Synthetic Tabular Data Generation

Roshan Pandey
School of Computing
Newcastle University
Newcastle Upon Tyne, UK
r.k.pandey2@newcastle.ac.uk

Prof. Nic Palmarini
Population Health Sciences
Newcastle University
Newcastle Upon Tyne, UK
nic.palmarini@newcastle.ac.uk

Abstract

Generative adversarial networks (GANs) have been extensively used to generate synthetic image data and it does really well in that regard but what about tabular data? The purpose of this study is to see how well GAN-based framework like Synthetic Data Vault (SDV) which uses CTGAN and other variations of GAN under the hood would perform on different encoding techniques like *count* encoding, *spss* encoding for categorical data. This study also evaluates which method is better, whether to merge multiple tables into one and then generate the synthetic data using SDV or generate data separately and then merge them together. After all the testing, it was found that when there are multiple tables in a dataset it is better to merge tables in the required format and then generate synthetic data. Also, spss encoding for categorical data worked better than count encoding. *SPSS* encoded data achieved *KSTest* score of 0.785 This is one of its kind work which made extensive use of SDV on behavioural data which is highly randomised and has many different distributions and at the same time tried various configurations of the data such as different encoding techniques and merging of different tables.

***K*eywords** Synthetic Data Generation · SDV · Random Forest · DALL-E · Imagen · GAN · CTGAN · GaussianCoupla · CouplaGAN · TVAE · Cycle-GAN · KSTest

1 Introduction

The modern world is running on data and 2.5 quintillion bytes of data is being generated every day. However, *"With great power comes great responsibility"* especially when it comes to protecting people's privacy. This issue can be solved by generating synthetic data which will have a close correlation with the original data. Another issue is that even though more and more data is being generated every day there are certain sectors like medical, health, charity etc. where there is a scarcity of data. So, to help such sectors synthetic data can be generated from the small set of data which is already available.

This project uses The UK Time Diary Study 2014 - 2015 data collected from the UK Data Services [8] which is open source and it includes dairy data of 9388 individuals from randomly selected 4238 households generating a total of 16533 unique records. Data is consist of what each individual in a family does at a certain point in time of a day, they have recorded their activities every 10 minutes throughout the day. It also consists of demographic information about the individuals. All the files are in *spss* format and all the categorical variables are represented as numeric codes and in the metadata file corresponding labels have been mentioned. For generating synthetic tabular data, Synthetic Data Vault (SDV) [1] is been used. Apart from *spss* encoding, count encoding has also been used just to see if it makes any difference and to check the reliability of the synthetic data, a correlation-heatmap is generated. Also, a regression model was trained on real and synthetic data to see whether there is any shift in error rate from real data to synthetic data.

Figure 1: Image Generated by DALL-E

2 Objective(s)

As the dataset contains sensitive data such as demographic data of people and even though the data has been sanitised (personal details like name, phone number, and email address has been removed) there will always be some kind of privacy issues hence, the main objective of this work is to generate synthetic data from this existing behavioural data using Synthetic Data Vault (SDV) [1] which will protect the privacy of the individuals and at the same time, the data will keep the same statistical consistency as that of real data so that reliable analysis can be carried out or used for training machine learning/deep learning models with reasonable reliability.

Apart from privacy issues, synthetic data generation solves the problem of having lack of data to train deep learning models as deep neural nets are very data hungry and sectors like health and medical often tend to face the issue of lack of data.

3 Related Work

Generating synthetic data has been a popular technique in the recent past however most of the existing work has been carried out for images and some of the state-of-the-art image generators are mentioned below:

DALL-E [10] is a text-to-image conversion Transformer based multi model developed by OpenAI which takes in natural language as input and generates a digital image. See figure 1. It uses a version of GPT-3 in the back-end to extract the features like concepts, attributes and style. DALL-E was trained on 400 million pairs of images and their text captions taken from the internet. DALL-E has the ability to generate images in various styles like painting, photo-realistic, emojis etc. and also it can place subjects in certain orientations such as the "rule of thirds" without being explicitly told.

Recently Google also released its own text-to-image generation model "Imagen" [4] which uses text transformer model T5 for text embedding and after that diffusion models for both generation of small images and then turn them into high-resolution images. See figure 2. Google believes Imagen is different from other image generation models because of 2 reasons. The first is its deep understanding of natural language and the second is "unprecedented photorealism".

Apart from text-to-images, there are some Generative Adversarial Networks (GAN) [11] [14] based models which can do image-to-image conversions such as Cycle GAN [15], which can convert one type of image to another for example horse images to zebra images and vice versa. See figure 3

As for tabular data, Tabular GAN [2] [3] is one such framework which can generate synthetic tabular data and it is also based on Synthetic Data Vault (SDV) [1]. Tabular GAN does fairly well to generate synthetic tabular data. In this work KDD99 and Covertype datasets have been used. These datasets are not very complex and do possess patterns within the dataset which makes the model pick up features relatively easily. Also, there are several paid platforms as well which help businesses to generate synthetic data. One of them being AINDO [20].

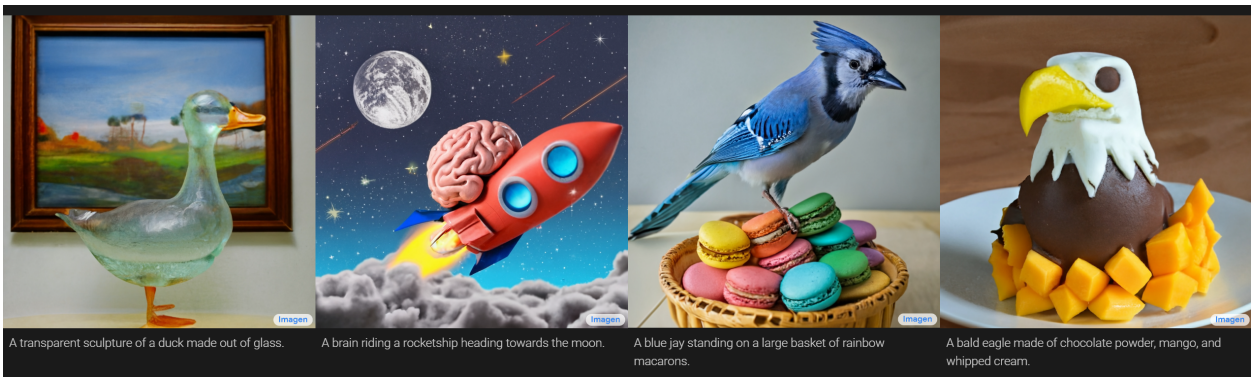


Figure 2: Image Generated by Imagen

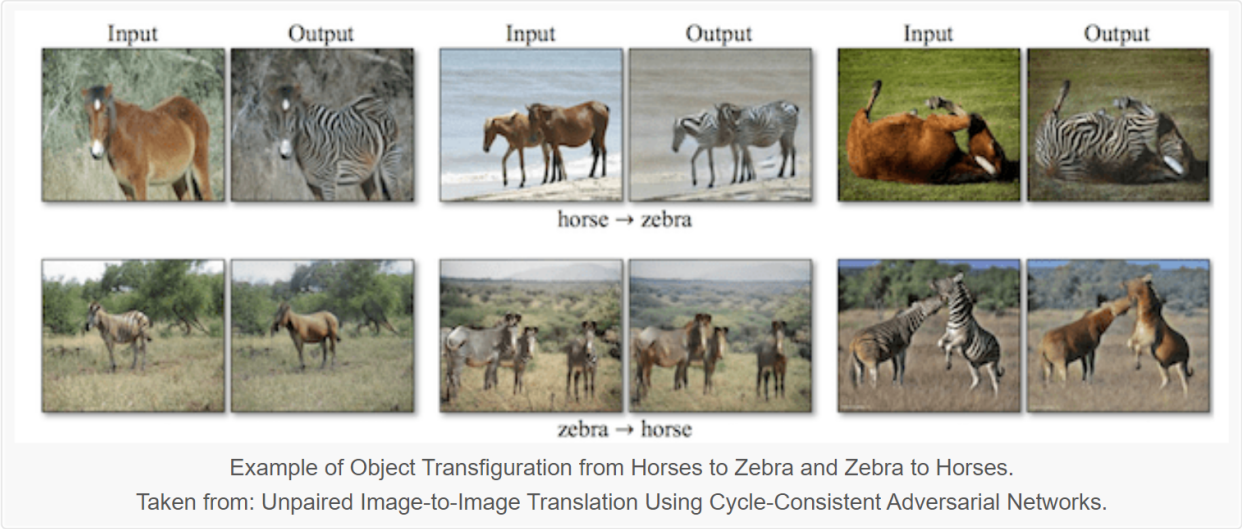


Figure 3: Image Generated by Cycle-GAN

4 Methodology

The figure 4 shows the overall workflow of this work, right from data collection to selecting the best synthetic data that has close statistical consistency to that of real data. Blue boxes represent processes, red represents operations, and the green represents data.

4.1 Business Understanding

As discussed in the objectives section 2 the dataset contains respondents' sensitive personal data hence there is a risk of privacy breach which needs to be taken care of. So, to tackle this issue, synthetic data generation [19] is one solution where the synthetic data that possesses a high correlation with the actual data needs to be generated so that reliable analysis can be performed on it without having to worry about privacy issues and with that, this work solves the issue of lack of data as well.

To evaluate the consistency of the synthetic data, the following steps have been taken:

- Heatmap of correlation values of real and synthetic data in different configurations has been plotted to see the closeness of pattern in both the plots.
- Now the synthetic data that very closely matches the real data is selected and the log of mean and standard deviation, the cumulative sums [18] per feature, and distribution [17] per feature are plotted.

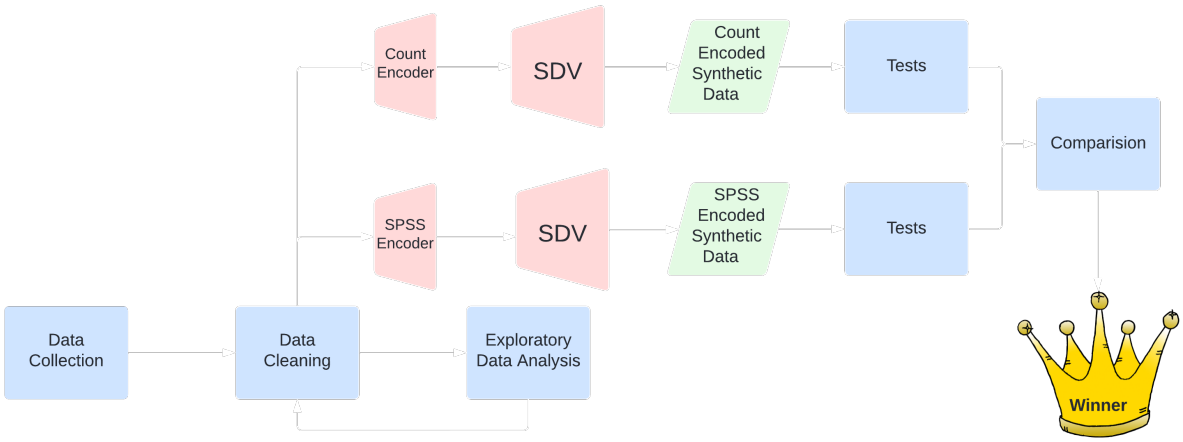


Figure 4: Methodology

- After this, regression models were trained on real and synthetic data to predict how much time a person would spend on a particular task at a given point of time in a day. If the error rate is similar or close in both the models then we can state that synthetic data has the consistency of the real data.

4.2 Data collection

The data collection process was designed and organized by the authors of the book "The Centre for Time Use Research" [7] in 2014-15 and it was funded by ESRC [24]. The data is publicly available on UK Data Services [8]. 2014-15 Time Use Survey [6] [7] is the most recent survey in the UK historical sequence of time-use diary surveys and it formed the UK's contribution to the Harmonised European Time Use Survey (HETUS) [23] programme and followed its guidelines.

The main idea behind the UK 2014-15 Time Use Survey [6] [7] is to record how people in the United Kingdom spend their time. Respondents are asked to document what they are doing, with whom they are spending their time, do they enjoy doing things that they do, etc. See Figure 5. For this exercise, the data was collected for a single day and the whole day was divided into 10 minutes time intervals. This data can be considered as continuous data or logs but it can also be treated as the sequence of discrete events. This turned out to be an effective way to collect behavioural data and give an idea of how people in the UK like to spend their time.

4.3 Data Cleaning

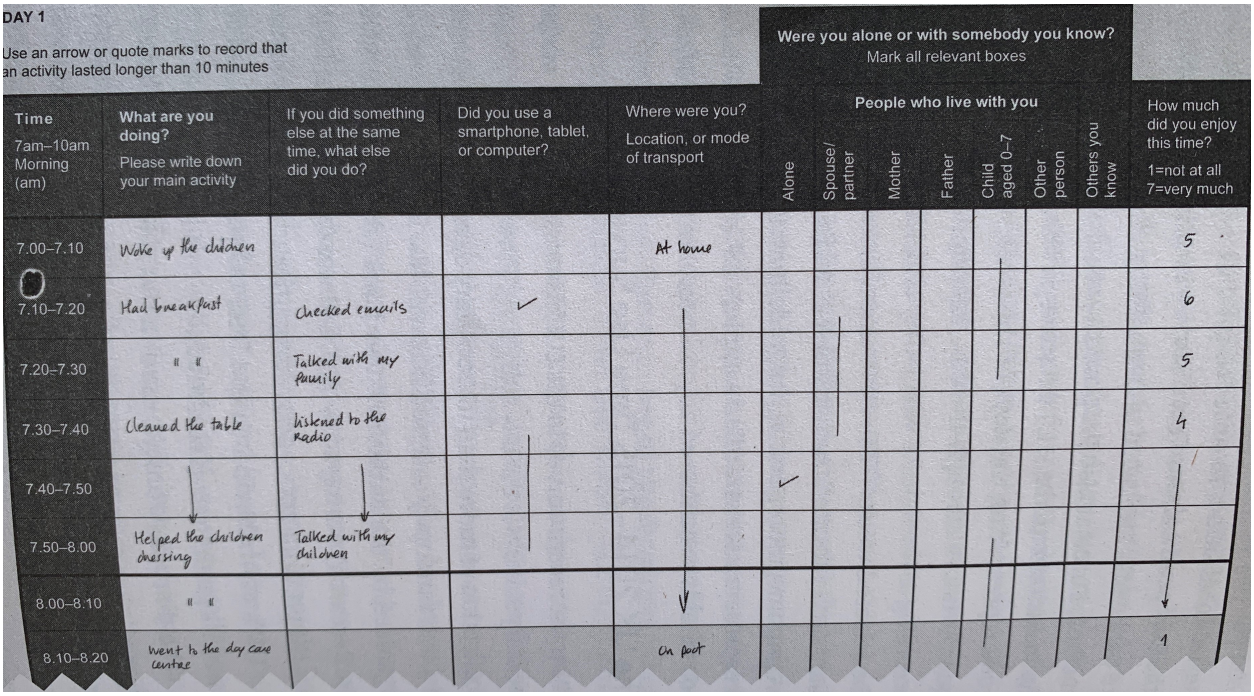
The dataset contains multiple files in .sav (spss) format. For the purpose of this work, 2 files are being used. They are described as below:

- Activity file: This file contains the data of the activity people are performing at intervals of 10 mins for 24 hrs period time, how much they are enjoying the task they are doing, which place the task was started and ended, with whom (friends and family members) the task was performed and some metadata about the nature of the interview, date and time when the diary entry was recorded.
- Individual info file: This file consists of demographic data of the respondents such as age, gender, employment status, income, number of people in the household, marital status, what appliances they have in their house etc.

Most of the columns have null values in the individual info file hence only relevant columns have been selected manually and some of the null values from those columns have been dropped. There are some potential outliers in the data but there is no concrete evidence of them being outliers hence they are not dropped.

4.4 Data Exploration and Manipulation

The activity log file is in long format i.e., the time-stamp of each activity is in column format and there are 587632 data points and 50 features. In the individual info table, there are 11421 rows and 603 columns.



DAY 1

Use an arrow or quote marks to record that an activity lasted longer than 10 minutes

Time 7am–10am Morning (am)	What are you doing? Please write down your main activity	If you did something else at the same time, what else did you do?	Did you use a smartphone, tablet, or computer?	Where were you? Location, or mode of transport	Were you alone or with somebody you know? Mark all relevant boxes: Alone	Spouse/ partner	Mother	Father	Child aged 0–7	Other person	Others you know	How much did you enjoy this time? 1=not at all 7=very much
7.00–7.10	Woke up the children			At home								5
7.10–7.20	Had breakfast	Checked emails	✓									6
7.20–7.30	" "	Talked with my family										5
7.30–7.40	Cleaned the table	Listened to the radio										4
7.40–7.50	↓	↓			✓							↓
7.50–8.00	Helped the children dressing	Talked with my children										
8.00–8.10	" "			↓								↓
8.10–8.20	Went to the day care centre			On foot								1

Figure 5: Time-Use Survey

In both the tables, most of the columns are categorical and they are encoded using count encoder and *spss* encoding techniques. One-hot encoding has been avoided because large number of categories were present in most of the columns. One-hot encoding would have caused a curse of dimensionality. The data is cross-sectional data [21], it has been collected by observing a set of people over a period of time (24 hours).

The data that will be used to train regression model will use 14 features from activity log table ('dmonth', 'ddayw', 'WhereStart', 'WhereEnd', 'RushedD', 'KindOfDay', 'dia_wt_a', 'Trip', 'tid', 'Device', 'WhereWhen', 'whatdoing', 'eptime') and 7 from individual info table ('DMSex', 'WorkSta', 'DVAge', 'Income', 'Sector', 'NumChild', 'NumAdult'). Both these tables are merged using 'serial' and 'pnum' as they both together form the primary key.

The figure 10 shows the pair-plot of the activity log table, from which we can clearly infer that eptime (time spent on each task) does not have much correlation with other independent features.

The figure 11 is a pair-plot of the merged table which again does not show much relationship between the target variable (eptime) and independent variables hence from both the pair-plots we can suggest that data is highly randomised, non-linear, and it supports the fact that human beings are one of the most unpredictable creatures on the planet [5].

There are different configurations of the dataset used to generate synthetic data:

- Config 1: Activity log and individual info tables were encoded using *spss* encoder and generated separately then merged to train a regression model to predict time spent on a particular task.
- Config 2: Activity log and individual info tables were encoded using *spss* encoder then merged and then generated synthetic data.
- Config 3: Activity log and individual info tables were encoded using a count encoder and generated separately then merged to train a regression model to predict time spent on a particular task.
- Config 4: Activity log and individual info tables were encoded using a count encoder then merged and then generated synthetic data.

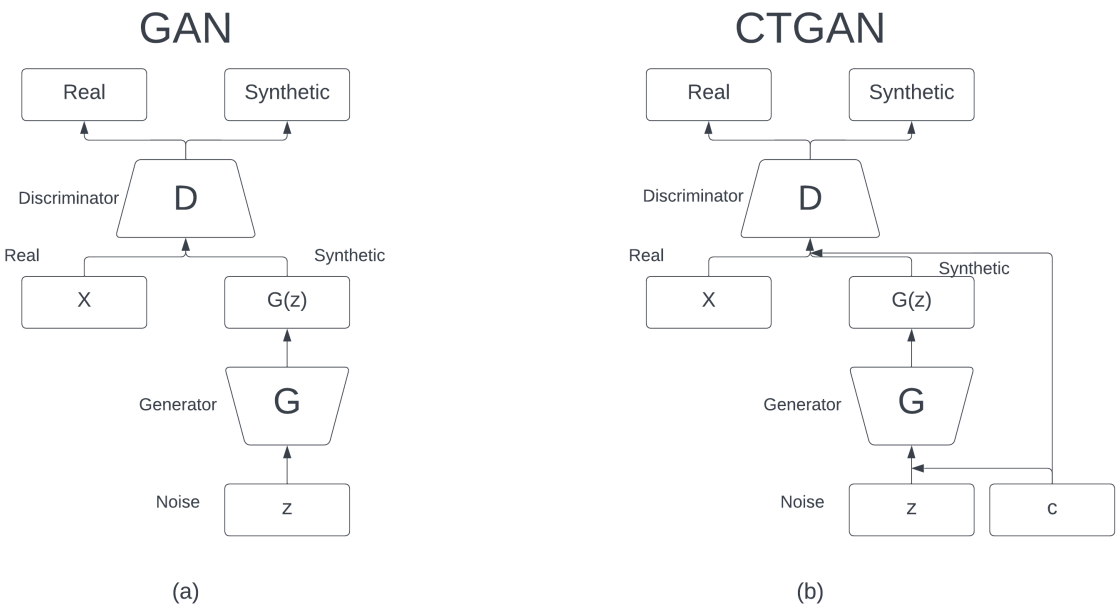


Figure 6: (a) GAN, (b) CTGAN

4.5 Modelling

4.5.1 Synthetic Data Generation

Initially a Generative Adversarial Network (GAN) [11] [14] was trained from scratch on the behavioural data just to see whether the network is able to pick different distributions and data types of the features and it was observed that model was generating values which had mean and standard deviation close to that of real data however, it was not taking the data types into consideration. For example if a column had all integer values and mean was around 30 and standard deviation was around 4, the generated data had mean approximately close to 30 and standard deviation close to 4, however, it was containing float values and the column could not possibly have a float value. To fix this issue Wasserstein GAN [22] could be a possible solution.

For the scope of this work, synthetic data is being generated using Synthetic Data Vault (SDV) [1]. SDV is a set of libraries that can learn the statistical properties of single-table, relational table, and time-series data and then generate synthetic data which will have the same statistical consistency as that of the real data. Under the hood, SDV uses several probabilistic graphical modelling and deep learning techniques. SDV has different algorithms to generate synthetic data such as Tabular Preset, GaussianCoupla [13], Conditional-Tabular Generative Adversarial Network (CTGAN) [12], CouplaGAN, and a form of Variational Auto-Encoder (TVAE) models.

In Generative Adversarial Network (GAN) [11] [14] 6 ($eq1$) there are two 'adversarial' networks: a generator network G that tries to learn the distribution P of the data X, and a discriminator network D which predicts whether the input data point is real or fake. To learn the distribution of the data, the generator G creates a mapping function from noise distribution $P_z(Z)$ to data space. They both are trained simultaneously, parameters of G are adjusted to minimize $log(1 - D(G(z)))$ and parameters of D are adjusted to minimize $logD(X)$.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim Pdata(x)}[\log D(x)] + \mathbb{E}_{x\sim Pz(z)}[\log(1 - D(G(z)))] - (1)$$

For the purpose of this work, CTGAN [12] and Automated generative modelling have been used. CTGAN 6 ($eq2$) is a variation of GAN where the generator and discriminator networks are conditioned on c: c can be data from other models or a class label. C is then combined with the input and passed on to both generator and discriminator networks.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim Pdata(x)}[\log D(x|c)] + \mathbb{E}_{x\sim Pz(z)}[\log(1 - D(G(z|c)))] - (2)$$

4.5.2 Random Forest Regression Model

To evaluate the consistency of the synthetic data, random forest regression model [9] was trained to predict the time spent by people on a particular activity. A total of 3 different regression models were trained, 2 models were trained on different configurations 4.4 of the synthetic data and 1 model on real data using *spss* encoding technique. Out of all the features from both activity log and individual info table 20 features 4.4 have been selected to predict time spent (eptime) mainly because most of the features have null values and also to reduce the complexity of the model. Data points are normalised to make the scale of all the features uniform.

Random forest [9] was used because the data is non-linear (discussed in section 4.4). Random forest selects a set of data points, builds a decision tree, and repeats this process until the training is done. Because of its ensemble nature, random forest handles bias-variance trade-offs reasonably well.

5 Results

To check the statistical consistency of the synthetic data following tests have been performed:

5.1 Heatmap of Correlation of data points

4 heat maps are plotted on different configurations 4.4 of the synthetic data and they are compared with the heat map of real data.

Figure 12 shows the correlation of the real and synthetic data both encoded using *spss* encoder. From the figure, it can be inferred that most of the major variations in the data have been picked by the model, however, it is missing out on smaller details. Inverted Kolmogorov-Smirnov D statistic Test (KSTest) [16] was also performed and it achieved a score of 0.769.

Figure 13 is the comparison of the heat map of correlation of all the data points of real data and synthetic data, both are *spss* encoded. Activity log and individual info tables were merged first and then synthetic data was generated. In this figure it can be seen that along with major variations most of the smaller details have also been picked properly. The KSTest score of this model is 0.785.

In figure 14 real and synthetic data have been encoded using a count encoder and both activity log and individual info tables were generated separately. From this figure it can be observed that both the correlation plots are very different and they don't share many statistical properties and a low KSTest score of 0.55 also supports the aforementioned statement.

The last heat map 15 shows the correlation of real and synthetic data encoded using a count encoder. Activity log and individual info tables were merged first and then synthetic data was generated. Here the model has done a bit better as compared to the previous model 14 and this model achieved the KSTest score of 0.684.

All the further tests will be performed on *spss* encoded real data, and synthetic data that was generated after merging the activity log and individual info table because it achieved the maximum KSTest score (0.785) among all the other configurations of the data and in the figure 7 it can be deduced that the model is not simply generating the same data points because there is actually some difference between real and synthetic data correlation heat map.

5.2 Mean and standard deviation of real and synthetic data

Figure 8 shows the plot of the log of mean and standard deviation of real data vs synthetic data. From the figure, it can be observed that both mean and standard deviation plots are in straight line suggesting that both real and synthetic data shares similar statistical properties.

5.3 Cumulative Distribution Curve

Figure 16 shows the cumulative distribution [18] of all the features of real (Blue) and synthetic (Orange) data. From the figure, it can be deduced that most of the features are following similar distribution on both real

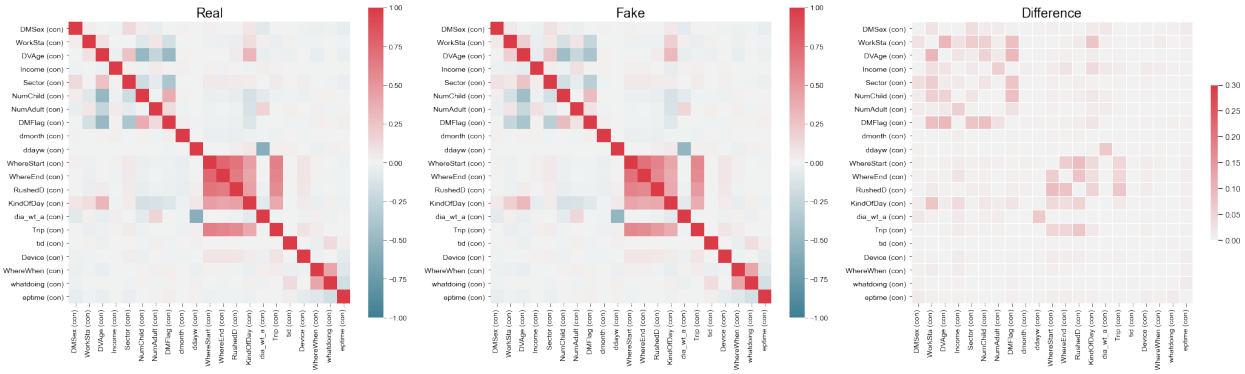


Figure 7: Heatmap of Correlation Values (Merged Data 1st then Generated) and Difference

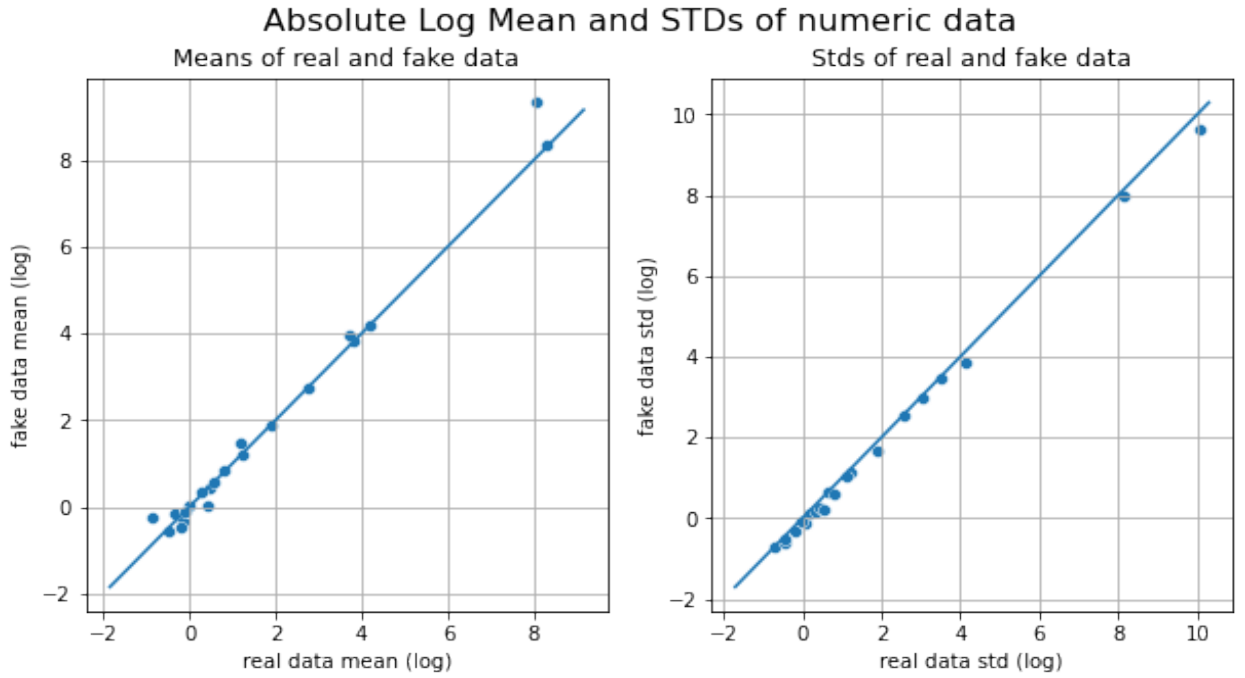


Figure 8: Log of Mean and Std

and synthetic data, however, there are few categorical features such as *WhereStart*, *WhereEnd*, *RushedD*, *Device*, and *Trip* are deviating significantly.

5.4 Probability Distribution Curve

Similar to cumulative distribution, the probability distribution [17] of most of the features of synthetic data are somewhat close to that of the real data with some features deviating significantly which can be seen in the figure 17.

It can also be observed that the model has generalised the probability distribution [17]. When zoomed on to the probability distribution of *Age* and *Gender* in figure 9, it can be inferred that the distribution of *Age* in real data is not exactly normally distributed but it can be considered as a form of normal distribution whereas the distribution of *Age* in synthetic data is almost normally distributed. As for *Gender*, the distribution in both the data is very closely matched.

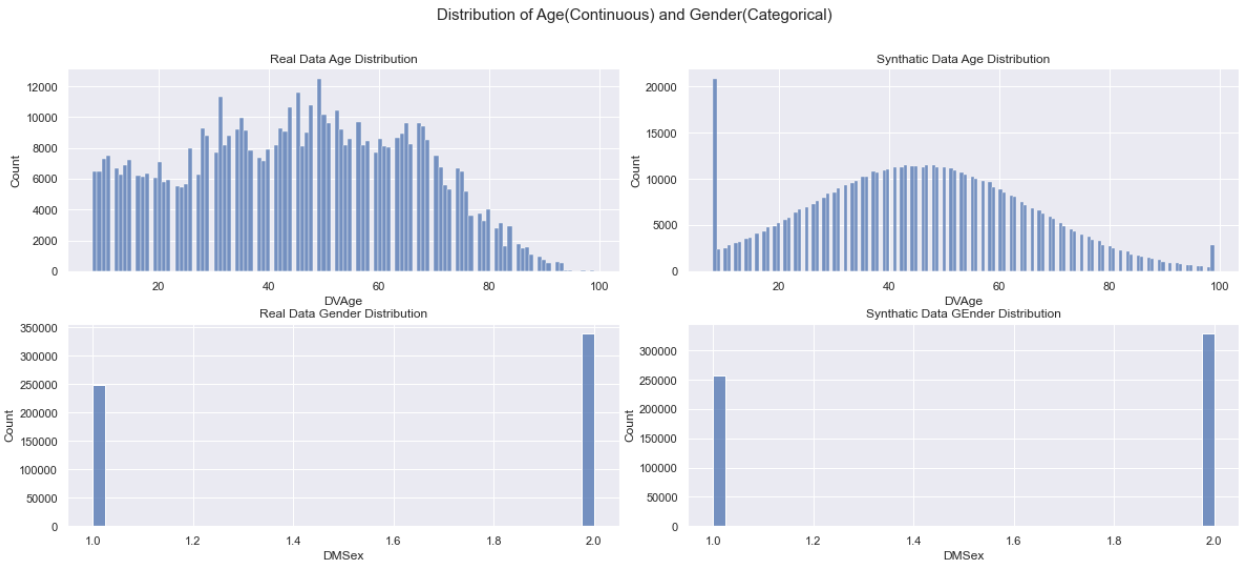


Figure 9: Probability distribution of Age and Gender

5.5 Regression Model Accuracy

Random Forest [9] regression models were trained on real and synthetic data encoded in *spss* format in various data configurations 4.4. Table 1 shows the results of the models and it can be seen models are performing similarly. All the 3 models have mean squared error close to 2200, root mean square error close to 47. The mean absolute errors of both the models trained on synthetic data are close to 39 but they are relatively more than that of the model trained on real data which is still acceptable as the difference is not too much.

Table 1: Regression model Performance on Different Data Config.

Data Config.	Mean Sq. Error	Root Mean Sq. Error	Mean Abs. Error
Real Data, SPSS Encoded	2176.068	46.648	25.775
Synthetic Data, SPSS Encoded, Generated Separately	2298.557	47.943	39.412
Real Data, SPSS Encoded, Merged then Generated	2189.053	46.787	38.622

6 Conclusion

To summarise, synthetic data generated using Synthetic Data Vault (SDV) [1] is reliable enough to perform real-world analysis and also to train machine learning models. This work demonstrated the real-world use case of synthetic tabular data generation by using a complex behavioural dataset which was highly randomised and yet SDV did reasonably well to pick the different distributions of each feature. There were some categorical features where distribution was off by quite some margin but overall it was acceptable.

Also, if the dataset contains multiple tables and they are linked in some way but not exactly relational database tables, it has been observed that it is better to first merge the tables in the required format and then generate synthetic data because from the tests performed in this work, data that was first merged in the required format and then synthetic data was generated, it produced relatively better results.

7 Future Scope

- Although the data generated in this work was reliable enough and can be used to carry out good analysis or even train machine learning models there is still room for improvement such as some of the feature distributions were not matching with the real data which can be improved.
- SDV [1] provides various different models to generate synthetic data as discussed in section 4.5.1. For the scope of this work, GAN [11] [14] based models have been used. Other models can also be tested and see how they stack up against GAN-based models.
- In this work, encodings like *spss* and count encoder have been used, however, there are many other categorical encoding techniques in the market, that can also be tried.
- As discussed earlier in the section 4.5.1, if developing a synthetic data generator from scratch Wasserstein GAN [22] can be tried in the future to see if its able to learn the data type of the columns.

References

[1] GitHub. 2020. GitHub - sdv-dev/SDV: Synthetic Data Generation for tabular, relational and time-series data.. [online] Available at: <https://github.com/sdv-dev/SDV> [Accessed 20 May 2022].

[2] Ashrapov, I., 2020. Tabular GANs for uneven distribution. arXiv preprint arXiv:2010.00638.

[3] Xu, L. and Veeramachaneni, K., 2018. Synthesizing tabular data using generative adversarial networks. arXiv preprint arXiv:1811.11264.

[4] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G. and Salimans, T., 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv preprint arXiv:2205.11487.

[5] Markman, A., 2008. Unpredictability is in our nature.. [online] Psychology Today. Available at: <https://www.psychologytoday.com/us/blog/ulterior-motives/200811/unpredictability-is-in-our-nature> [Accessed 29 July 2022].

[6] Morris, S., Humphrey, A., Cabrera Alvarez, P. and D'Lima, O., 2016. The UK Time Diary Study 2014 - 2015. [online] Doc.ukdataservice.ac.uk. Available at: <http://doc.ukdataservice.ac.uk/doc/8128/mrdoc/pdf/8128_natcen_reports.pdf> [Accessed 29 April 2022].

[7] Gershuny, J. and Sullivan, O., 2019. What we really do all day: Insights from the Centre for Time Use Research. Penguin UK.

[8] Beta.ukdataservice.ac.uk. 2017. UK Data Service › Study. [online] Available at: <https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=8128> [Accessed 22 April 2022]

[9] contributors, W., 2022. Random forest - Wikipedia. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/w/index.php?title=Random_forest&oldid=1102453131> [Accessed 22 July 2022].

[10] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M. and Sutskever, I., 2021, July. Zero-shot text-to-image generation. In International Conference on Machine Learning (pp. 8821-8831). PMLR.

[11] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

[12] Mino, A. and Spanakis, G., 2018, December. Logan: Generating logos with a generative adversarial neural network conditioned on color. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 965-970). IEEE.

[13] Pitt, M., Chan, D. and Kohn, R., 2006. Efficient Bayesian inference for Gaussian copula regression models. Biometrika, 93(3), pp.537-554.

[14] contributors, W., 2022. Generative adversarial network - Wikipedia. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/w/index.php?title=Generative_adversarial_network&oldid=1102465217> [Accessed 3 June 2022].

[15] Zhu, J.Y., Park, T., Isola, P. and Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[16] Durbin, J., 1975. Kolmogorov-Smirnov tests when parameters are estimated with applications to tests of exponentiality and tests on spacings. Biometrika, 62(1), pp.5-22.

[17] contributors, W., 2022. Probability distribution - Wikipedia. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/w/index.php?title=Probability_distribution&oldid=1102516344> [Accessed 30 July 2022].

[18] contributors, W., 2022. Cumulative distribution function - Wikipedia. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/w/index.php?title=Cumulative_distribution_function&oldid=1104350887> [Accessed 30 July 2022].

[19] Dilmegani, C., 2020. Synthetic Data Generation: Techniques, Best Practices & Tools. [online] AIMultiple. Available at: <https://research.aimultiple.com/synthetic-data-generation/> [Accessed 21 April 2022].

[20] Aindo.com. 2022. Aindo AI - Your AI partner. [online] Available at: <https://www.aindo.com/> [Accessed 24 April 2022].

[21] contributors, W., 2022. Cross-sectional data - Wikipedia. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/w/index.php?title=Cross-sectional_data&oldid=1082685505> [Accessed 25 April 2022].

[22] Arjovsky, M., Chintala, S. and Bottou, L., 2017, July. Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[23] 2019. Harmonised European Time Use Surveys (HETUS). 2nd ed. Luxembourg: Publications Office in Luxembourg.

[24] Ukri.org. n.d. Economic and Social Research Council (ESRC). [online] Available at: <https://www.ukri.org/councils/esrc/> [Accessed 9 June 2022].

Appendices

This section contains all relevant plots which were developed to analyse the data and test the synthetic data for its statistical consistency and close representation of the real data.

Code base can be found at this link

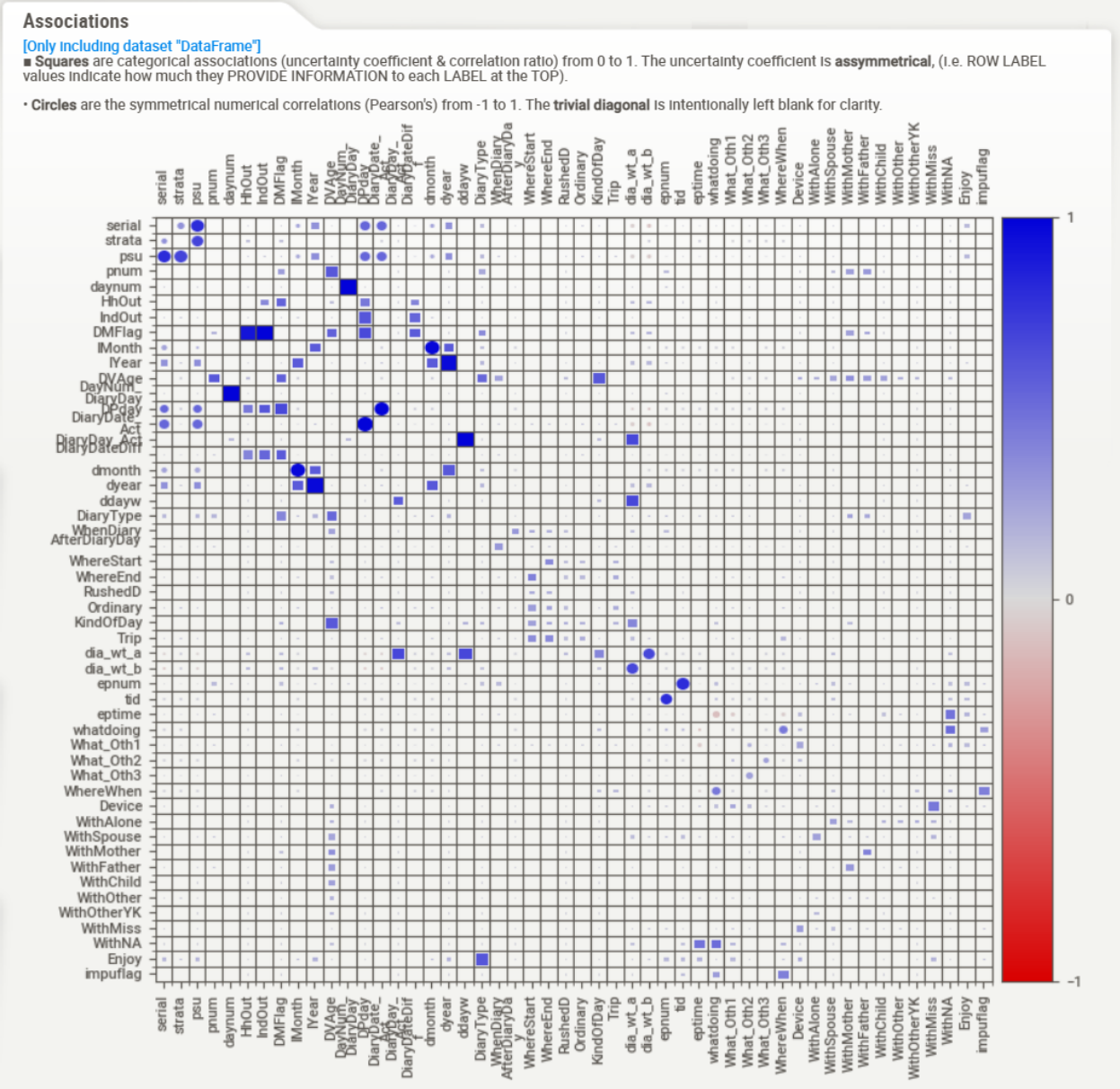


Figure 10: Activity Log Pairplot

Appendix A: Pairplot of Activity log table

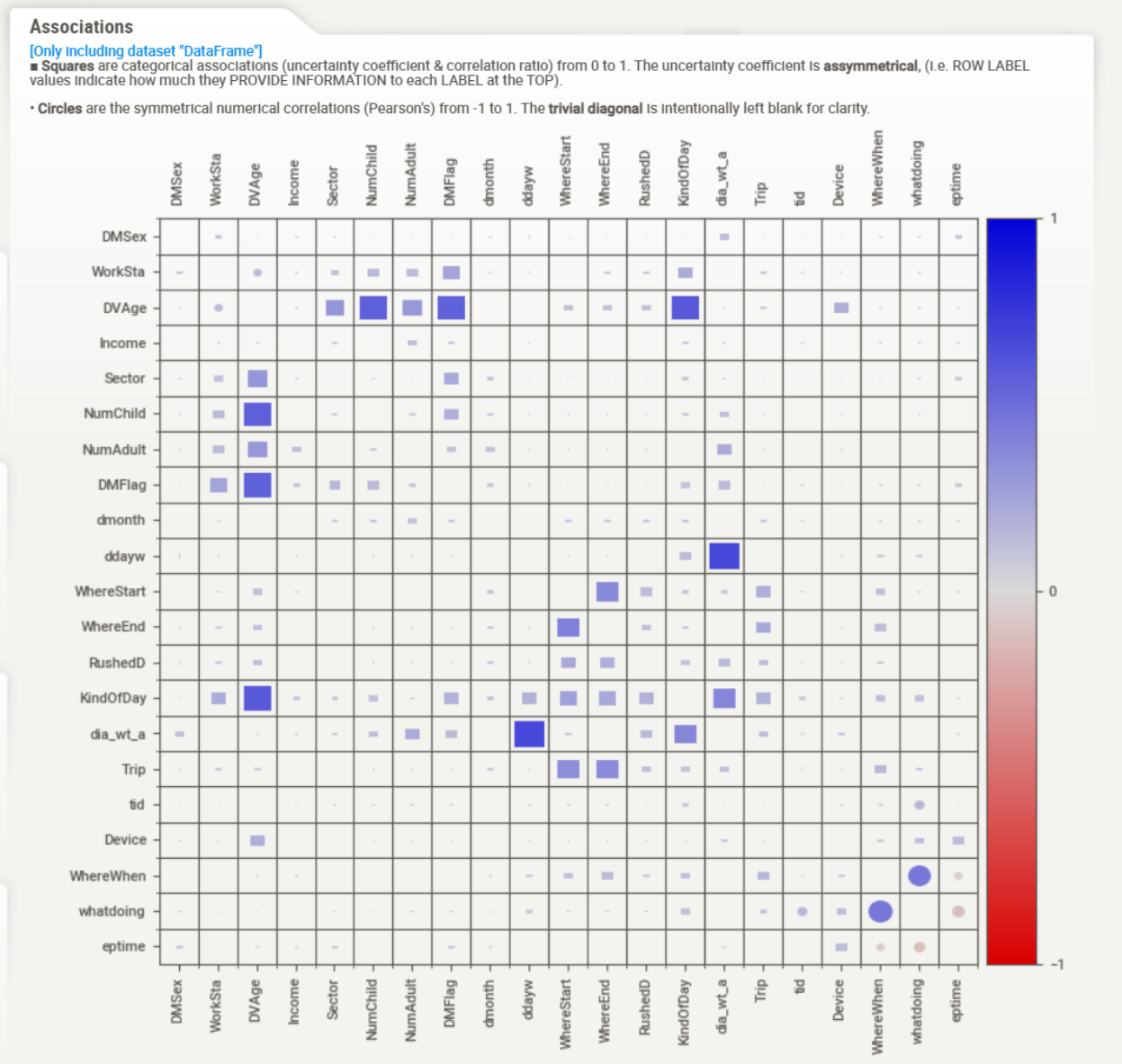


Figure 11: Merged table Pairplot

Appendix B: Pairplot of the dataset after merging them in required format

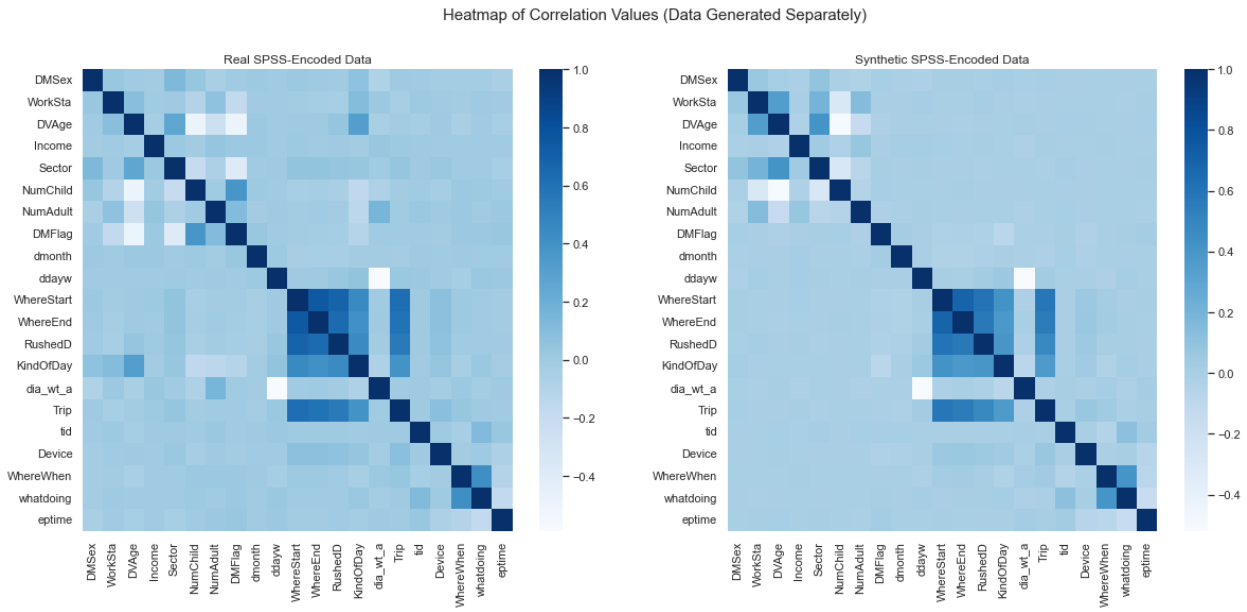


Figure 12: Heatmap of Correlation Values (Data Generated Separately)

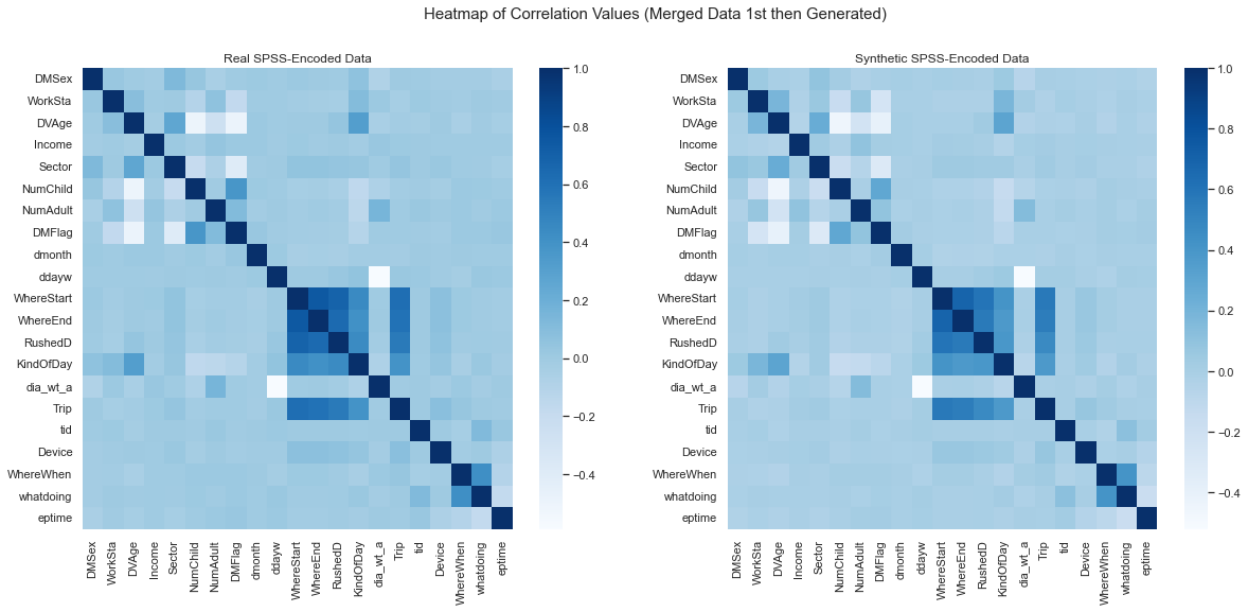


Figure 13: Heatmap of Correlation Values (Merged Data 1st then Generated)

Appendix C: Heatmap of Correlation Values (Data Generated Separately)

Appendix D: Heatmap of Correlation Values (Merged Data 1st then Generated)

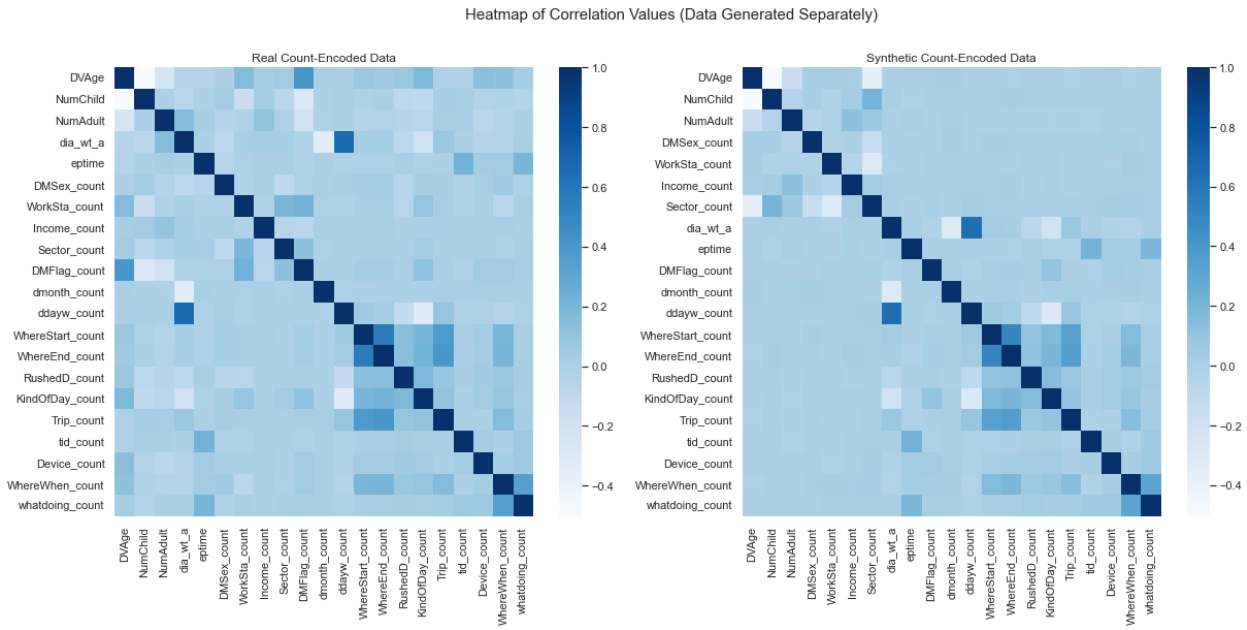


Figure 14: Heatmap of Correlation Values (Data Generated Separately)

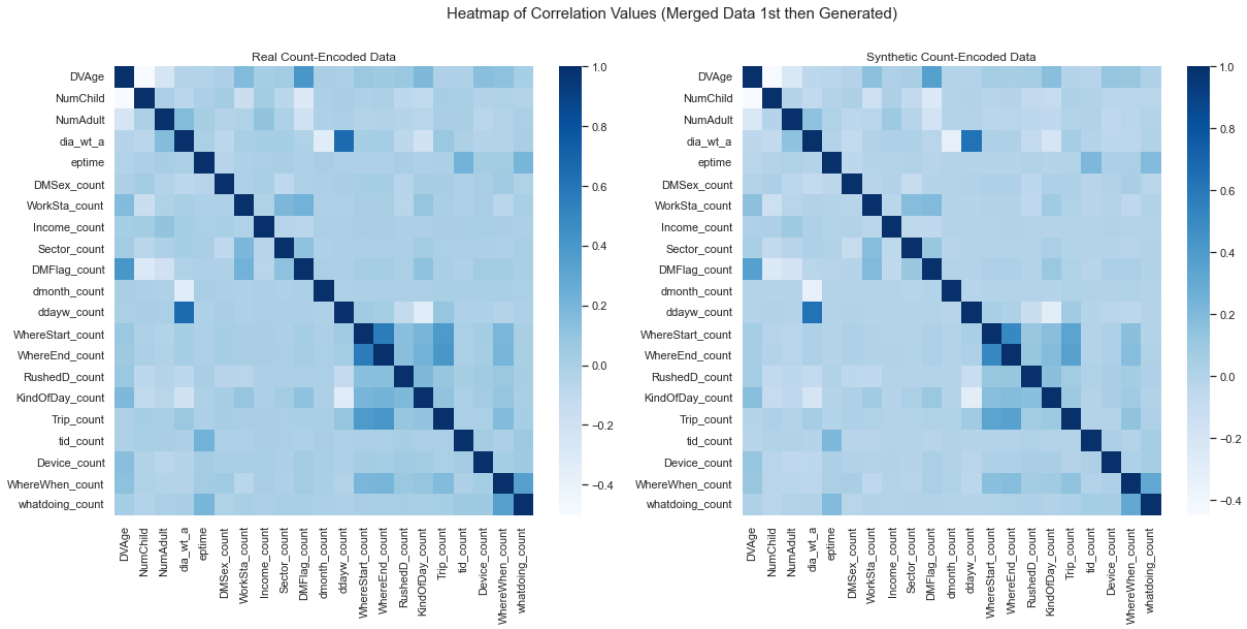


Figure 15: Heatmap of Correlation Values (Merged Data 1st then Generated)

Appendix E: Heatmap of Correlation Values (Data Generated Separately)

Appendix F: Heatmap of Correlation Values (Merged Data 1st then Generated)

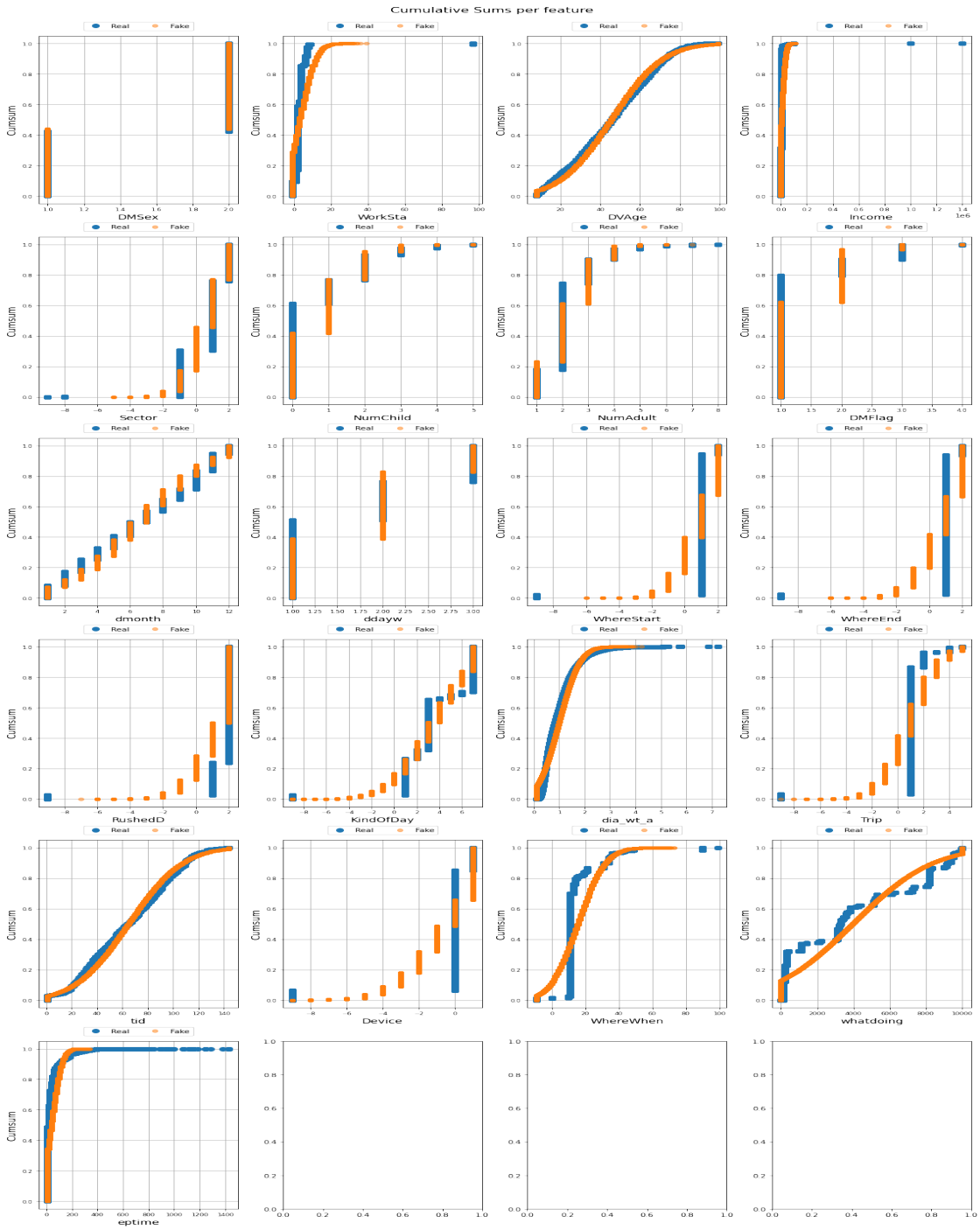


Figure 16: Cumulative Distribution Per Feature

Appendix G: Cumulative Distribution of all the features

Figure 17: Probability Distribution Per Feature

Appendix H: Probability Distribution of all the features