1st June 2022

5th web meeting of the Task Force on Mobile Network Operator data for official statistics (TF MNO)

Generation of synthetic data from real MNO data through Neural Network: motivations, techniques and preliminary results

Authors: Fabrizio De Fausti – Massimo De Cubellis – Francesco Pugliese – Roberta Radini

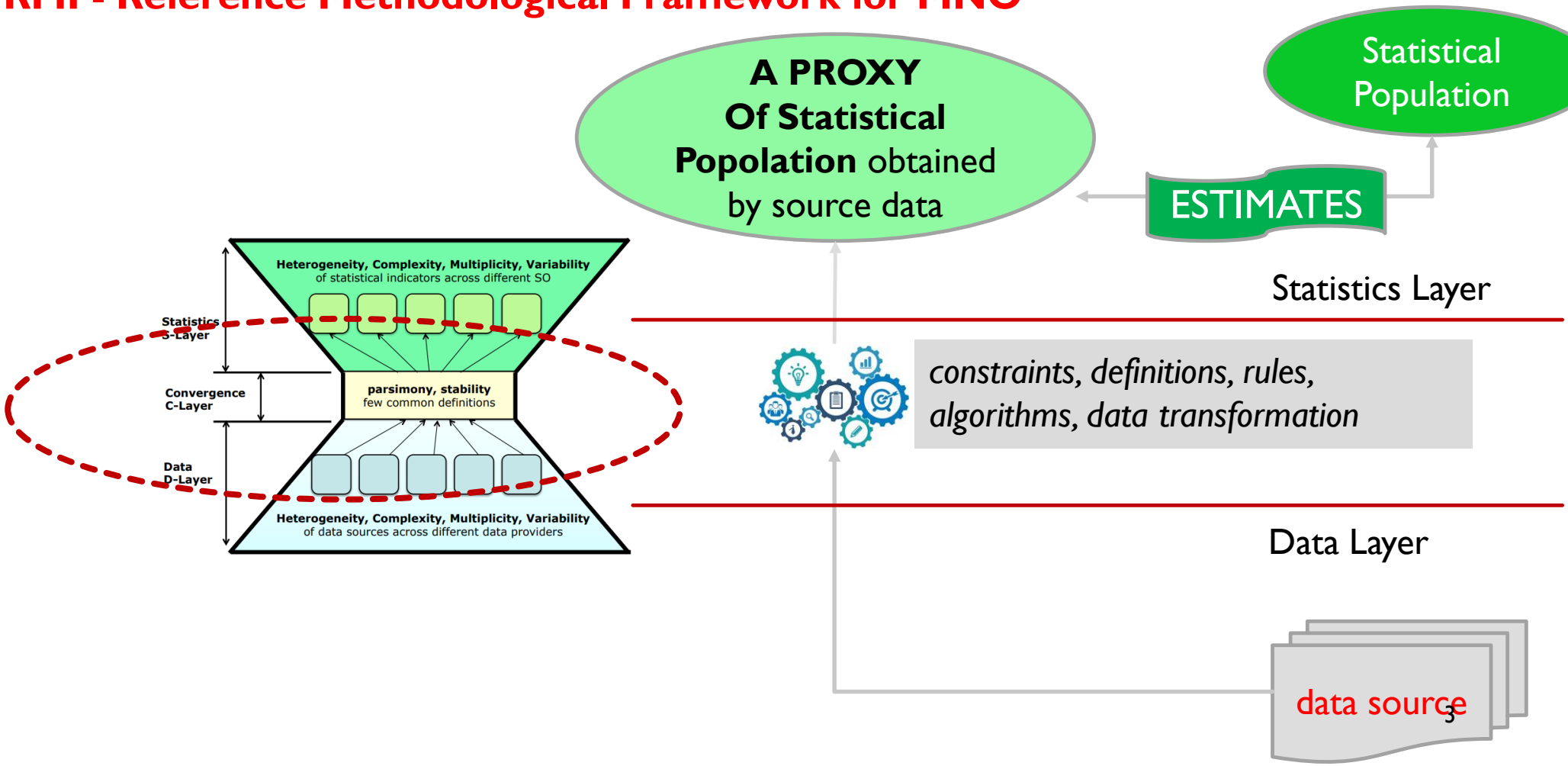
OUTLINE

- Why use synthetic data
- How to generate synthetic data

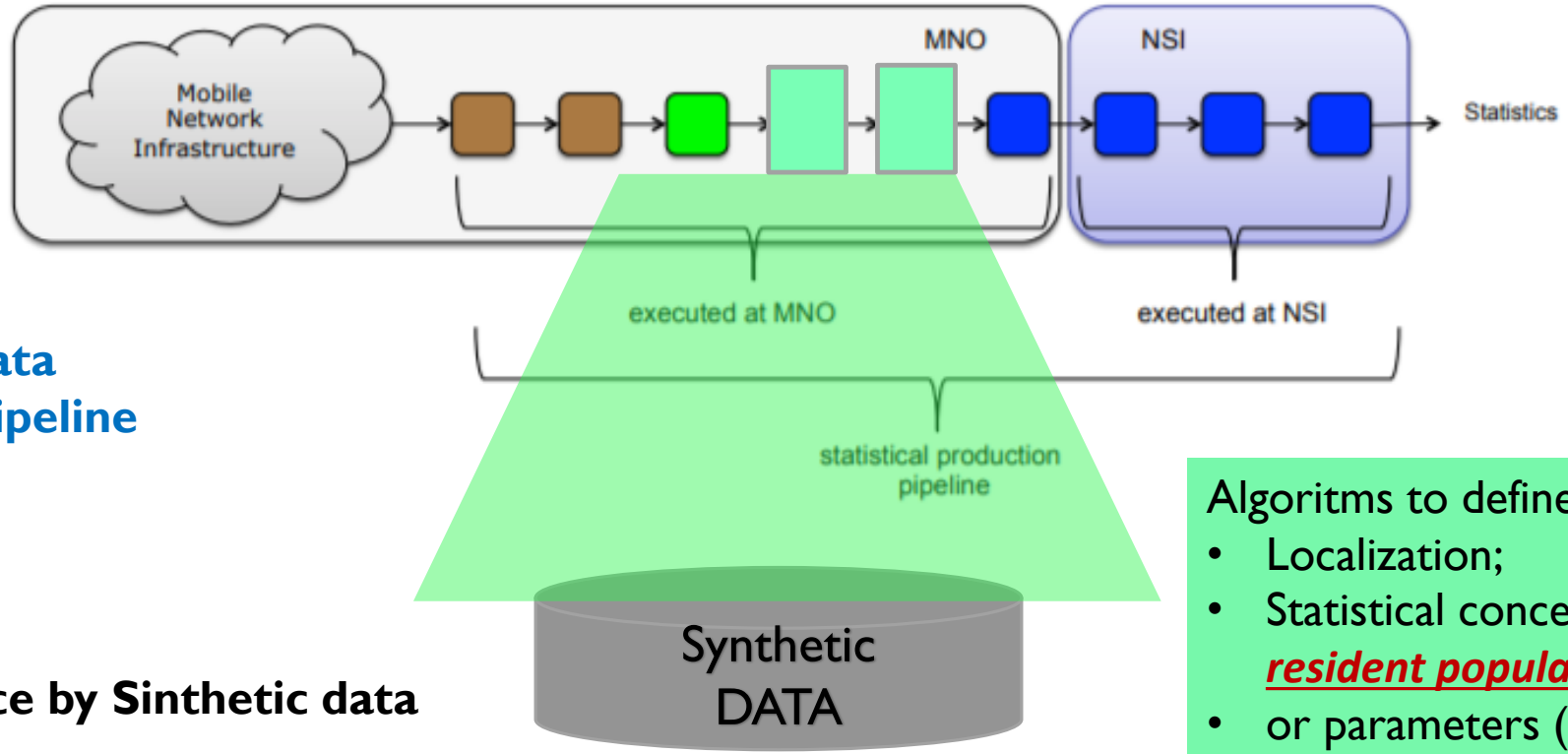
Istat

Why use synthetic data

RMF- Reference Methodological Framework for MNO



Why use synthetic data



Can synthetic data transform the pipeline in a new way?

Algoritms to define:

- Localization;
- Statistical concept (e.i: ***usually resident population)***
- or parameters (e.i.: what is the timing of night?)

Open-source by Sinthetic data

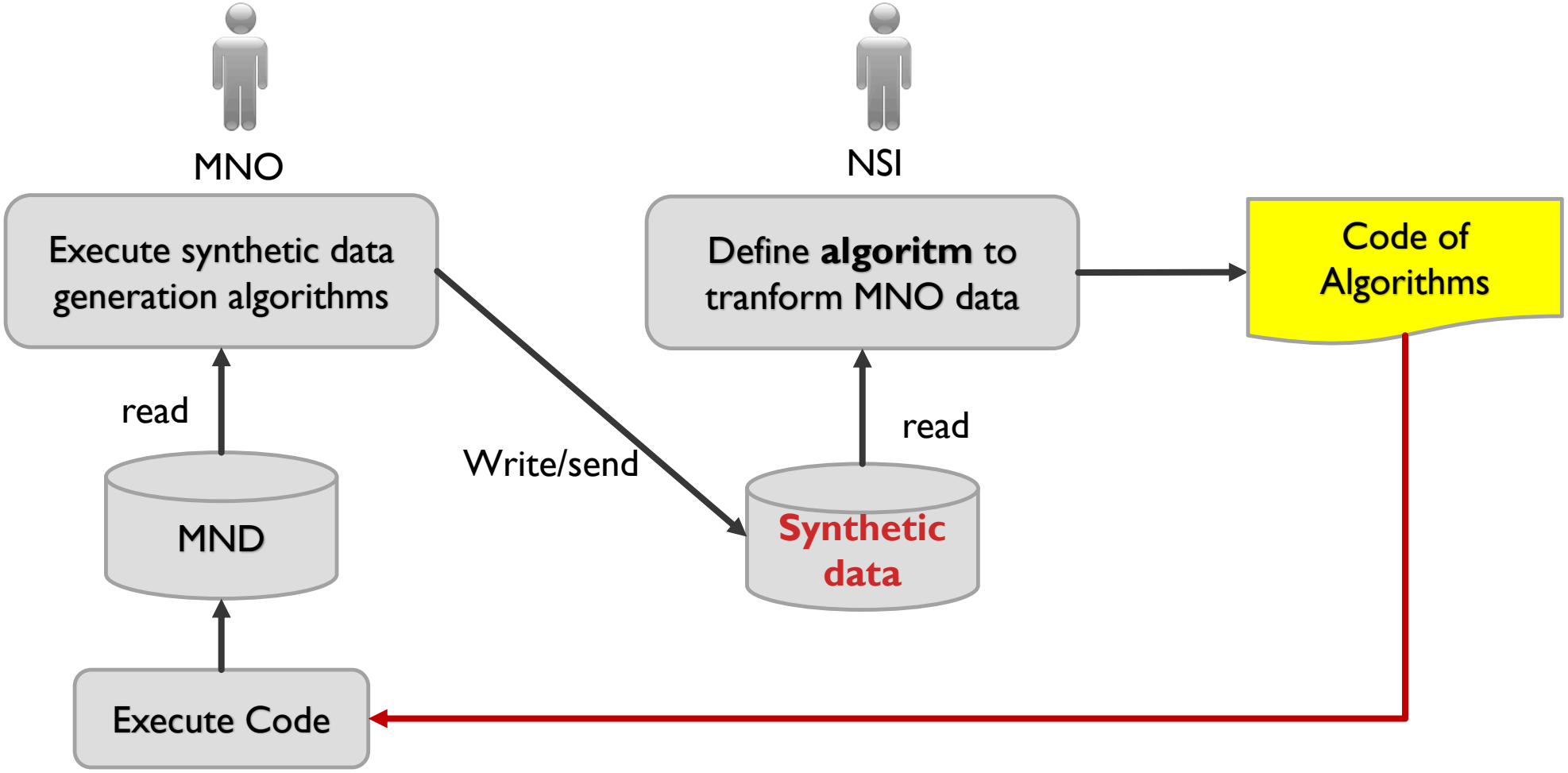
reference pseudo-code description

open-source

human-readable guidelines

Istat

How to use synthetic data



How to generate synthetic data

Statistical models, such as Bayesian networks, or Hidden Markov models, provide an explicit, parametric model of real data distribution. The features these models extract from their training data is determined upfront.

Non-parametric models, such as generative adversarial networks (GANs) or variational auto encoders (VAEs), do not estimate a parametric likelihood function to generate new samples from a representation of the joint multivariate distribution of real data.

We focused on:

- the development of **CTGAN** via a Framework called SDGym (Synthetic Data Gym)
- and the evaluation of **Utility metrics** and **Privacy metrics**

Always keeping in mind **use and nature of these senthetic data** which:

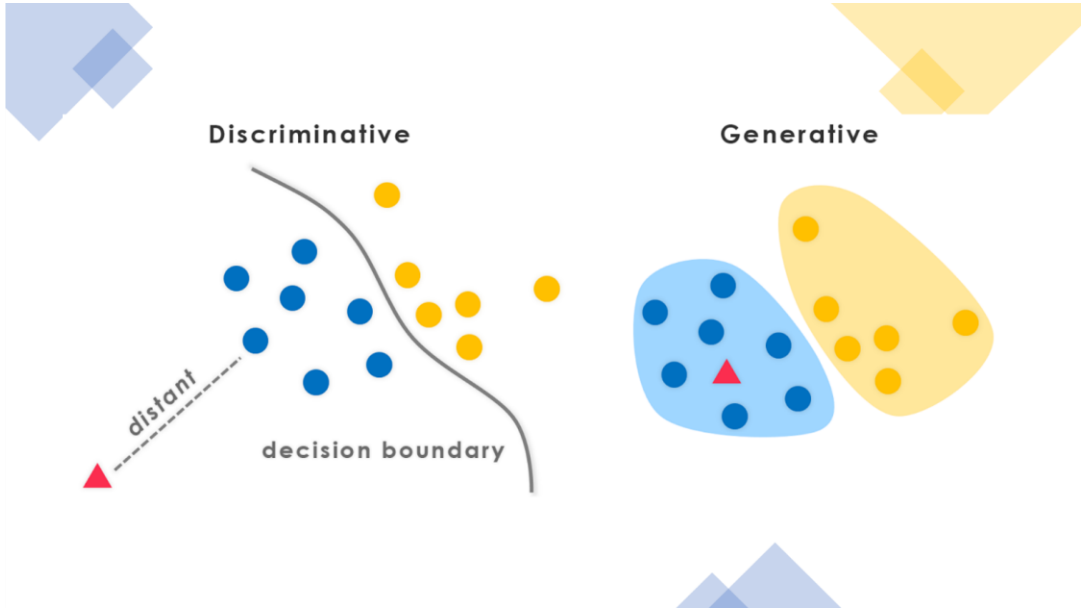
- must **represent the characteristics** of real data (SIM code, antenna/sector code and Time)
- **do not need** to be necessarily extended in space and time as they will be used to **develop algorithms** in house of NSI

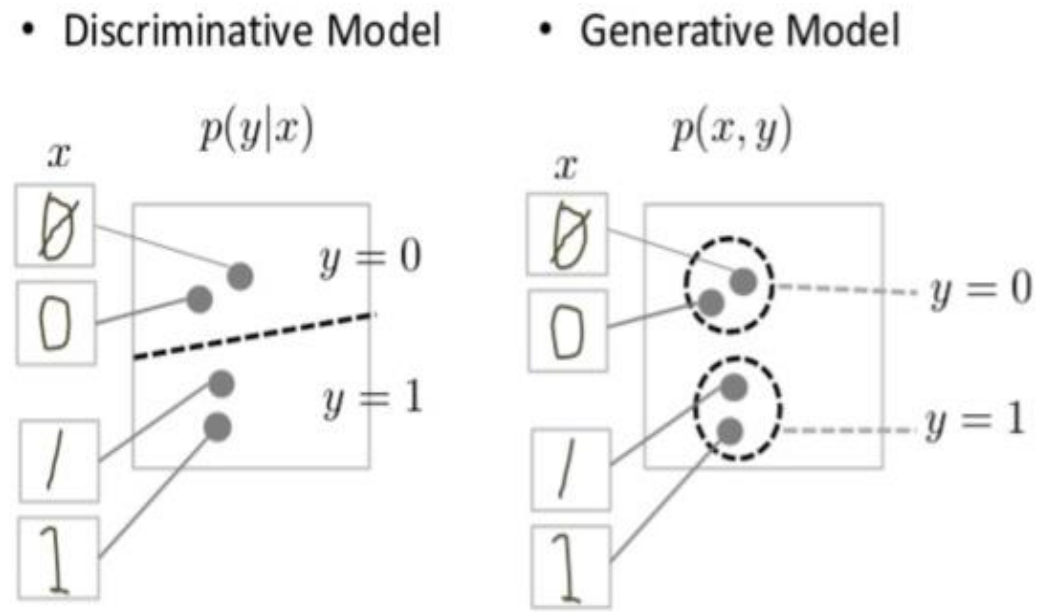
Istat

Generative Modeling

Generative Modeling is an **Unsupervised Learning Task** leading to the automatic discovering of regularities or «patterns» within the Original Data in order to generate new Synthetic Data.

Therefore, a **Generative Model** can be used to generate new data samples, which could teorethically belong to the Original Dataset, even though they are new synthetic data.

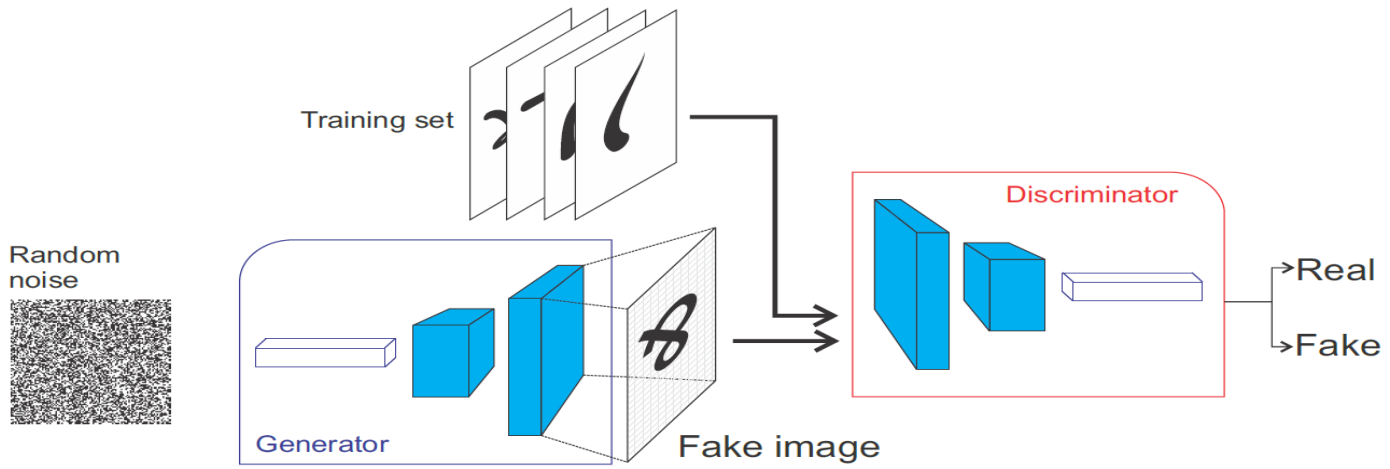




Generative Adversarial Networks (GAN)

- A **Generative Model** is a statistical model of the joint distribution *P(X,Y)* over a given observable variable X and a variable Y

- **Generative Adversarial Networks (GANs)** were introduced by the Guru of **Deep Learning**, **Ian Goodfellow** in **2014** and are a way to learn **Latent Spaces** of Images. A **Latent space** is a simple hidden representation of a data point. The generator model in the GAN architecture takes a point from the latent space as input and generates a new image. Through training, the generator learns to map points into the latent space with specific output images and this mapping will be different each time the model is trained.



Generative Adversarial Networks (GAN)

- Basically, a **GAN** is made of a "Fake" Network (**Generator**) and an "Expert" Network (**Discriminator**) which are adversarially trained to overcome the other one reciprocally.
- **Generator**: takes a random vector in input (a random point in the latent space) and decode it into a synthetic image. Once it is trained, the **Generator** generates new samples shaping new data over the input data distribution in a totally unsupervised way.
- **Discriminator** (or adversarial) : takes a real image or synthetic as input and classifies it as real or fake. In other words the Discriminator is a classifier trying to understand which samples come from the training set and which one are generated synthetic data.

GAN Algorithm:

for every epoch:

1) Extract random points from the Latent Space.

2) Generate images for the Generator adopting random noise (Gaussian Noise better).

3) Mix true images with generated ones.

4) Train the discriminator by these mixed images, with the labels: true for real images, and false for generated images.

5) Extract new points from the latent space.

6) Train the GAN adopting these random vectors, with all labels saying "true". This update Generators weights (only generator, discriminator is freezed) to check if the discrimator is able to recognize them or it says they are all "true" also the generated ones. This trains the generator to mislead the discriminator.

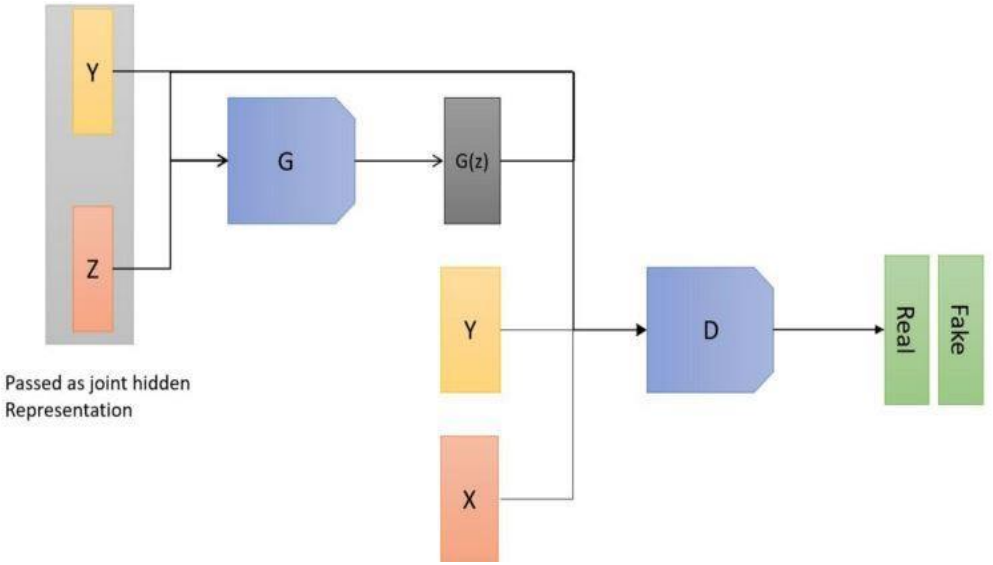
Istat

Conditional Tabular GAN (CTGAN)

- **SDGym's CTGANs** are a combination of Conditional and Tabular GANs.

- **Conditional GAN:** Although **GANs** are able to generate some good samples of data points, they are not able to generate data points with the target labels (i.e. categorical targets) and if generated these data lack diversity. **Conditional GAN** was proposed by *M. Mirza* in late 2014. He modified the architecture by adding the label ***y*** as a parameter to the input of the generator and try to generate the corresponding data point. It also adds labels to the discriminator input to distinguish real data better. In this architecture, the random input noise ***Z*** is combined with the label ***Y*** in the joint hidden representation.

- **Tabular GAN:** These are a class of **GANs** able to generate tabular data such as: with various data types (int, decimals, categories, time, text) and with different shapes of distribution.



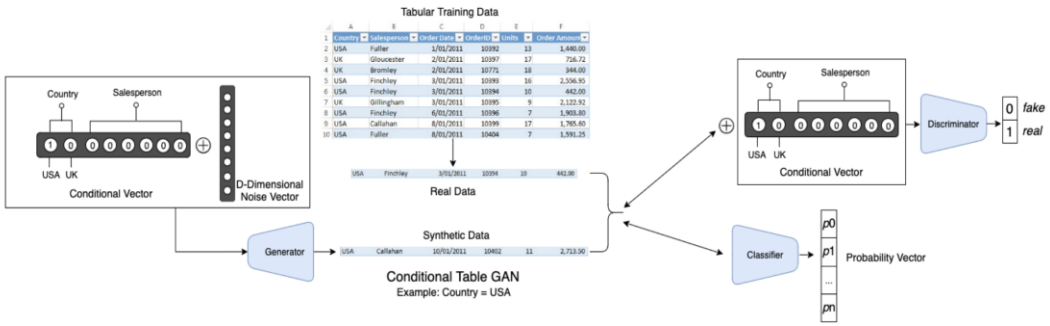


Figure 4.1: Synthetic Tabular Data Generation via CTAB-GAN

The experimentation (1/2)

Dataset

- A random sample of WIND's CDR
- Dimension: 10,000 rows – 4 attributes
- Attributes: SIM Code
  Call Date
  Time call
  Cell_Call_Code (Sector/Antenna code)

Pre-proceesing (steps)

- Pseudo-anonymization
- Setting of data-type: ***categorical*** (SIM code, Antenna code) or ***continuous***(date and time of call)

Istat

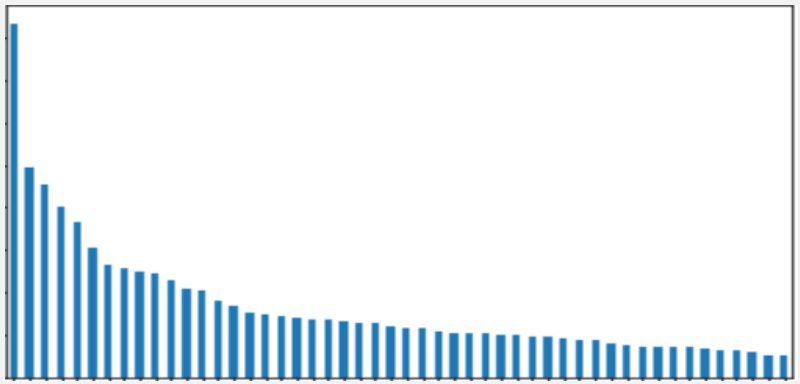
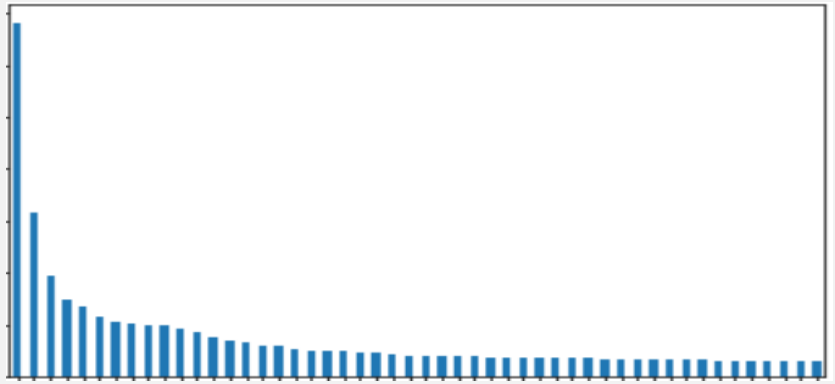
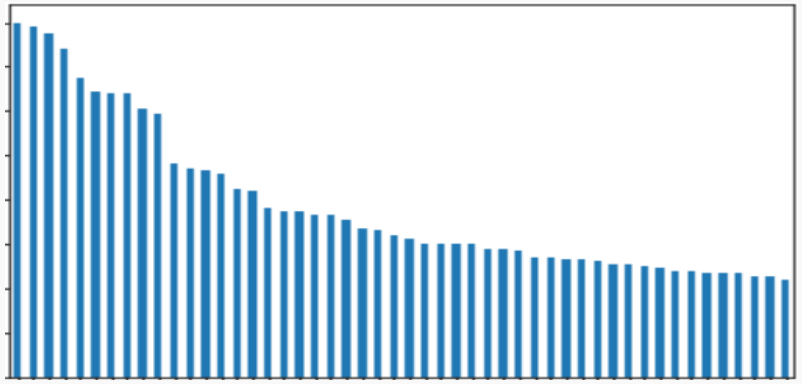
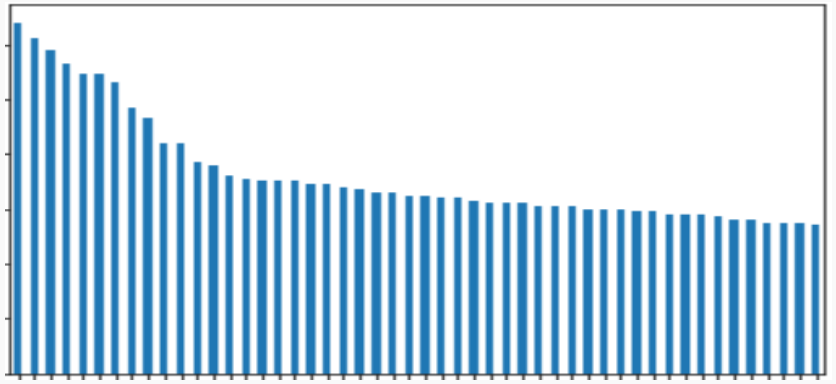
The experimentation (2/2)

Process

- Input : a random sample of WIND's CDR (***original dataset***)
- Type of process: generate synthetic data
- Framework used: SDGym - Synthetic Data Gym Metrics Evaluation (https://github.com/sdv-dev/SDGym)
- Algorithm used to generate synthetic data: Synthetic Data Vault (SDV) – based on CTGAN
- Output: synthetic dataset of WIND's CDR (***synthetic dataset)***

Original and Synthetic Data Univariate Distribution Visualization

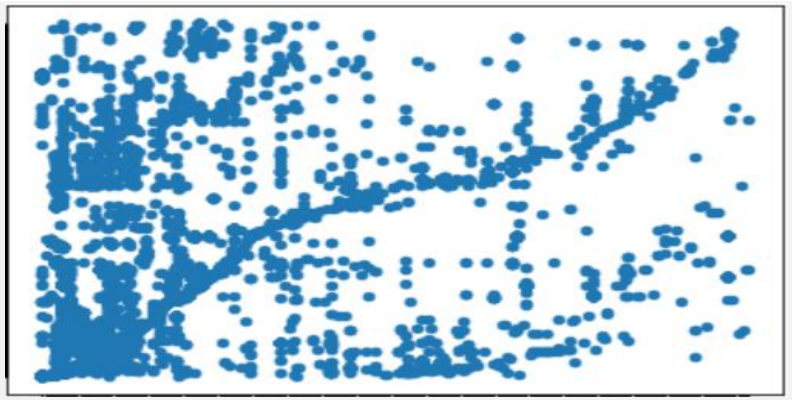
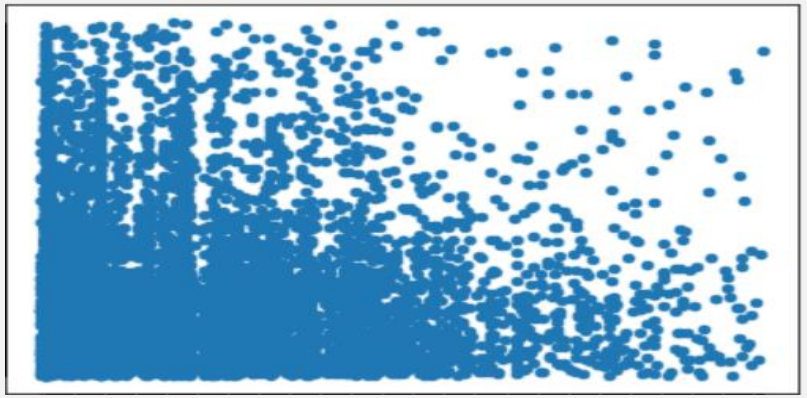
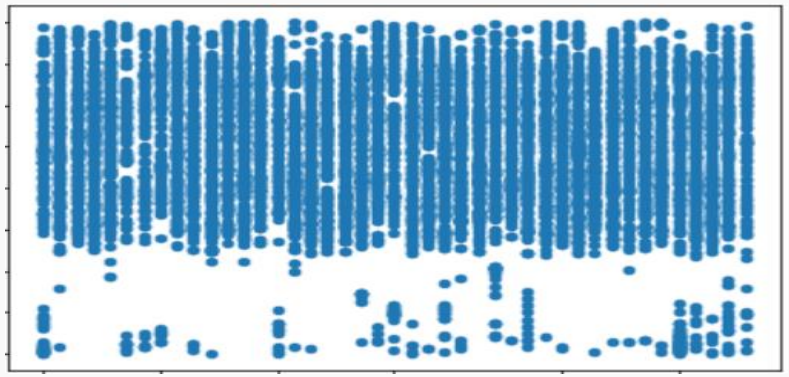
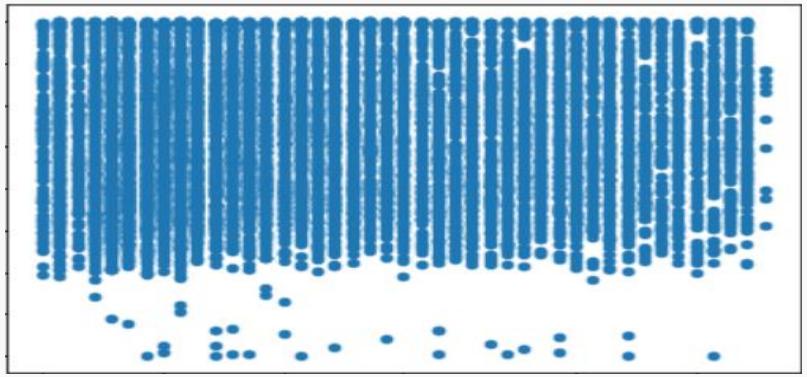
Graphical analysis with **Univariate Distributions Comparisons** on the **Categorical** Variables from **Real Data** and **Synthetic Data**.

	ORIGINAL DATASET	SYNTHETIC DATASET
SIM_CODE		
ANTENNA_CODE		

Istat

Original and Synthetic Data Bivariate Distribution Visualization

Graphical analysis via Scatterplots of **Univariate Distributions Comparisons** on the **Categorical** Variables from **Real Data** and **Synthetic Data**.

	ORIGINAL DATASET	SYNTHETIC DATASET
SIM_CODE – ANTENNA_CODE		
CALL_DATA – TIME_CALL		

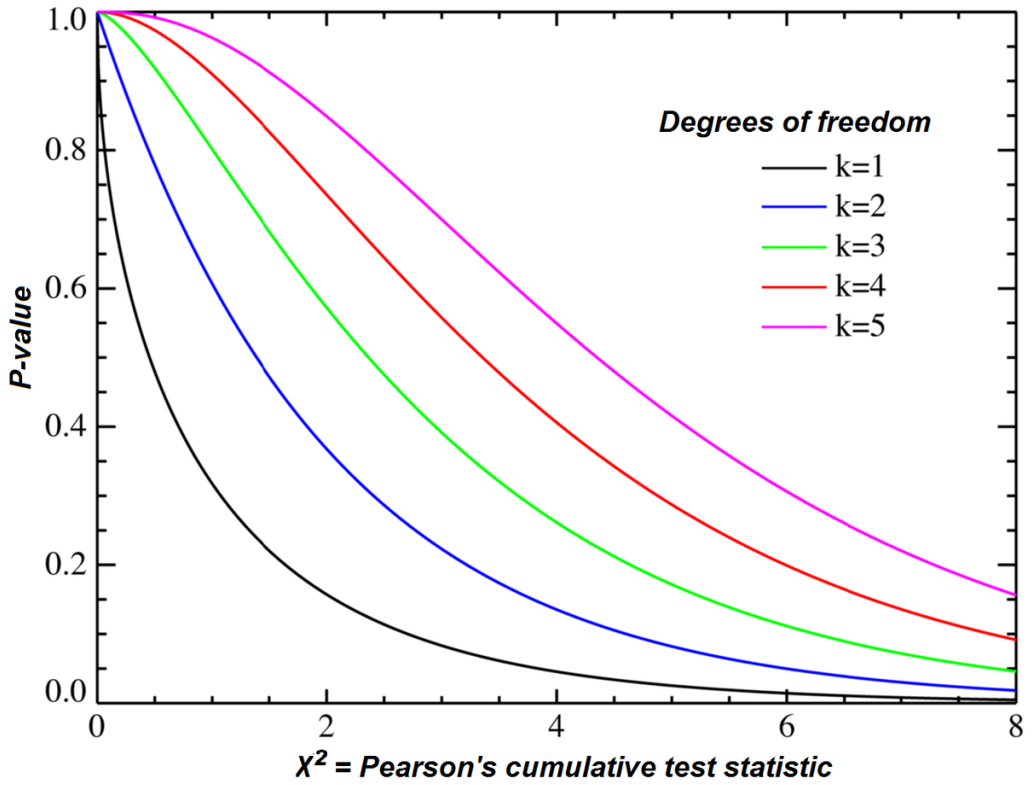
Istat

Utility Metrics - Model Evaluation via SDGym Tools: Statistical Metrics

- The **metrics** of the **Synthetic Data Gym** of the **Statistical Metrics Family** compare the tables by running different types of statistical tests on them. In the simplest scenario, these metrics compare individual columns from the real table with the corresponding column from the synthetic table.

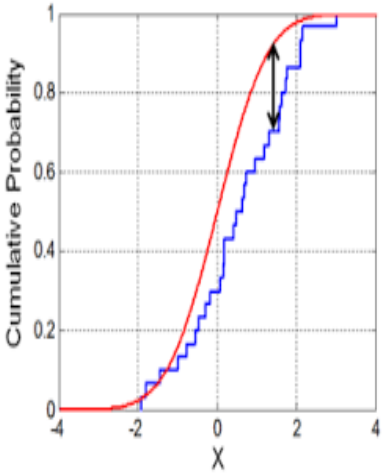
- **sdv.metrics.tabular.CSTest:** This metric make use of the **Chi-Squared Test** to compare the distributions of two discrete columns. The output for each column is the **CSTest p-value**, which indicates the probability of the two columns having been sampled from the same distribution.

- **Chi-Squared Test p-value** must be between 0 and 1. Since we achieved **1.0** in this test, it means that our distributions (original and synthetic) are sampled from the same distribution of data.



Utility Metrics - Model Evaluation via SDGym Tools: Statistical Metrics

- **sdv.metrics.tabular.KSTest:** This metric uses the two-sample **Kolmogorov–Smirnov Test** to compare the distributions of continuous columns using the empirical **CDF (Cumulative Distibution Function: P(X≤x)**). The output of **KSTest** for each column is 1 minus the **KS Test D** statistic, which indicates the maximum distance between the expected CDF and the observed CDF values.



- The letter **"D"** stands for "distance." Geometrically, D measures the maximum vertical distance between the empirical cumulative distribution function (ECDF) of the sample and the cumulative distribution function (CDF) of the reference distribution.

- If the two samples were randomly sampled from identical populations, what is the probability that the two cumulative frequency distributions would be as far apart as observed? More precisely, what is the chance that the value of the **Komogorov-Smirnov D statistic** would be as large or larger than observed? If the P value is small, conclude that the two groups were sampled from populations with different distributions. The populations may differ in median, variability or the shape of the distribution.

- So our result **0.95375** which is 1-D, means that the distance between CDF of original data and CDF of Synthetic Data is low, hence the two distribution are very close.

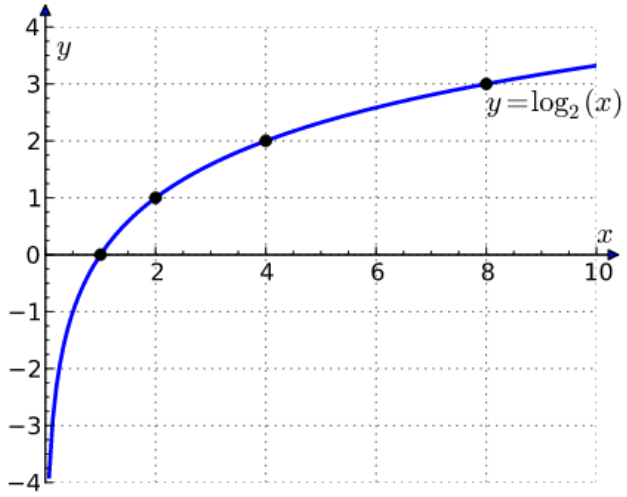
Istat

Utility Metrics - Model Evaluation via SDGym Tools: Likelihood Metrics

- The **metrics** of this family compare the tables by fitting the real data to a probabilistic model and afterwards compute the likelihood of the synthetic data belonging to the learned distribution.

- **sdv.metrics.tabular.BNLikelihood:** This metric fits a **Bayesian Network** to the real data and then evaluates the average likelihood of the rows from the synthetic data on it.

- **Bayesian Networks Likelihood** is the Error calculated on Synthetic data after fitting the model on Real Data. Very low error (likelihood) like **0.00013183** means the two datasets are very close in terms of probabilistic models.



- **sdv.metrics.tabular.BNLikelihood:** T his metric fits a **Bayesian Network** to the real data and then evaluates the average likelihood of the rows from the synthetic data on it. With very low error (close to 0) according to the Log function, the output value must be negative, therefore a score like this one: **-17.415473543655498**, it is a very good result.

Istat

Privacy Metrics - Model Evaluation by Matching Common Values

- **Privacy Metric:** In this kind of metric we merge the Original Dataset and the Synthetic Dataset doing and Inner Join over 1 or 2 columns. And we get the list of rows merged. After this step, we search for all the matches between the Original Dataset and the Synthetic Dataset over 1 other column. These matches will be depicted on a bar plot via histogram. The idea behind this Privacy Metric is looking for **“values”** within the Synthetic Dataset which are also within the Original Dataset. If there are many of these values, this means that the GAN model is not able to generate a Synthetic Dataset similar to the Original one, but preserving all the values within the Original Dataset, and so preserving its privacy.

- We performed **2 Privacy Metrics Tests**: In the **Privacy Metrics Test 1** we took the field **ANTENNA_CODE** as first field for the merge and **SIM_COSE** as second feature for the final match. Viceversa, in the **Privacy Metrics Test 2** we chose **SIM_COSE** as first field and **ANTENNA_CODE** as second field.

- **Aggregated Privacy Metric (APM**): Eventually, in order to have a final aggregate measure of privacy (**APM**) in the interval [0, 1] we calculated a normalized sum of all matches. More matches means less privacy, if 1 - (normalized sum) is equal to 1 means high privacy, 0 means low privacy. In our first test we achieved a value close to **0.987654320987654**3, which means very high privacy. In the second test we achieved **1.0** as aggregated value which means maximum privacy. The formula we adopted is reported in the next slide with all the results.

Privacy Metrics - Model Evaluation by Matching Common Values

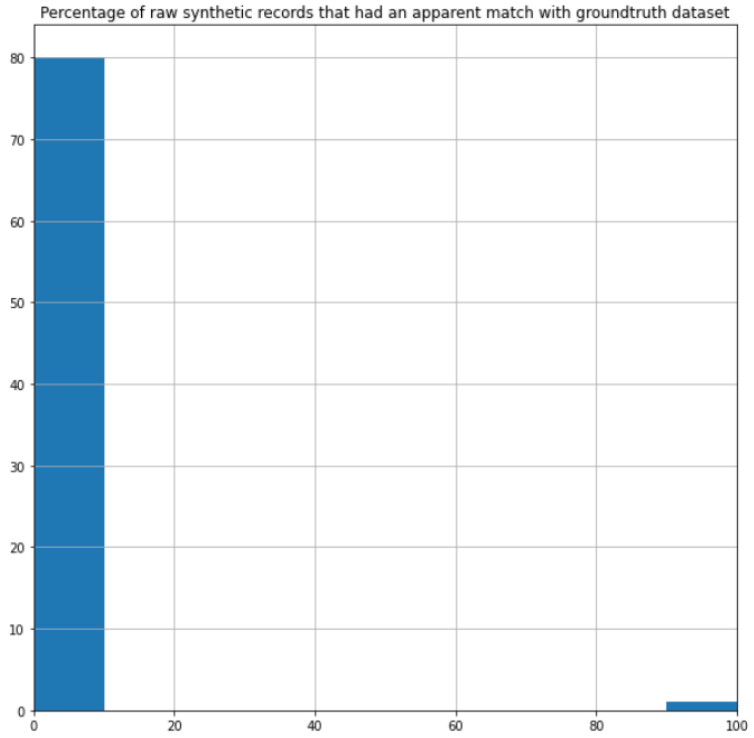
$$\mathbf{APM} = 1 - \frac{\sum_{k=0}^{n} \frac{S_k}{100}}{N}$$

The First Chart (**Test 1**) means that within the 81 rows obtained with the merge on **ANTENNA_CODE** there are 80 with 0% of matches on **SIM_CODE** and 1 with 100% of matches. So there is a small **failure** in the **Privacy Preserving Process**.

In the second chart there are no matches and failures.

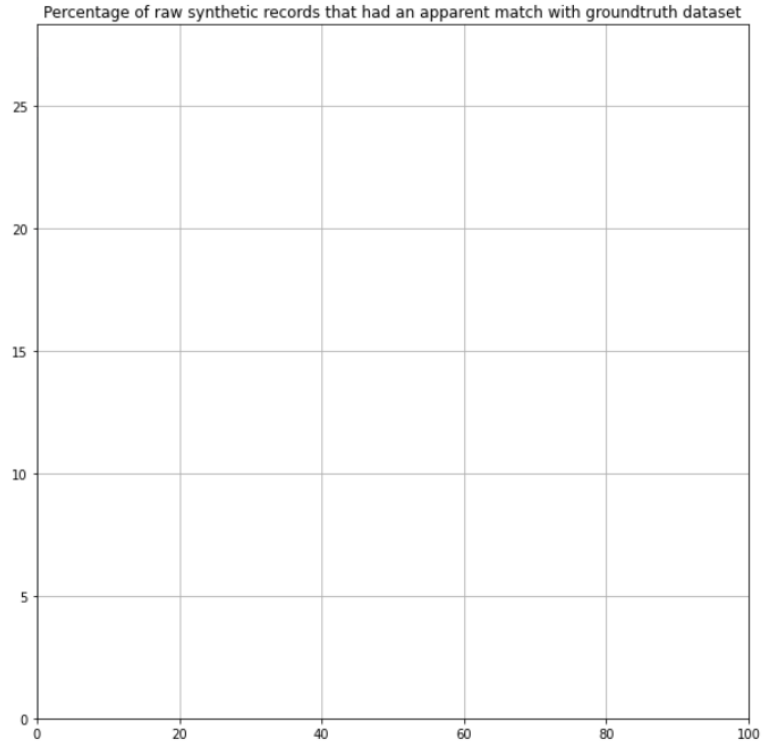
Privacy Metrics Test 1

Aggregate Data: 0.9876543209876543



Privacy Metrics Test 2

Aggregate Data: 1.0



Istat

Considerations and Future Directions

References

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Kunar, A. (2021). Effective and Privacy preserving Tabular Data Synthesizing. *arXiv preprint arXiv:2108.10064*.

Synthetic Data Gym Link: https://github.com/sdv-dev/SDGym

Colab Jupyther Notebook - Python Code Links of this Work:

Training Colab:
https://colab.research.google.com/drive/1v7UkWXnC_4qToHMB6QQWebUFTHJT6Qe4?usp=sharing

Inference and Metrics Evaluation Colab:
https://colab.research.google.com/drive/1FFX4ezYjwUbKiXGrqSuVV0iOr2wBakHa?usp=sharing

Thank You
for your attention

FABRIZIO DE FAUSTI | defausti@istat.it

MASSIMO DE CUBELLIS | decubell@istat.it

FRANCESCO PUGLIESE | francesco.pugliese@istat.it

ROBERTA RADINI | radini@istat.it

Istat | Istituto Nazionale di Statistica