BACUDA: BAnds of CUstoms Data Analytics

Synthetic Import Declaration Data and Hands-on-Workshop Summary

Sundong Kim (sundong@gist.ac.kr)

Assistant professor, GIST AI, Republic of Korea

Hands-on Data Workshop @ WCO PICARD 2022 Conference

Date: Dec 7, 10:00-15:00
Venue: WCO, Brussel
Speakers: Sundong Kim (GIST), Chaeyoon Jeong (KAIST)
Data generation with Jaewoo Park, Yeonsoo Choi (KCS)
Session-in-session: Sunmin Park (KCS), Sofia Douhdouh (Morocco Customs)
Support: Joel Choi, Sungsig Kim, Tetsuo Mizunuma, Thomas Cantens (WCO)

<Agenda>

Opening remarks (10:00-10:10, Thomas Cantens, Sungsig Kim)
Part 1: Hands-on-tutorial with the import declaration data (10:05-12:00) [Slides]

- Motivation for synthesizing the data: Sharing open data policy at Korea Customs Service (Sunmin Park)
- Hands-on-tutorial with the data (Sundong Kim, Chaeyoon Jeong)
    - GitHub: https://github.com/Seondong/Customs-Declaration-Datasets
    - Setup your laptop: Python and Jupyter Notebook
    - Exploratory Data Analysis: [Jupyter Notebook]
    - Synthesizing data with CTGAN: [Jupyter Notebook]
    - Application: Detecting frauds using AI models
        - Data preparation for fraud detection: [Jupyter Notebook]
        - Random Forest & XGBoost: [Jupyter Notebook]

Part 2: Invited session and discussion (14:00-15:00)

- Invited session: Morocco case: 15 min (Sophia)
- How to use the data: From the perspective of the technical side and policy side
    - Can these synthetic datasets be used as a learning resource, and okay to be shared?
    - What should be done to achieve good collaboration between academia, industry, and government agencies?
    - How to put lead research into practical use?
    - Synthesizing unstructured data (Image, text data, etc)
- Further discussion on research needs
- Short survey: [Forms]

GitHub repository: https://github.com/Seondong/Customs-Declaration-Datasets

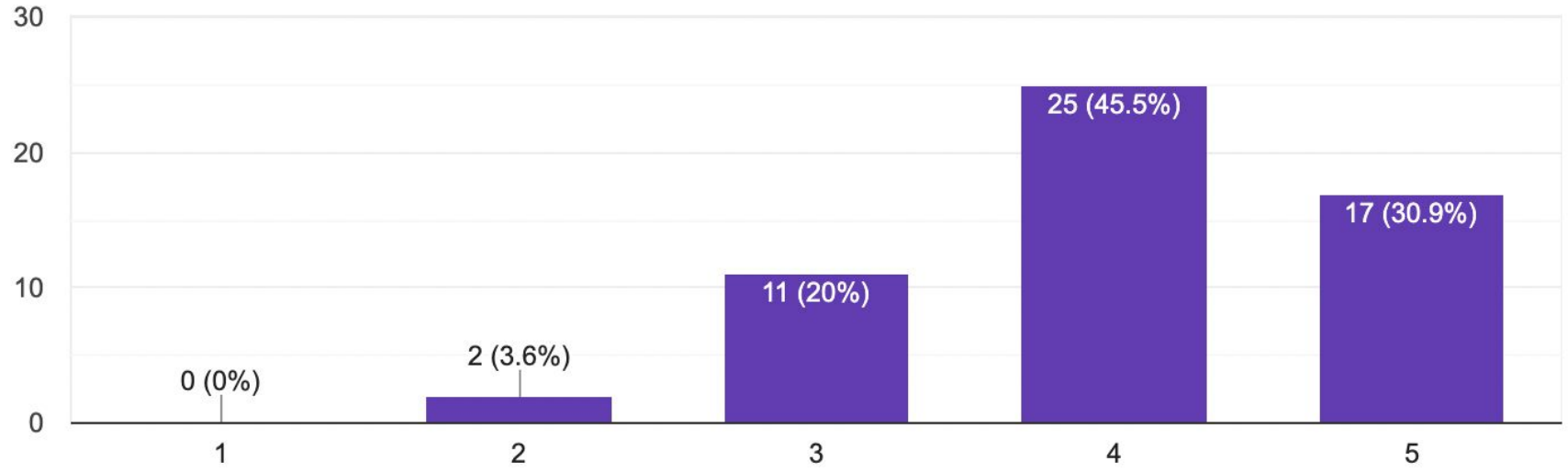
Paper: https://arxiv.org/abs/2208.02484

See more information at https://ds.ibs.re.kr/bacuda and https://sundong.kim.

Copy

Did the training Content meet your expectations and needs?

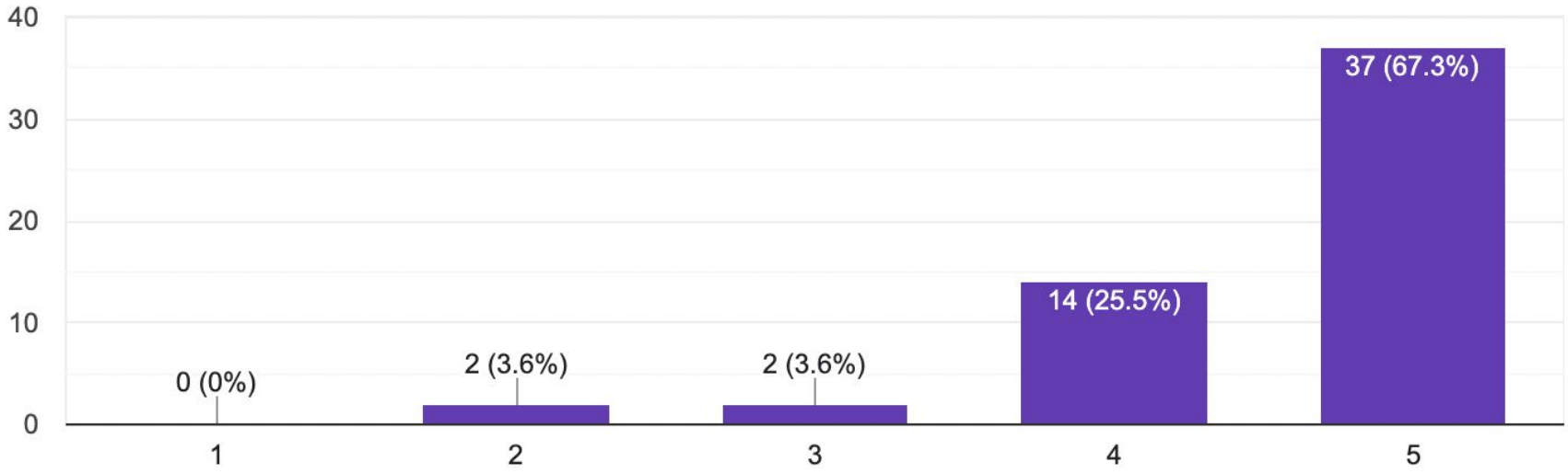
55 responses



Copy

Would you participate again if a similar session takes place next year?

55 responses



Do you have any other suggestions to improve this workshop?

55 responses

Develop tutorials

NA

No specifically. I am very satisfied.

No. It was superb! ;-)

Provide the details for the download of data etc before the session to reduce wasted time.

no thanks

Please share the set-up process along with the data and code in advance.

All is ok! But for me programming is something new, and so the practical implementation of such analysis at the moment is unpracticable

This is a very interesting WS. Thank you very much.

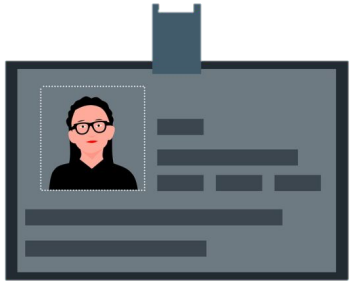
Open data policy in Korea

With the increasing importance of utilizing **data**…

- Complete real estate data
- National address data
- Food and drugs data
- Weather data
- Traffic data

"http://www.data.go.kr"

Privacy Concerns



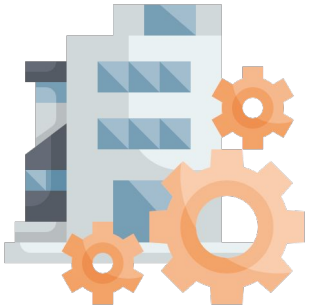
Personal info

Taxation Data

NOT for public disclosure

Needs for Synthetic Data

Pseudonymization
Is **NOT** enough



We need
SYNTHETIC DATA

For assess,
process,
analyze,
interpret…

Customs Import Declaration Datasets

Chaeyoon Jeong
KAIST
Daejeon, Republic of Korea
lily9991@kaist.ac.kr

Sundong Kim*
Institute for Basic Science
Daejeon, Republic of Korea
sundong@ibs.re.kr

Jaewoo Park
Korea Customs Service
Daejeon, Republic of Korea
jaeus@korea.kr

Yeonsoo Choi
Korea Customs Service
Daejeon, Republic of Korea
yschoi0817@korea.kr

ABSTRACT

Given the huge volume of cross-border flows, effective and efficient control of trades becomes more crucial in protecting people and society from illicit trades while facilitating legitimate trades. However, limited accessibility of the transaction-level trade datasets hinders the progress of open research, and lots of customs administrations have not benefited from the recent progress in data-based risk management. In this paper, we introduce an import declarations dataset to facilitate the collaboration between the domain experts in customs administrations and data science researchers. The dataset contains 54,000 artificially generated trades with 22 key attributes, and it is synthesized with CTGAN while maintaining correlated features. Synthetic data has several advantages. First, releasing the dataset is free from restrictions that do not allow disclosing the original import data. Second, the fabrication step minimizes the possible identity risk which may exist in trade statistics. Lastly, the published data follow a similar distribution to the source data so that it can be used in various downstream tasks. With the provision of data and its generation process, we open baseline codes for fraud detection tasks, as we empirically show that more advanced algorithms can better detect frauds.

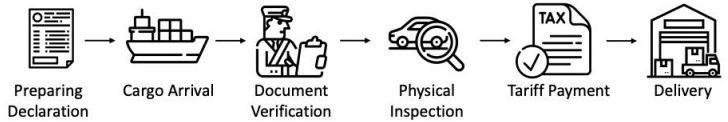


Figure 1: Import clearance process

This leads us to design synthetic data that can be open to the public. The dataset contained in this paper includes 54,000 artificially generated trades with 22 attributes. Using a tabular synthesizer with post-processing techniques, we maintain that the distribution and correlation among features in the synthetic dataset are similar to those of the source dataset. Empirical results on fraud detection demonstrate the usefulness of the data. Meanwhile, the data is used for competition in three universities to develop fraud-detection algorithms that can be applied in practice to detect illicit imports. We conclude the paper by discussing possible scenarios to use the data and summarizing necessary thoughts on the data synthesis. The data and code can be found in https://github.com/Seondong/Customs-Declaration-Datasets.

2 DATA DESCRIPTION

Import Declaration

(Customs Importation Certificate)

①Declaration No. Declaration ID	②Date of Declaration Date	③Customhouse/Section Office ID	⑥Date of Arrival			
④B/L(AWB) No.	⑤Cargo Control No.		⑦ Date of Warehousing	⑧Type of Tax Collection Payment Type		
⑨Declarant Declarant ID ⑩Importer Importer ID ⑪Taxpayer (Address) (Company Name) (Name) ⑫Trade agent ⑬Supplier Seller ID		Country of Origin, Country of Origin Indicator	⑭Type of Entry	⑱Origin Certificate	⑳Total Weight Net Mass	
			⑮Type of Entry-filing Process Type	⑲Price Statement	㉑Total Package No.	
			⑯Type of Transaction Import Type	㉒Port of Arrival	㉓Transport Mode Mode of Transport	
			⑰Purpose of Import Import Use	㉔Country of Loading	Country of Departure	
				㉕Vessel/Aircraft Name		
			㉖MASTER B/L No.		㉗Vessel/Aircraft Code Courier ID	
㉘Examination(Warehousing) Site						

https://bit.ly/PICARD22-dataset

Attribute	Value	Explanation
Declaration ID	97061800	Primary key of the record
Date	2020-01-01	Date when the declaration is reported
Office ID	13	Customs office that receives the declaration (e.g., Seoul regional customs)
Process Type	B	Type of the declaration process (e.g., Paperless declaration)
Import Type	11	Code for import type (e.g., OEM import, E-commerce)
Import Use	21	Code for import use (e.g., Raw materials for domestic consumption, from a bonded factory)
Payment Type	11	Distinguish tariff payment type (e.g., Usance credit payable at sight)
Mode of Transport	10	Nine modes of transport (e.g., maritime, rail, air)
Declarant ID	L77JJEG	Person who declares the item
Importer ID	HQ0W7JA	Consumer who imports the item
Seller ID	PBP2MYI	Overseas business partner which supplies goods to Korea
Courier ID	MWIDNS	Delivery service provider (e.g., DHL, FedEx)
HS10 Code	0901210010	10-digit product code (e.g., 090121xxxx = Coffee, Roasted, Not Decaffeinated)
Country of Departure	JP	Country from which a shipment has or is scheduled to depart
Country of Origin	JP	Country of manufacture, production or design, or where an article or product comes from
Country of Origin Indicator	B	Way of indicating the country of origin (e.g., B = Mark on package)
Tax Rate	8.0	Tax rate of the item (%)
Tax Type	A	Tax types (e.g., FTA Preferential rate)
Net Mass	1262.0	Mass without any packaging (kg)
Item Price	1437418.0	Assessed value of an item (KRW)
Fraud	1	Any fraudulent attempt to reduce the customs duty? (0/1)
Critical Fraud	1	Critical case which may threaten the public safety (0/1)

```
# Create data as many as the number of sampled data
count_row = df_sample.shape[0]
df_syn = ctgan.sample(count_row)
```

```
df_syn
```

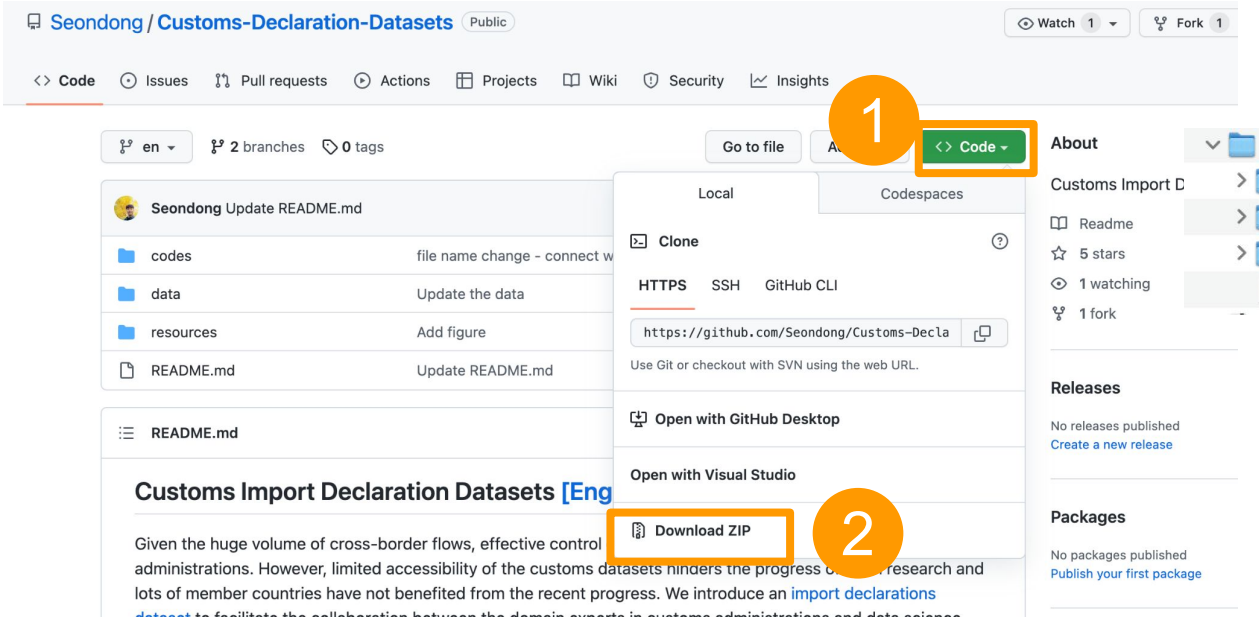
	Declaration ID	Date	Office ID	Process Type	Import Type	Import Use	Payment Type	Mode of Transport	Declarant ID	Importer ID	Seller ID	Courier ID	Country of Origin Indicator	Aggregated
0	37508957	2020-02-13	13	B	11	21	11	41	EAO05UQ	0MX22NJ	5G06UT6	NaN	E	6307909000^CN^CN^10.0^A^20.0^0^0
1	40789470	2020-03-05	39	B	11	25	15	41	P0PG8TK	0H3KUE6	WRC110G	NaN	E	8525802090^US^CN^0.0^A^11.8^1^1
2	77409982	2020-01-26	30	B	10	21	45	42	4QADEO0	3I7CR54	PMN0COE	NaN	G	710807000^CN^CN^27.0^A^100000.0^0^0
3	72078802	2021-03-24	31	B	11	21	19	11	VCV1EJF	1VHJG03	XBU3AO0	MWIDNS	B	8542311000^US^MX^0.0^CIT^0.1^0^0
4	82802220	2020-05-04	29	B	10	21	12	41	VBOQV8H	XIDTSU2	2100L9Z	NaN	G	3908103000^JP^JP^6.5^C^20.0^0^0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2995	49957284	2020-10-14	20	B	11	21	45	11	OZB7KED	W0RGTOU	SVODT0Z	NaN	E	6304990000^CN^CN^5.2^FCN1^600.0^0^0
2996	31063869	2021-03-07	20	B	11	21	44	10	FAG093P	SU59N3V	IWSDF6W	NaN	B	4911100000^JP^JP^0.0^C^0.5^0^0
2997	82189653	2020-11-24	30	B	10	21	44	55	QZE38LM	5EJ36OK	6ZQTY16	NaN	E	6214900000^CN^CN^3.2^FCN1^53.0^1^1
2998	21998474	2020-06-10	30	B	11	21	44	11	CIN0OY4	D8DTWCP	J52UCA9	MWIDNS	B	8538909000^US^DE^8.0^A^1.0^1^1
2999	14398653	2020-09-02	20	B	15	21	43	41	HHBAZKA	98VUP2X	VXIMC4A	NaN	Y	6204530000^CN^CN^13.0^A^78.0^0^0

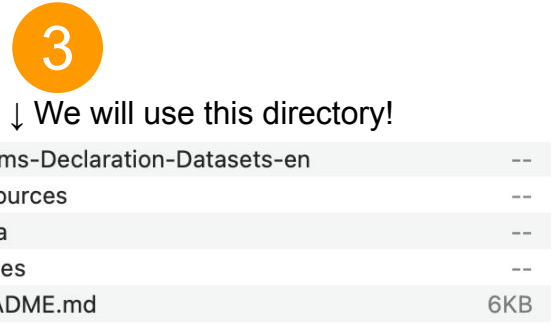
3000 rows × 14 columns

How to use?

Download the Data & Codes

1. Go to : https://github.com/Seondong/Customs-Declaration-Datasets
2. Click Code -> Download ZIP
3. Unzip the downloaded ZIP file

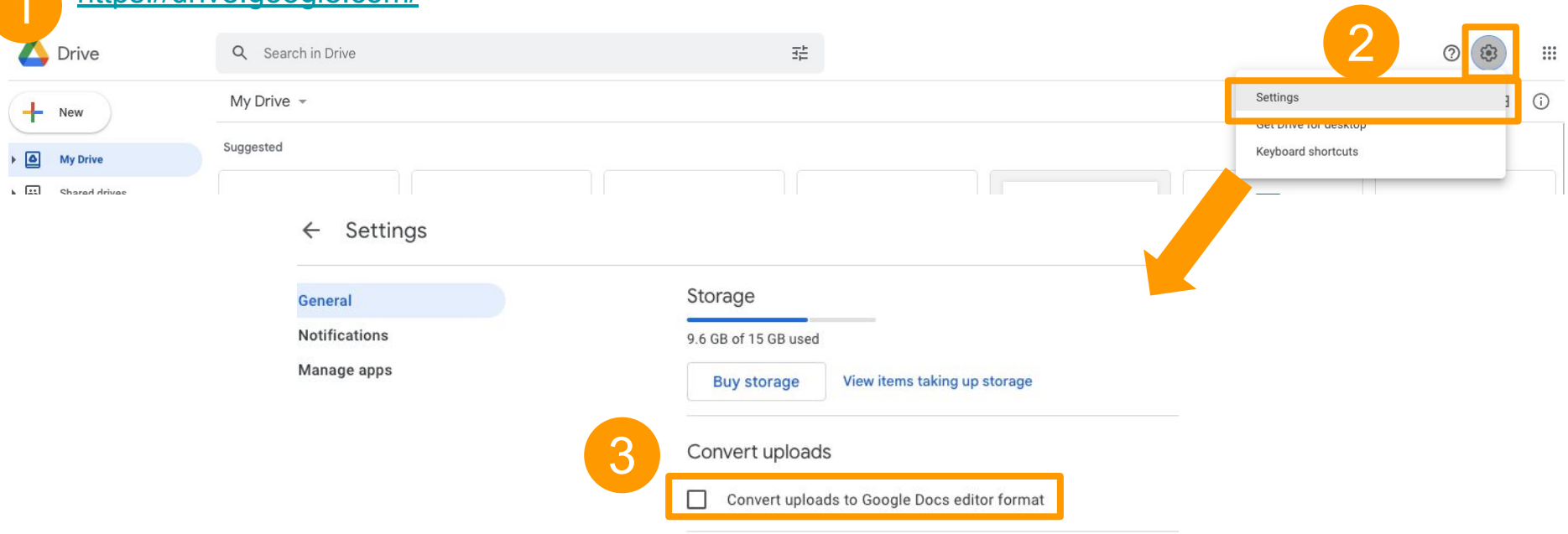




Upload the Data & Codes to Google Drive

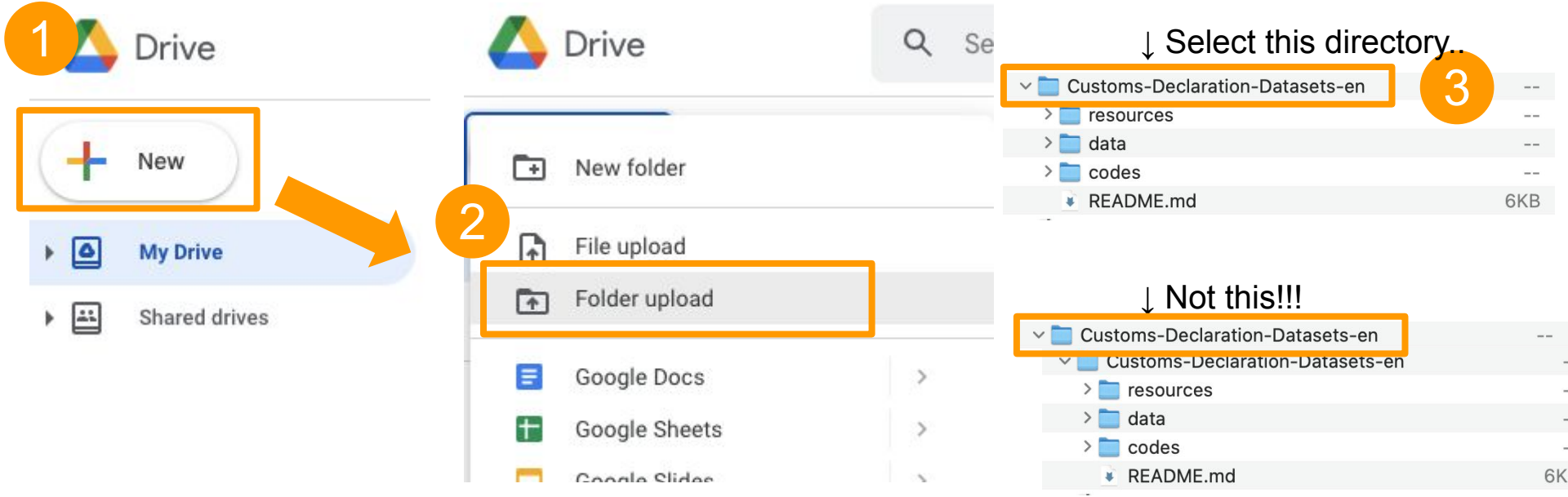
4. Open Google Drive -> Click Settings -> ***Uncheck*** “Convert uploads ~~”

1 https://drive.google.com/



Upload the Data & Codes to Google Drive

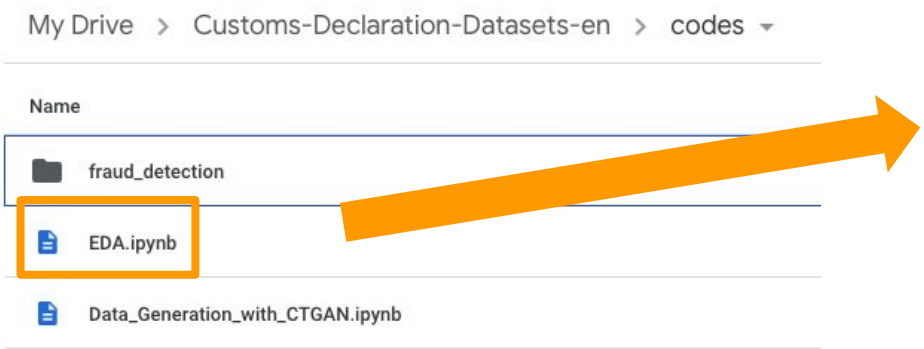
5. Return to Google Drive main page -> Click “New” -> “Folder upload”
6. Select Customs-Declaration-Datasets-en folder (which is directly contains /resources, /data, /codes) to upload



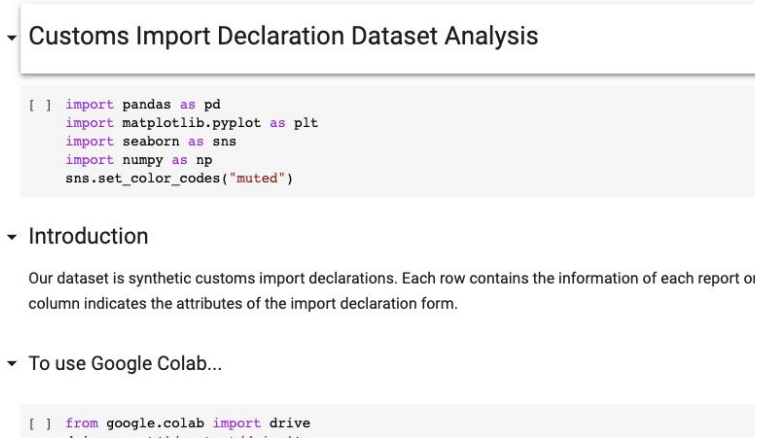
Open .ipynb file

7. Open Customs-Declaration-Datasets-en/codes/EDA.ipynb

If you already have Colaboratory installed, this page will open automatically



EDA.ipynb ⤵

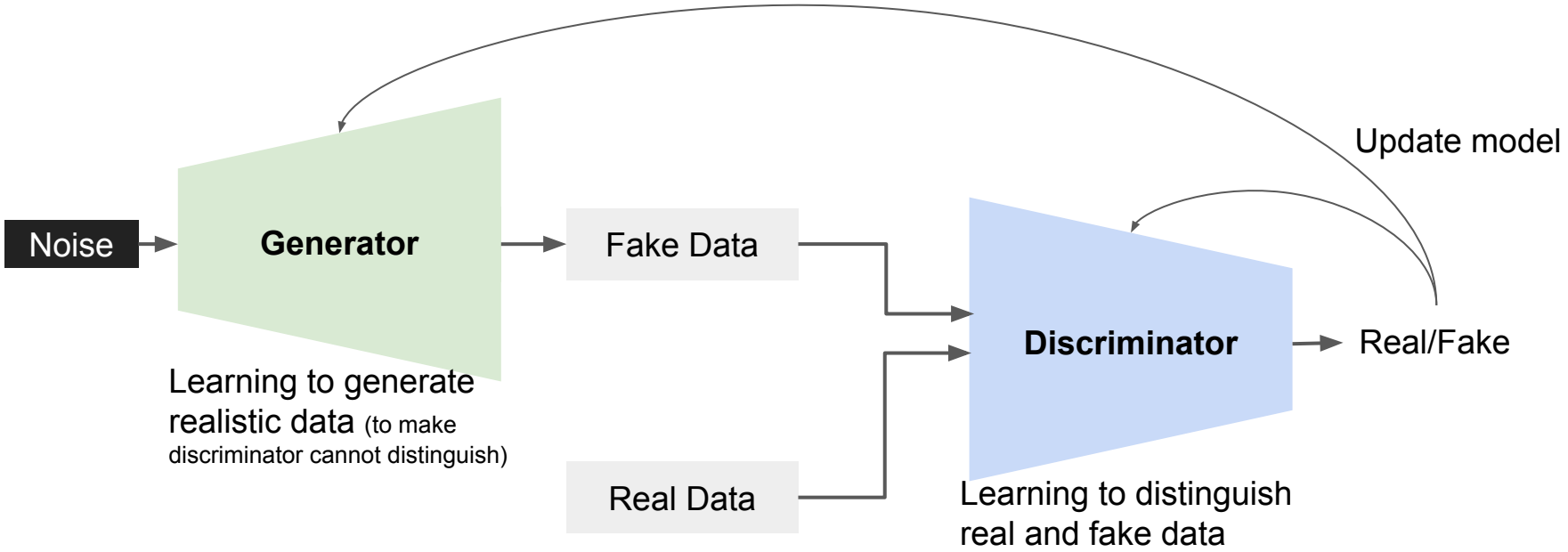


.ipynb = IPython Notebook file

Techniques for Generating Synthetic Data

CTGAN: Conditional GAN for Tabluar data

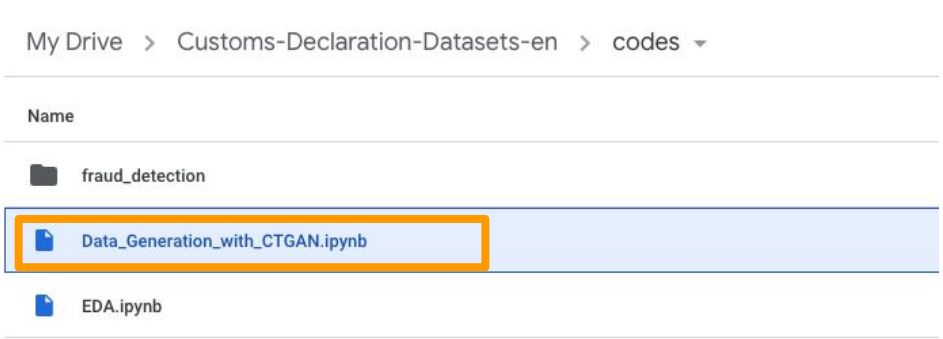
GAN: Generative Adversarial Network

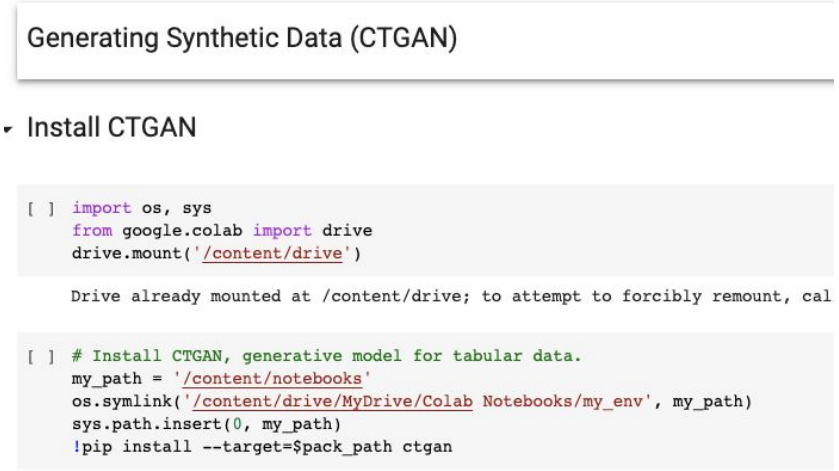


L Xu et. al., Modeling Tabular Data using Conditional GAN, NeurIPS 2019

Open Data Generation Code

Open codes/Data_Generation_with_CTGAN.ipynb

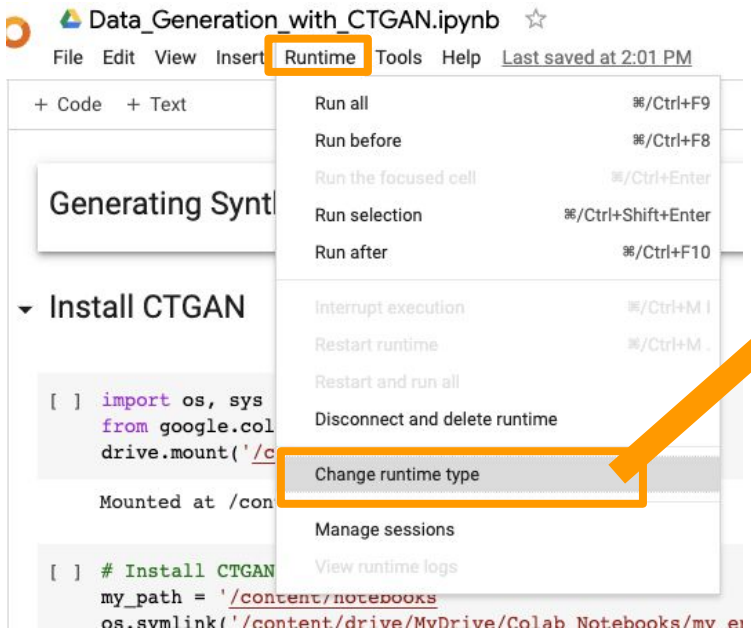


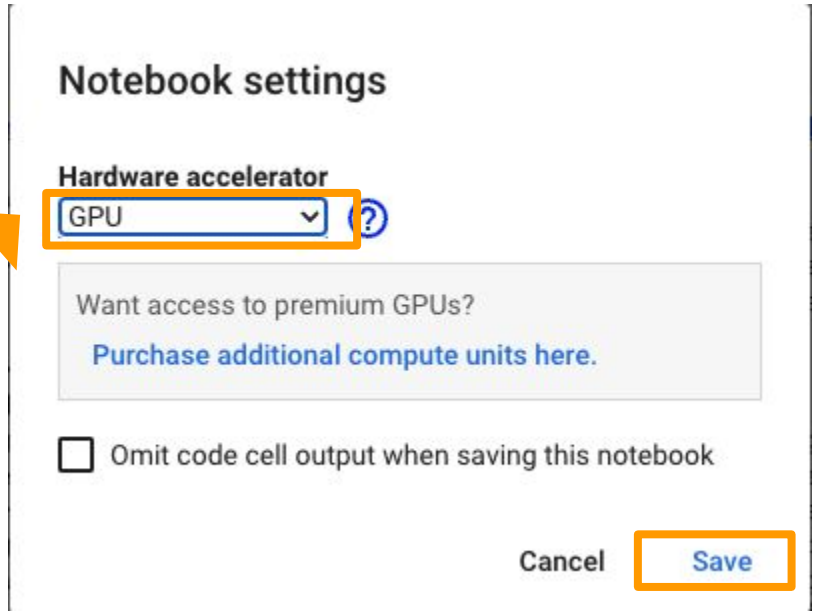


Setting GPU to Run CTGAN

Click “Runtime” → “Change runtime type”

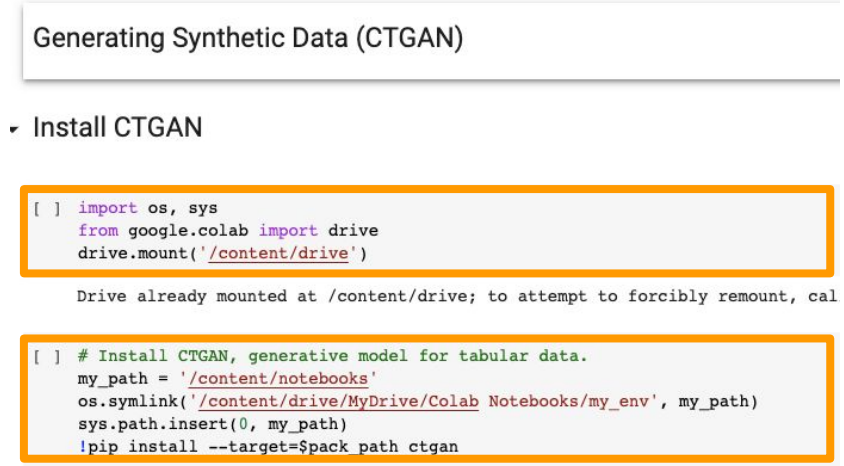
Set Hardware accelerator as **GPU** → click “Save”





Install CTGAN

Run the first two cells by clicking the first cell → press ctrl+Enter (or command+Enter)



Mounting Google drive

Install CTGAN module
(This code installs CTGAN on Google drive, not on your local device)

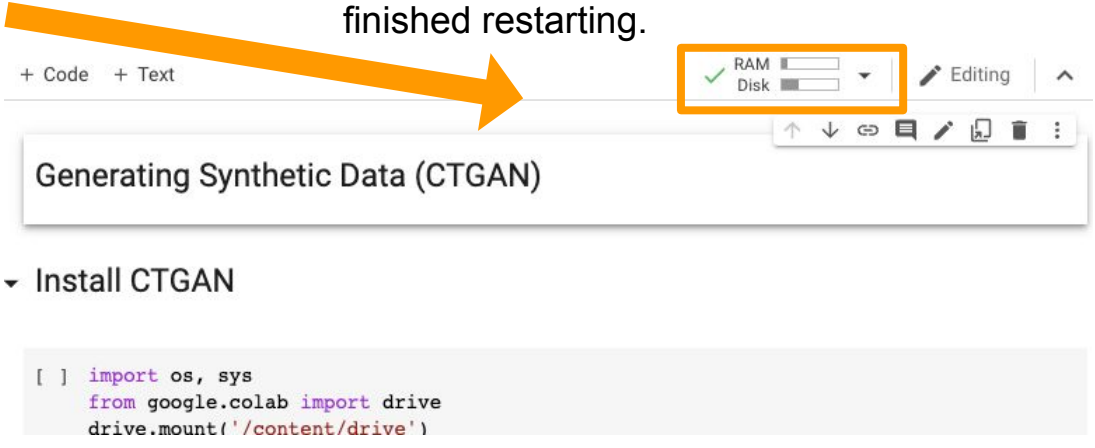
```
# Install CTGAN, generative model for tabular data.
my_path = '/content/notebooks'
os.symlink('/content/drive/MyDrive/Colab Notebooks/my_env', my_path)
sys.path.insert(0, my_path)
!pip install --target=$pack_path ctgan
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting ctgan
  Downloading ctgan-0.6.0-py2.py3-none-any.whl (24 kB)
Requirement already satisfied: numpy<2,>=1.20.0 in /usr/local/lib/python3.7/dist-packages (from ctgan) (1.21.6)
Requirement already satisfied: torch<2,>=1.8.0 in /usr/local/lib/python3.7/dist-packages (from ctgan) (1.12.1+cu113)
Collecting rdt<2.0,>=1.2.0
  Downloading rdt-1.2.1-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████████████| 61 kB 292 kB/s
Requirement already satisfied: pandas<2,>=1.1.3 in /usr/local/lib/python3.7/dist-packages (from ctgan) (1.3.5)
    Successfully uninstalled psutil-5.4.8
Successfully installed Faker-15.3.2 ctgan-0.6.0 psutil-5.9.4 pyyaml-5.4.1 rdt-1.2.1
WARNING: The following packages were previously imported in this runtime:
  [psutil]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

If you get this message, click “RESTART RUNTIME”
The code will be restarted in a few seconds

When you see this again, it means the code finished restarting.



Data Preprocess

1. Select columns that are import in administrative process
2. Randomly select 3000 datapoints
3. Aggregate columns into a single column to preserve relation between columns (Columns you want to strongly preserve)

▾ Preprocess Source Data

```
[ ] %cd drive/MyDrive/Customs-Declaration-Datasets-en/

    /content/drive/MyDrive/Customs-Declaration-Datasets-en

[ ] # Load train data
    df_raw=pd.read_csv('./data/df_syn_eng.csv', encoding='utf-8-sig')

[ ] # Select columns to use
    df_org=df_raw[["Declaration ID", "Date", "Office ID", "Process Type", "Import Type", "Import Use", "Payment Type",
                   "Mode of Transport", "Declarant ID", "Importer ID", "Seller ID", "Courier ID",
                   "HS10 Code", "Country of Departure", "Country of Origin",  "Tax Rate", "Tax Type",
                    "Country of Origin Indicator", "Net Mass", "Item Price", "Fraud", "Critical Fraud"]]

[ ] # Since CTGAN cannot handle large input size, we sample 3000 Train Data from Source Data
    df_sample=df_org.sample(3000, replace=False)
    df_sample.to_csv('./data/df_sample.csv', index=False, encoding='utf-8-sig')

[ ] # Aggregate Reletive Columns
    cols = ['HS10 Code', 'Country of Departure', 'Country of Origin', 'Tax Rate', 'Tax Type', 'Net Mass','Fraud', 'Critical Fraud']
    df_sample['Aggregated'] =df_sample[cols].apply(lambda row: '^'.join(row.values.astype(str)), axis=1)
    df_sample=df_sample.drop(cols, axis=1)
    df_sample=df_sample.drop(['Item Price'], axis=1)

[ ] df_sample['Date']=df_sample['Date'].astype('str')
```

Train CTGAN

Give the list of categorical columns to the CTGAN model

Train for 100 epochs (In our paper, we trained for 300 epochs)

Epochs = number of training rounds

```
[12] categorical_columns =['Declaration ID', 'Date','Process Type','Declarant ID','Importer ID','Seller ID',
                           'Courier ID','Country of Origin Indicator','Aggregated']
```

```
[13] # It will take around 5~10 min for training 100 epochs.
     ctgan = CTGAN(verbose=True)
     ctgan.fit(df_sample, categorical_columns, epochs = 100)
```

```
Epoch 43, Loss G:  4.8997,Loss D: -0.0039
Epoch 44, Loss G:  4.9795,Loss D: -0.0646
Epoch 45, Loss G:  4.5358,Loss D: -0.0081
Epoch 46, Loss G:  4.5927,Loss D:  0.0165
Epoch 47, Loss G:  4.9038,Loss D:  0.0322
Epoch 48, Loss G:  4.8420,Loss D: -0.0273
Epoch 49, Loss G:  4.9053,Loss D: -0.0695
Epoch 50, Loss G:  4.7931,Loss D:  0.0114
Epoch 51, Loss G:  4.9860,Loss D:  0.0196
Epoch 52, Loss G:  4.6584,Loss D:  0.0009
Epoch 53, Loss G:  5.2062,Loss D: -0.1014
Epoch 54, Loss G:  4.8563,Loss D: -0.1061
Epoch 55, Loss G:  4.7988,Loss D:  0.0176
Epoch 56, Loss G:  4.7786,Loss D:  0.0136
Epoch 57, Loss G:  4.6389,Loss D: -0.1083
```

Generate Synthetic Data

Generate 3000 datapoints from trained CTGAN

Each rows have the same format as the training data

```
# Create data as many as the number of sampled data
count_row = df_sample.shape[0]
df_syn = ctgan.sample(count_row)
```

```
df_syn
```

	Declaration ID	Date	Office ID	Process Type	Import Type	Import Use	Payment Type	Mode of Transport	Declarant ID	Importer ID	Seller ID	Courier ID	Country of Origin Indicator	Aggregated
0	37508957	2020-02-13	13	B	11	21	11	41	EAO05UQ	0MX22NJ	5G06UT6	NaN	E	6307909000^CN^CN^10.0^A^20.0^0^0
1	40789470	2020-03-05	39	B	11	25	15	41	P0PG8TK	0H3KUE6	WRC110G	NaN	E	8525802090^US^CN^0.0^A^11.8^1^1
2	77409982	2020-01-26	30	B	10	21	45	42	4QADEO0	3I7CR54	PMN0COE	NaN	G	710807000^CN^CN^27.0^A^100000.0^0^0
3	72078802	2021-03-24	31	B	11	21	19	11	VCV1EJF	1VHJG03	XBU3AO0	MWIDNS	B	8542311000^US^MX^0.0^CIT^0.1^0^0
4	82802220	2020-05-04	29	B	10	21	12	41	VBOQV8H	XIDT5U2	2100L9Z	NaN	G	3908103000^JP^JP^6.5^C^20.0^0^0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2995	49957284	2020-10-14	20	B	11	21	45	11	OZB7KED	W0RGTOU	SVODT0Z	NaN	E	6304990000^CN^CN^5.2^FCN1^600.0^0^0
2996	31063869	2021-03-07	20	B	11	21	44	10	FAG093P	SU59N3V	IWSDF6W	NaN	B	4911100000^JP^JP^0.0^C^0.5^0^0
2997	82189653	2020-11-24	30	B	10	21	44	55	QZE38LM	5EJ36OK	6ZQTY16	NaN	E	6214900000^CN^CN^3.2^FCN1^53.0^1^1
2998	21998474	2020-06-10	30	B	11	21	44	11	CIN0OY4	D8DTWCP	J52UCA9	MWIDNS	B	8538909000^US^DE^8.0^A^1.0^1^1
2999	14398653	2020-09-02	20	B	15	21	43	41	HHBAZKA	98VUP2X	VXIMC4A	NaN	Y	6204530000^CN^CN^13.0^A^78.0^0^0

3000 rows × 14 columns

How can we practically adopt new technology?

- Can these synthetic datasets be used as a learning resource, and okay to be shared?
- What should be done to achieve good collaboration between academia, industry and government agencies?
- How to put lead research into practical use

https://sundong.kim/