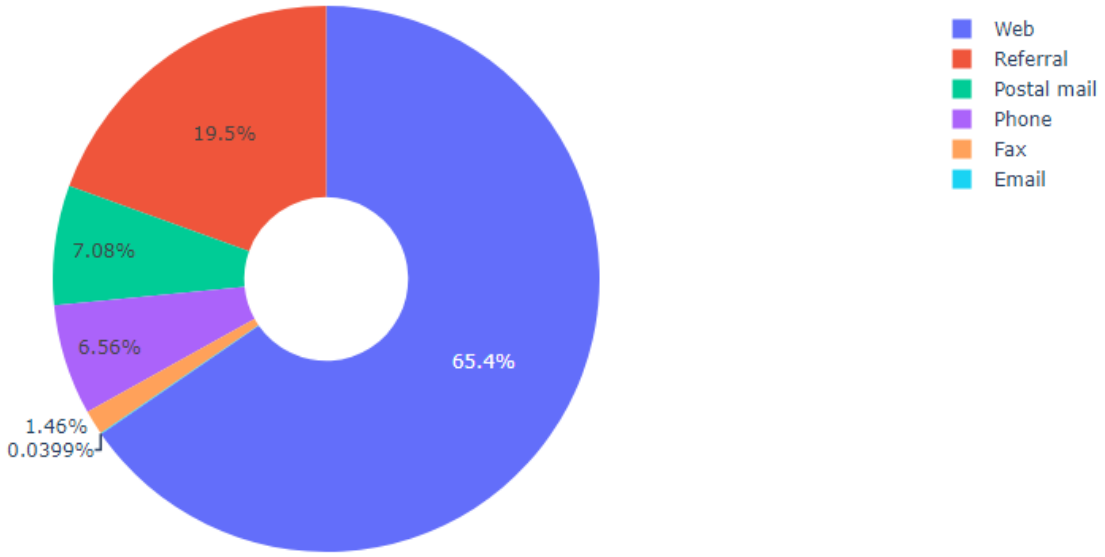


Figure 40: Pie chart submitted_via frequency for original subset

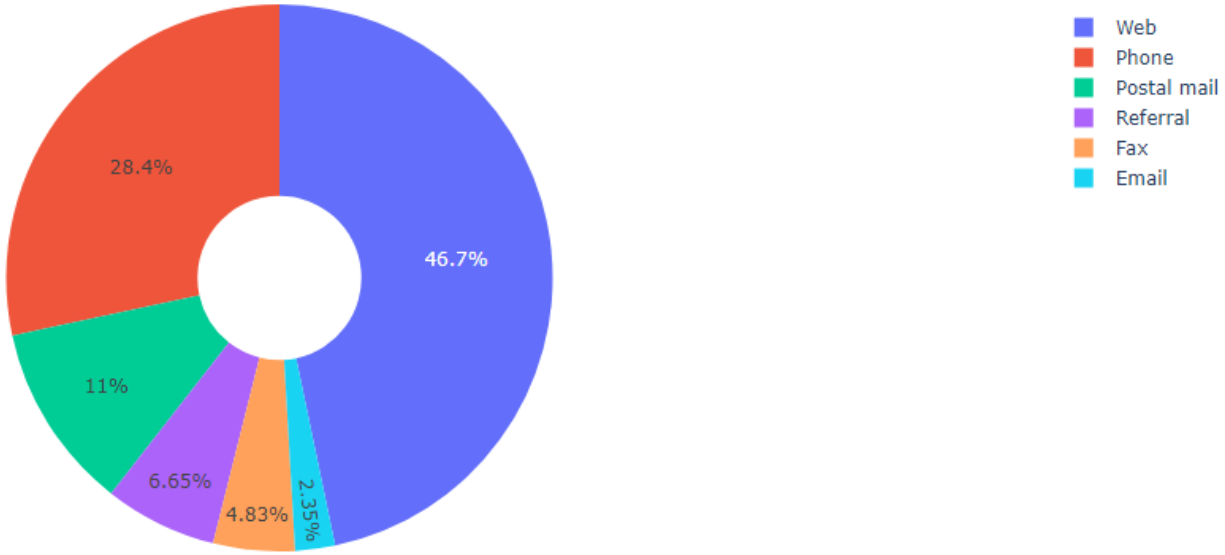


Figure 41: Pie chart submitted_via frequency for synthetic sample

For company response to consumer columns, a horizontal bar plot reveals that closed with explanation is the highest reaction for both samples. Additionally, some responses in both samples are not positioned identically. This suggests that the model trained with a larger sample size would have produced comparable findings. (Fig 42,43)

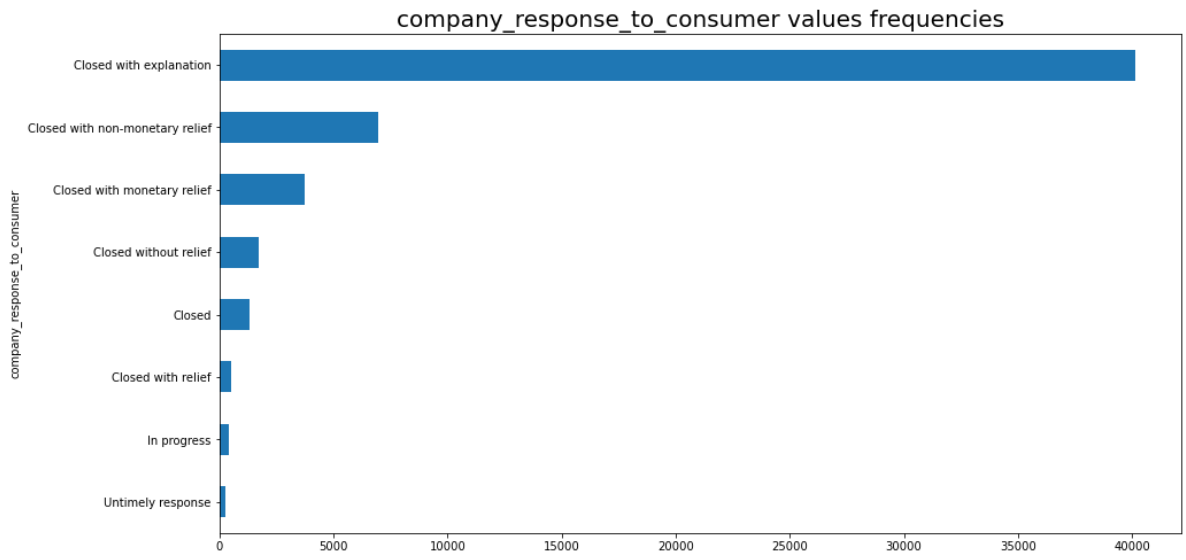


Figure 42: Vertical Bar plot of company response to consumer frequency for original subset

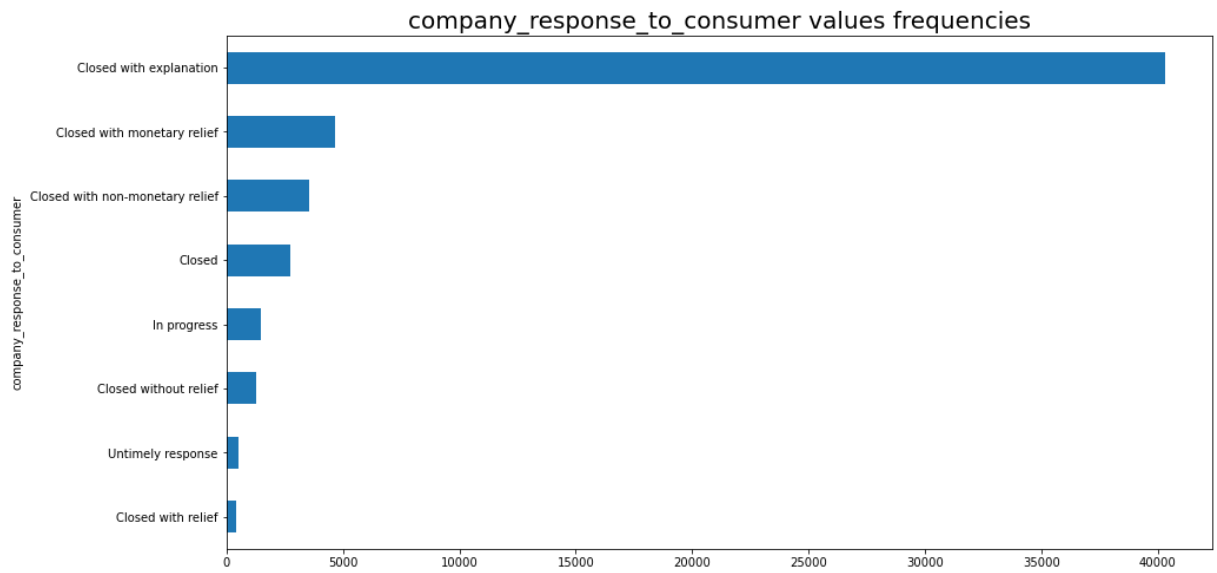


Figure 43: Vertical Bar plot of company response to consumer frequency for synthetic sample

20. 2ND DATASET CONCLUSION AND EVALUATION LIMITATIONS

The visualizations reveal that the resemblance between the original data and the generated data is appropriate. We processed only 0.1 frac or ~10% percent of the randomly selected sample of the original data, which was around 550000, then trained our model with 100 epochs and compared it to the synthetic sample data. Each time we did evaluation, we were able to get an accuracy of 60+ percent. Due to system limitations, we were unable to process more data, but we are achieving good accuracy with relatively few data. This gives us great confidence that if we process more data, we will achieve even greater accuracy. It is important to note that the model cannot manage relational datasets by default, which is why we eliminated complaint id prior to modelling. By default, this modelling technique cannot handle restrictions.

21. REFERENCES

Algoscale, 2021. *Artificial Intelligence: Using GANs for Natural Language Generation.* [Online] Available at: https://algoscale.com/blog/generative-adversarial-networks-for-natural-language-generation/

Ashrapov, I., 2020. *Tabular GANs for uneven distribution,* Moscow, Russia: Moscow Institute of Physics and Technology, Moscow, Russia.

Cem Dilmegani, 2022. *What is Synthetic Data? What are its Use Cases & Benefits?.* [Online] Available at: https://research.aimultiple.com/synthetic-data/

Claire Little, M. E. A. S. S., 2021. Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study. *CONFERENCE OF EUROPEAN STATISTICIANS*, 1-3 12.

Edward Choi, S. B. B. M. J. D. W. F. S. J. S., 2018. *Generating Multi-label Discrete Patient Records.* [Online]
Available at: https://arxiv.org/pdf/1703.06490.pdf

EFF Gennie Gebhart, 2018. *Facebook Doesn't Need To Listen Through Your Microphone.* [Online] Available at: https://www.eff.org/deeplinks/2018/04/facebook-doesnt-need-listen-through-your-microphone-serve-you-creepy-ads

Experian, 2021. *Global Identity and Fraud Report.* [Online] Available at: https://www.experian.com/content/dam/noindex/na/us/im/reports/experian-global-identity-and-fraud-report-2021.pdf?intcmp=2021-gidf-report

FLoC alog, 2022. *Federated Learning of Cohorts.* [Online] Available at: https://github.com/WICG/floc

Florian Eckerli, J. O., 2021. *Generative Adversarial Networks In Finance,* Zurich: Zurich University of Applied Sciences.

Google User Trust Chetna Bindra, 2021. *Building a privacy-first future for web advertising.* [Online] Available at: https://blog.google/products/ads-commerce/2021-01-privacy-sandbox/

Hazel Tang, 2020. *GANs in medicine.* [Online] Available at: https://ai-med.io/neurology/gans-in-medicine/

Ian J. Goodfellow, J. P.-A. a. o., 2014. *Generative Adversarial Nets.* [Online] Available at: https://arxiv.org/pdf/1406.2661.pdf

Ian J. Goodfellow, J. P.-A. M. B. X. D. W.-F. O. A. C. Y. B., 2014. *Generative Adversarial Nets.* [Online] Available at: https://arxiv.org/pdf/1406.2661.pdf

Jeff Heaton , 2021. *Applications of Deep Neural Networks Module 7: Generative Adversarial Networks.* [Online]
Available at: https://github.com/jeffheaton/t81_558_deep_learning/blob/master/t81_558_class_07_5_tabular_synthetic.ipynb
[Accessed 2022].

L. Xu, M. S. A. C.-I. K. V., 2019. *Modeling Tabular Data using Conditional GAN.* [Online] Available at: https://arxiv.org/pdf/1907.00503

Lei Xu, M. S. A. C.-I. K. V., 2019. *Modeling Tabular Data using Conditional GAN,* Cambridge, MA; Cambridge, UK; Móstoles, Spain: MIT LIDS, MRC-BSU, University of Cambridge, Universidad Rey Juan Carlos.

Matt J. Kusner, J. M. H.-L., 2016. *GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution.* [Online] Available at: https://arxiv.org/abs/1611.04051

McKinsey - Behavioral insights, 2018. *Behavioral insights and innovative treatments in collections.* [Online] Available at: https://www.mckinsey.com/business-functions/risk-and-resilience/our-insights/behavioral-insights-and-innovative-treatments-in-collections

McKinsey, 2021. *Buy now, pay later: Five business models to compete.* [Online] Available at: https://www.mckinsey.com/industries/financial-services/our-insights/buy-now-pay-later-five-business-models-to-compete

Mehdi Mirza, S. O., 2014. *Conditional Generative Adversial Nets,* 6: 11.

MIT Data To AI Lab, 2018. *sdv.evaluation.evaluate.* [Online] Available at: https://sdv.dev/SDV/api_reference/api/sdv.evaluation.evaluate.html

Mohammed Alhamid, 2020. *Generative Adversarial Networks GANs: A Beginner's Guide.* [Online] Available at: https://towardsdatascience.com/generative-adversarial-networks-gans-a-beginners-guide-f37c9f3b7817

Nilson Report, 2021. *Nilson Report (David Robertson).* [Online] Available at: https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf

Noseong Park, M. M. K. G. S. J. H. P. Y. K., 2018. *Data Synthesis based on Generative Adversarial Networks.* [Online] Available at: https://arxiv.org/abs/1806.03384

Patrick L, 2015. *Paper review on Generative Adversarial Network (GAN) part 1.* [Online] Available at: https://medium.com/@patrickhk/paper-review-on-generative-adversarial-network-gan-part-1-48597bcc96df

Piacentino, E., 2019. *Master's Degree In Industrial Engineering - Generative Adversarial Network,* Barcelona: Escuela Técnica Superior de Ingeniería Industrial de Barcelona.

Qi Chang, H. Q. Z. Y. Y. G. L. B. D. M., 2022. *Modality Bank: Learn multi-modality images across data centers.* [Online] Available at: https://arxiv.org/pdf/2201.08955.pdf

Ramiro Camino, C. H. R. S., 2018. *Generating Multi-Categorical Samples with Generative Adversarial Networks.* [Online] Available at: https://arxiv.org/abs/1807.01202

Sanket Kamthe, S. A. M. D., 2021. *Copula Flows for Synthetic Data Generation.* [Online] Available at: https://arxiv.org/abs/2101.00598

Sergiu Oprea, G. K. P. M.-G. N. K. S. O.-E. I. O. A. G.-G. A. T. J. G.-R. A. A., 2021. *H-GAN: the power of GANs in your Hands.* [Online]
Available at: https://arxiv.org/abs/2103.15017

Thales Group, 2022. [Online]
Available at: https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/dcv
[Accessed 2022].

The Verge, 2022. *Google parent company Alphabet broke $200 billion in annual revenue for the first time.* [Online]
Available at: https://www.theverge.com/2022/2/1/22912196/google-alphabet-200-billion-annual-revenue-youtube-pixel-search

Tong Che, Y. L. R. Z. R. D. H. W. L. Y. S. Y. B., 2017. *Maximum-Likelihood Augmented Discrete Generative Adversarial Networks.* [Online]
Available at: https://arxiv.org/abs/1702.07983

Udemy Computer Vision, n.d. [Online]
Available at: https://cglearning.udemy.com/course/computer-vision-a-z/learn/lecture/8226364#learning-tools

Unzueta, D., 2021. *How to Generate Tabular Data Using CTGANs.* [Online]
Available at: https://towardsdatascience.com/how-to-generate-tabular-data-using-ctgans-9386e45836a6

WANG, Y., 2020. *A MATHEMATICAL INTRODUCTION TO GENERATIVE ADVERSARIAL NETS (GAN).* [Online]
Available at: arXiv:2009.00169v1

Wang, Y., 2020. *Mathematical Introduction To Generative Adversarial Nets (GAN),* Ithaca, New York: Cornell University.

WORD COUNTS

TEAM MEMBER NAME	TOPICS	APPROX. WORDS
George Karseras	BUSINESS CASE, THE MATHEMATICS OF GENERATIVE ADVERSARIAL NETWORKS, LOAN DATASET EXPLORATORY DATA ANALYSIS EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES EVALUATION LIMITATIONS, EXECUTIVE SUMMARY, CODE ANALYSIS	~2575
Mateusz Milczarek	MOTIVATION, WHAT ARE GANS, LOAN DATASET CTGAN FITTING, EXECUTIVE SUMMARY, CODE ANALYSIS	~2530
Dev Upadhyay	COMPLAINT DATASET DATA PRE-PROCESSING EXPLORATORY DATA ANALYSIS CTGAN SYNTHESIZER MODEL FITTING EVALUATION OF SYNTHETIC AND ORIGINAL SAMPLES EVALUATION LIMITATIONS, EXECUTIVE SUMMARY, CODE ANALYSIS	~2440
Hussain Aftab	LITERATURE REVIEW, EXECUTIVE SUMMARY, CODE ANALYSIS	~2240
Anurag Sinha	STRUCTURE, BUSINESS CASE, THE MATHEMATICS OF GENERATIVE ADVERSARIAL NETWORKS, GAN SYNTHETIC DATA AND ITS APPLICATIONS, EXECUTIVE SUMMARY, CODE ANALYSIS	~2475
		TOTAL APPROX. WORDS: **12,260** excluding executive summary, references, code, appendix

22. APPENDIX

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887379 entries, 0 to 887378
Data columns (total 74 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   id                            887379 non-null  int64
 1   member_id                     887379 non-null  int64
 2   loan_amnt                     887379 non-null  float64
 3   funded_amnt                   887379 non-null  float64
 4   funded_amnt_inv               887379 non-null  float64
 5   term                          887379 non-null  object
 6   int_rate                      887379 non-null  float64
 7   installment                   887379 non-null  float64
 8   grade                         887379 non-null  object
 9   sub_grade                     887379 non-null  object
 10  emp_title                     835917 non-null  object
 11  emp_length                    842554 non-null  object
 12  home_ownership                887379 non-null  object
 13  annual_inc                    887375 non-null  float64
 14  verification_status           887379 non-null  object
 15  issue_d                       887379 non-null  object
 16  loan_status                   887379 non-null  object
 17  pymnt_plan                    887379 non-null  object
 18  url                           887379 non-null  object
 19  desc                          126028 non-null  object
 20  purpose                       887379 non-null  object
 21  title                         887227 non-null  object
 22  zip_code                      887379 non-null  object
 23  addr_state                    887379 non-null  object
 24  dti                           887379 non-null  float64
 25  delinq_2yrs                   887350 non-null  float64
 26  earliest_cr_line              887350 non-null  object
 27  inq_last_6mths                887350 non-null  float64
 28  mths_since_last_delinq        433067 non-null  float64
 29  mths_since_last_record        137053 non-null  float64
 30  open_acc                      887350 non-null  float64
 31  pub_rec                       887350 non-null  float64
 32  revol_bal                     887379 non-null  float64
 33  revol_util                    886877 non-null  float64
 34  total_acc                     887350 non-null  float64
 35  initial_list_status           887379 non-null  object
 36  out_prncp                     887379 non-null  float64
 37  out_prncp_inv                 887379 non-null  float64
 38  total_pymnt                   887379 non-null  float64
 39  total_pymnt_inv               887379 non-null  float64
 40  total_rec_prncp               887379 non-null  float64
 41  total_rec_int                 887379 non-null  float64
 42  total_rec_late_fee            887379 non-null  float64
 43  recoveries                    887379 non-null  float64
 44  collection_recovery_fee       887379 non-null  float64
 45  last_pymnt_d                  869720 non-null  object
 46  last_pymnt_amnt               887379 non-null  float64
 47  next_pymnt_d                  634408 non-null  object
 48  last_credit_pull_d            887326 non-null  object
 49  collections_12_mths_ex_med    887234 non-null  float64
 50  mths_since_last_major_derog   221703 non-null  float64
 51  policy_code                   887379 non-null  float64
 52  application_type               887379 non-null  object
 53  annual_inc_joint              511 non-null     float64
 54  dti_joint                     509 non-null     float64
 55  verification_status_joint     511 non-null     object
 56  acc_now_delinq                887350 non-null  float64
 57  tot_coll_amt                  817103 non-null  float64
 58  tot_cur_bal                   817103 non-null  float64
 59  open_acc_6m                   21372 non-null   float64
 60  open_il_6m                    21372 non-null   float64
 61  open_il_12m                   21372 non-null   float64
 62  open_il_24m                   21372 non-null   float64
 63  mths_since_rcnt_il            20810 non-null   float64
 64  total_bal_il                  21372 non-null   float64
 65  il_util                       18617 non-null   float64
 66  open_rv_12m                   21372 non-null   float64
 67  open_rv_24m                   21372 non-null   float64
 68  max_bal_bc                    21372 non-null   float64
 69  all_util                      21372 non-null   float64
 70  total_rev_hi_lim              817103 non-null  float64
 71  inq_fi                        21372 non-null   float64
 72  total_cu_tl                   21372 non-null   float64
 73  inq_last_12m                  21372 non-null   float64
dtypes: float64(49), int64(2), object(23)
memory usage: 501.0+ MB
```

Appendix Figure 1: Loan dataset description

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 773968 entries, 42535 to 887378
Data columns (total 43 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   loan_amnt                   773968 non-null  float64
 1   funded_amnt                 773968 non-null  float64
 2   funded_amnt_inv             773968 non-null  float64
 3   term                        773968 non-null  object
 4   int_rate                    773968 non-null  float64
 5   installment                 773968 non-null  float64
 6   sub_grade                   773968 non-null  object
 7   emp_length                  773968 non-null  object
 8   home_ownership              773968 non-null  object
 9   annual_inc                  773968 non-null  float64
 10  verification_status         773968 non-null  object
 11  loan_status                 773968 non-null  object
 12  pymnt_plan                  773968 non-null  object
 13  purpose                     773968 non-null  object
 14  addr_state                  773968 non-null  object
 15  dti                         773968 non-null  float64
 16  delinq_2yrs                 773968 non-null  float64
 17  inq_last_6mths              773968 non-null  float64
 18  open_acc                    773968 non-null  float64
 19  pub_rec                     773968 non-null  float64
 20  revol_bal                   773968 non-null  float64
 21  revol_util                  773968 non-null  float64
 22  total_acc                   773968 non-null  float64
 23  initial_list_status         773968 non-null  object
 24  out_prncp                   773968 non-null  float64
 25  out_prncp_inv               773968 non-null  float64
 26  total_pymnt                 773968 non-null  float64
 27  total_pymnt_inv             773968 non-null  float64
 28  total_rec_prncp             773968 non-null  float64
 29  total_rec_int               773968 non-null  float64
 30  total_rec_late_fee          773968 non-null  float64
 31  recoveries                  773968 non-null  float64
 32  collection_recovery_fee     773968 non-null  float64
 33  last_pymnt_amnt             773968 non-null  float64

 34  collections_12_mths_ex_med  773968 non-null  float64
 35  application_type            773968 non-null  object
 36  acc_now_delinq              773968 non-null  float64
 37  tot_coll_amt                773968 non-null  float64
 38  tot_cur_bal                 773968 non-null  float64
 39  total_rev_hi_lim            773968 non-null  float64
 40  Last_Major_Derog_Length     773968 non-null  category
 41  Last_Delinq_Length          773968 non-null  category
 42  Last_Record_Length          773968 non-null  category
dtypes: category(3), float64(29), object(11)
memory usage: 244.3+ MB
```

Appendix Figure 2: Loan dataset processed data type

```
Classifier F1-scores and their Jaccard similarities::
                             f1_real  f1_fake  jaccard_similarity
index
DecisionTreeClassifier_fake   0.6188   0.8075              0.4467
DecisionTreeClassifier_real   0.9150   0.8600              0.7467
LogisticRegression_fake       0.6787   0.8638              0.5399
LogisticRegression_real       0.9613   0.9025              0.8779
MLPClassifier_fake            0.5787   0.8225              0.4260
MLPClassifier_real            0.8125   0.8062              0.5968
RandomForestClassifier_fake   0.8050   0.8775              0.7699
RandomForestClassifier_real   0.9513   0.9137              0.8980

Privacy results:
                                        result
Duplicate rows between sets (real/fake)  (0, 0)
nearest neighbor mean                    4.2863
nearest neighbor std                     1.3785

Miscellaneous results:
                                 Result
Column Correlation Distance RMSE  0.0842
Column Correlation distance MAE   0.0531

Results:
                                                 result
Basic statistics                                 0.9774
Correlation column correlations                  0.8920
Mean Correlation between fake and real columns      NaN
1 - MAPE Estimator results                       0.8429
Similarity Score                                    NaN
```

Appendix Figure 3: F1-score and RMSE analysis of generative loan dataset

23. INPUT DATA

The first is a financial loans dataset. This dataset is particularly category and numerically heavy and thus is the easier of the two datasets to work with.
https://www.kaggle.com/datasets/husainsb/lendingclub-issued-loans?select=lc_loan.csv

The second dataset is a list of complaints. This dataset is particularly string focused.
https://www.kaggle.com/datasets/kaggle/us-consumer-finance-complaints