Volume 7, Issue 4, April – 2022 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165

An Enhancement of Deep Feature Synthesis Algorithm Using Mean, Median, and Mode Imputation

Josefa Ysabelle J. Maliwat, Princess A. Ylade,
Richard C. Regala, Dan Michael A. Cortez, Antolin J. Alipio, Khatalyn E. Mata, Mark Christopher R. Blanco,
Computer Science Department Pamantasan ng Lungsod ng Maynila Manila, NCR, Philippines

Abstract:- The Deep Feature Synthesis (DFS) algorithm automates feature engineering and is capable of extracting and applying complicated features to a variety of processes. Due to the novelty of DFS as a method for feature engineering, critical ways for dealing with missing values and unwanted data in a dataset have yet to be established. This paper discusses the usage of mean, median, and mode imputation to preprocess data before analyzing it. However, it is only limited to displaying the differences between non-imputed and imputed datasets. This strategy enables users to obtain more precise results by eliminating biased estimations. This study demonstrates that there is a distinct difference between the two datasets. This paper is concluded by proving that imputing datasets will cause distinctness in the results compared to the results of the datasets with missing and unwanted values.

Keywords:- *Deep Feature Synthesis, Auto Feature Engineering, Imputation.*

I. INTRODUCTION

Automated feature engineering intends to aid data scientists through the automatic creation of features from a dataset in which, the best out of all the features, can be chosen and can be used for training [1]. Creating features, especially through the use of data from different tables, can be a tedious process for the reason that each new feature typically requires a number of steps to construct.

The Deep Feature Synthesis (DFS) is an algorithm that automatically allows users to generate features for relational datasets. Relational data is currently the most commonly used type of enterprise data, which is why this type of data is being focused on. DFS is auspicious when aggregating features in a relational database structure [2]. The algorithm analyzes and follows the relationships of the different data in a base field. Then, it applies mathematical functions over the features to create the final output, a base table that portrays a summary of the relational data.

By applying mathematical functions and calculations, the final features of the datasets can be defined. Its process has a set of related entities and tables. Although the natural language descriptions are entirely different, the fundamental math remains the same. The same operation on a list of numeric values to generate a new numeric feature unique to the dataset is used. These could be repeatedly stacked to establish complex features. Features can be created with many tables because DFS applies primitives across different relationships between datasets. Several mathematical methods create new features, including getting the mean, median, or mode. However, DFS still does not have procedures to treat the null and unwanted values from each dataset.

The main focus of this study is to handle missing values by using mean, median, and mode imputation. The researchers recognize that some situations do not require imputation, so imputing is made optional for the users. After utilizing the imputation technique, this paper only shows the differences between the non-imputed and imputed datasets.

II. REVIEW OF RELATED LITERATURE

The automation of feature engineering opens the door to expedite the process of applying machine learning to the datasets that are collected by data science teams nowadays [3]. Data scientists can now address new problems swiftly as they arise. Additionally, automated feature engineering will make it easier for new learners of data science to develop the necessary skills needed and apply it to their own field of expertise.

Feature extraction is an essential method in computer vision and it is used for object recognition, image alignment, navigation for robots, etc [4]. Convolutional Neural Network (CNN) is one of the popular deep learning techniques that deal with knowledge representation. It divides the images into layers so that each layer is more carefully studied compared to traditional analysis procedures. CNN's strong capacity in extracting complex features makes it easier to learn task specific features making it more efficient.

While deep learning automates feature engineering for text, audio, and images wherein a huge training set is required, the target of deep feature synthesis is to automatically engineer features for structured transactional and relational datasets. DFS can already start producing features based only on the schema of a dataset. On the other hand, many training examples are vital in order to train the complex architectures of deep learning for it to work.

ISSN No:-2456-2165

III. PROPOSED METHOD

To achieve the objective of having the option to preprocess the data before extracting its features, this paper proposes the use of imputation, particularly the mean, median, and mode imputation methods.

A. Adding the mean, median, and mode imputation into the DFS Algorithm

Mean and median imputation works by filling in the missing values by the mean or median for quantitative attributes in a dataset. Mode imputation uses the most common value to replace the missing values for qualitative attributes. After loading the datasets and containing them in their respective variables, a temporary list holding the datasets to be imputed is made. This is done to loop through the datasets and pass them into the imputation function. The function will return the imputed datasets and pass them back to the temporary list to overwrite non-imputed datasets. After this process, the datasets are now ready for feature extraction.

B. Imputing the dataset using the mean, median, and mode

The imputation Module function will receive the data frame and the choice of imputation method to be used for numerical data types. The data frame to be imputed will enter the function. Then, the column Name and column Data will loop through the contents. After identifying the column's data type, the missing values will be imputed according to the choice of imputation method passed in the parameter. For string data types, mode imputation will automatically be used.

C. Pseudocode

function IMPUTATION(dataframe, method):
for columnName, columnData in dataframe.iteritems():
if dataframe[columnName].dtype == ‘object’:
dataframe[columnName] =
dataframe[columnName].fillna(dataframe[columnName].mode()[0])
elif dataframe[columnName].dtype == ‘float64’ or ‘int64’:
if method == ‘mean’:
dataframe[columnName]
=dataframe[columnName].fillna(dataframe[columnName].mean())
elif method == ‘median’:
dataframe[columnName]
=dataframe[columnName].fillna(dataframe[columnName].median())
elif method == ‘mode’:
dataframe[columnName]
=dataframe[columnName].fillna(dataframe[columnName].mode())
return dataframe

IV. RESULTS AND DISCUSSION

Fig. 1 shows the result of data aggregation with the use of the raw dataset. Fig. 2 shows the results of data aggregation with the use of the imputed dataset. The highlighted cells in Figures 1 and 2 are the values that have a difference between the results. As shown in Fig. 2, there is a difference between the values in the feature matrix of the non-imputed datasets and the imputed ones.

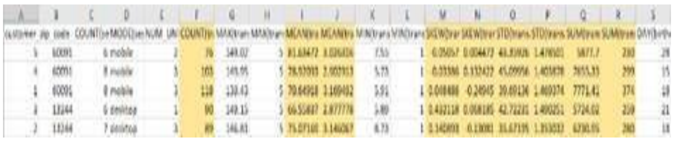


Fig. 1. (a) The image shows the first 19 columns of the feature matrix using the non-imputed datasets.

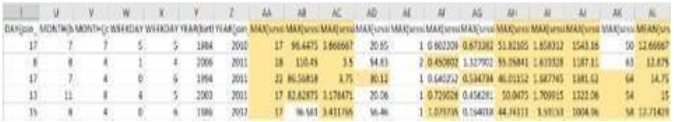


Fig. 1. (b) The image shows the second 19 columns of the feature matrix using the non-imputed datasets.

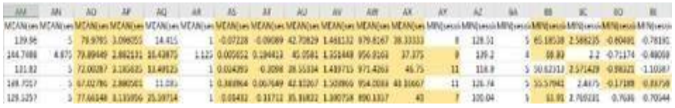


Fig. 1. (c) The image shows the third 19 columns of the feature matrix using the non-imputed datasets.

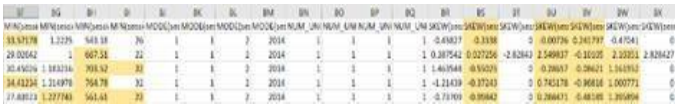


Fig. 1. (d) The image shows the fourth 19 columns of the feature matrix using the non-imputed datasets.

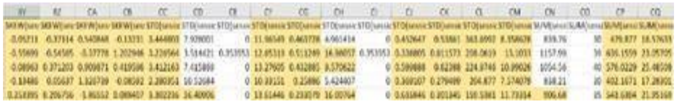
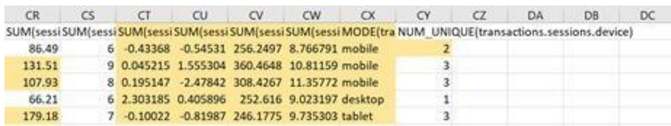


Fig. 1. (e) The image shows the fifth 19 columns of the feature matrix using the non-imputed datasets.



CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC
SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	MODE(tra	NUM_UNIQUE(transactions.sessions.device)				
86.49	6	-0.43368	-0.54531	256.2497	8.766791	mobile	2				
131.51	9	0.045215	1.555304	360.4648	10.81159	mobile	3				
107.93	8	0.195147	-2.47842	308.4267	11.35772	mobile	3				
66.21	6	2.303185	0.405896	252.616	9.023197	desktop	1				
179.18	7	-0.10022	-0.81987	246.1775	9.735303	tablet	3				

Fig. 1. (f) The image shows the last 8 columns of the feature matrix using the non-imputed datasets.

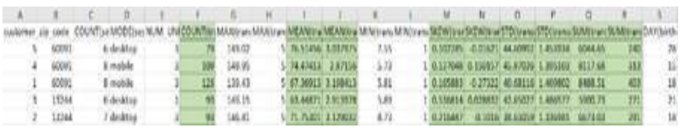
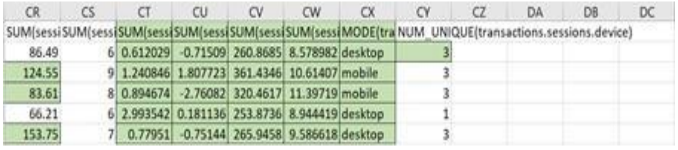


Fig. 2. (a) The image shows the first 19 columns of the feature matrix using the imputed datasets

ISSN No:-2456-2165

Fig. 2. (d) The image shows the fourth 19 columns of the feature matrix using the imputed datasets.

Fig. 2. (e) The image shows the fifth 19 columns of the feature matrix using the imputed datasets.



CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC
SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	SUM(sessi	MODE(tra	NUM_UNIQUE(transactions.sessions.device)				
86.49	6	0.612029	-0.71509	260.8685	8.578982	desktop	3				
124.55	9	1.240846	1.807723	361.4346	10.61407	mobile	3				
83.61	8	0.894674	-2.76082	320.4617	11.39719	mobile	3				
66.21	6	2.993542	0.181136	253.8736	8.944419	desktop	1				
153.75	7	0.77951	-0.75144	265.9458	9.586618	desktop	3				

Fig. 2. (f) The image shows the last 8 columns of the feature matrix using the imputed datasets.

V. CONCLUSION AND RECOMMENDATION

The researchers implemented the mean, median, and mode imputation into DFS to handle missing and unnecessary data. It showed that imputing the missing and unnecessary values in a dataset will establish that there is a distinction between using imputation and not. Moreover, the imputation outcome is more evident if large datasets are utilized. Because the researchers identify further gains in this study, they recommend these essential directions for future research. First, the imputation method was performed on a separate file. Future work could concentrate on incorporating the code into the *AggregationPrimitive* file of the *featuretools* package. The present work still has no imputation methods for boolean data type. Use statistical analysis techniques if future researchers want to check the accuracy of the datasets. Finally, the imputation methods for float and int data types should be separated.

ACKNOWLEDGMENT

The researchers are thankful to the Almighty God for providing enough will and strength to make this study possible, to the professors of Computer Science Department at Pamantasan ng Lungsod ng Maynila for their help and guidance, and to their family and friends for their love, support, and company throughout the days of making this research paper.

REFERENCES

[1]. Koerhsen, W. (2018, June 2). Automated Feature Engineering in Python. Retrieved from https://towardsdatascience.com/automated-feature-engineering-in-python-99baf11cc219

[2]. Kanter, J. M. and Veeramachaneni, K., "Deep feature synthesis: Towards automating data science endeavors," *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015, pp. 1-10, doi: 10.1109/DSAA.2015.7344858.

[3]. Kanter, M. (2018, January 16). Deep Feature Synthesis: How Automated Feature Engineering Works. Retrieved from https://www.linkedin.com/pulse/deep-feature-synthesis-how-engineering- automation-works-max-kanter

[4]. Patel, K. (2020, September 9). Image Feature Extraction: Traditional and Deep Learning Techniques. Retrieved from https://towardsdatascience.com/image-feature-extraction-traditional-and-deep- learning-techniques-ccc059195d04