2021 International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 25-27, 2021

Insurance Data Analysis with COGNITO: An Auto Analysing and Storytelling Python Library

Anshul Saxena
Management Studies
Christ University
Bangalore, India
anshul.saxena@christuinversity.in

Dr Vandana Vijay Bhagat
Data Science
Christ University
Bangalore, India
vandana.bhagat@christuniversity.in

Bastin Robins
Data Science
Cleverinsight Inc.
Bangalore, India
robin@cleverinsight.co

2021 International Conference on Intelligent Technologies (CONIT) | 978-1-7281-8583-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CONIT51480.2021.9498523

Abstract: **Data pre-processing has taken an enhanced role with the advent of Machine learning. It is a vital element that forms the encore of the data science and business analytics process. Data pre-processing involves generating descriptive statistical summary, data cleaning, and data manipulation based on inputs gained after the initial analysis. Of late, it has been observed that data science practitioners spend 45% to 50% of their time cleaning and processing the data. Much time can be saved if the data transformation process can be automated. The COGNITO framework helps in performing the automated feature engineering and data storytelling of the dataset based on end-user discretion. The present work discusses the process and results obtained when automated feature engineering was performed on an insurance dataset using COGNITO.**

Key words: Feature engineering, automation, pre-processing, storytelling, NLP, encoding, auto-analysis, problem solving

I. INTRODUCTION

Clean and reliable data is instrumental in knowledge discovery via data analytics. Noisy and redundant information hampers the training process of the predictive model. Under the KDD process, data preparation forms a significant chunk of time taken for model preparation. Data pre-processing includes data cleaning, normalization, transformation, feature extraction, and selection.

Before building the machine learning (ML) model for further classification and forecasting, it is imperative to process the data to remove potential identifiers, missing values, and outliers. Data preprocessing significantly impacts the performance of the ML algorithm. Removal of noise from the dataset results in better reliability of the outcomes obtained from the model. Handling outliers, sampling, and missing values are quite a significant task that consumes much time.One more preliminary step which is performed manually is to obtain the statistical summary of data and look for the insights which can be obtained directly from simple descriptive analysis

COGNITO, a data pre-processing python library cleans the data automatically and provides a compressed data file. COGNITO helps data analyst's community in overcoming the above stated problem on two level:It performs an automated feature engineering diagnostic check looking for potential chances of level reduction, filling in missing values and highlighting the potential outliers. On second level it act as an voice based agent which answers the general queries like highest and lowest values present among variables that are the related to the datasets. The current research paper considers insurance data to show the usage of COGNITO framework for automated data preprocessing and basic statistical analysis.

This paper is divided into four sections. Section 1: Gives the essential introduction to the problem discussed. Section 2: Chronicles the recent attempt made in data pre-processing. Section 3: Introduces COGNITO and describes its features. Section 4: It shows the results obtained after processing the insurance dataset using the COGNITO library.

II. RELATED WORK

Data pre-processing is a complicated but equally crucial task in the data analysis. Data pre-processing involves various steps to be followed depending upon the quality and content of the data. Common methods used to clean the data are missing values imputation, Noise treatment, Dimensionality reduction, Instance reduction and many more[1]. It is also observed and proved that robust data pre-processing steps can improve the performance of the Supervised Learning algorithms like logistic regression, decision tree, XG Boost, clustering and linear regression model in terms of accuracy [2].

TABLE I. SURVEY OF FEATURES FORM EXISTING PRE-PROCESSING PYTHON LIBRARY

Package	Author	Contribution
PyHealth	Zhao et al.(2021)[3]	The data preprocessing module enables the transformation of complex healthcare datasets (medical images and clinical notes into machine learning friendly formats. into machine learning friendly formats.
MuSA	Zanfardino et al.(2021)[4]	Pre-processing section allows data filtering and normalization
dame-flame	Gupta et al.(2021)[5]	Use propensity score matching technique for data reduction
4SpecID	Neto et al.(2021)[6]	It performs raw database grading and semiautomated data curation for the datasets.
Sparx	Bhagat et al.(2019)[7]	It is data-preprocessing library, which involves transforming raw data into a machine-understandable format.
tableone	Pollard et al.(2018)[8]	It provides summary statistics in fixed format for research paper
bandicoot	Montjoye et al.(2016)[9]	Feature extraction library which fetch data from mobile handsets.
Pandas	Wes McKinney(2015)[10]	Python library used in providing tabular structure and data manipulation features.

From the literature review it was observed that data pre-processing is a crucial phase in data analysis. The libraries mentioned in Table 1 provides the data pre-processing ability in silos. There is a need for a stop gap solution for this kind of pre-processing solution.

III. COGNITO DATA PRE-PROCESSING LIBRARY

COGNITO is automated suite of various functionalities like converting uncleaned data into cleaned data, reduce the dataset size for faster execution, removal of unnecessary features from the dataset to avoid overfitting, generate a summary report of the dataset, provide basic questions and their answer from the dataset and offer the platform-independent facility to achieve data pre-processing [11]. It is always good to visualize the summary of the dataset. Various existing tools give a summary of the dataset without providing in-depth information about each feature of the data. COGNITO provides a summary of the CSV file as well as a description of each column. It can suggest the importance of each column in analysis and the anomalies which need to be fixed according to the standards for better model prediction.

COGNITO being a combined suite provides various features like,

- Command-line environment: Being a command-line environment, COGNITO is platform-independent. It can be executed with any programming language on any operating system. This makes the COGNITO more compatible for all types of users.
- Summarised report of dataset structure: In addition to providing textual summary, COGNITO also provides the visualized summary of the dataset in a more precise format after the primary data analysis.
- Possible questions and answers on the summarized dataset:COGNITO performs fundamental analysis and provides a list of essential questions with answers to make the end-user familiar with the dataset before further processing.
- Voice recognition and Audio storyteller: This is an exciting feature provided by COGNITO. It recognizes the voice command. An end-user can ask any fundamental question related to the dataset. COGNITO finds the relevant answers and provides the list of it. It also helps to tell a story about fundamental data analysis.

COGNITO Architecture: In COGNITO, various tasks are segregated udert three heads. Various functions are written under three different classes, as shown in Figure 1.

1. Check class determines the variable type and its properties. It identifies column variables as categorical, continuous, Identifier, and discrete and helps in data manipulation and dimensionality reduction based on end-user discretion.

2. Transfer class lists the methods involved in the data transformation process. It performs the following process: encoding of the categorical variable, treatment of missing values by removing columns based on the percentage method or replacing them with mean, median, and mode. Similarly, the Identifier (ID Column) detected in the check module will be automatically removed in the Transfer module.

3. Table class catalogs the methods responsible for creating an intermediate data frame during pre-processing. It helps in data analysis through binning, slicing, and scaling the data. Data cleaning and feature engineering functions like outlier identification, missing value treatment, and encoding of the column will be done under the table module

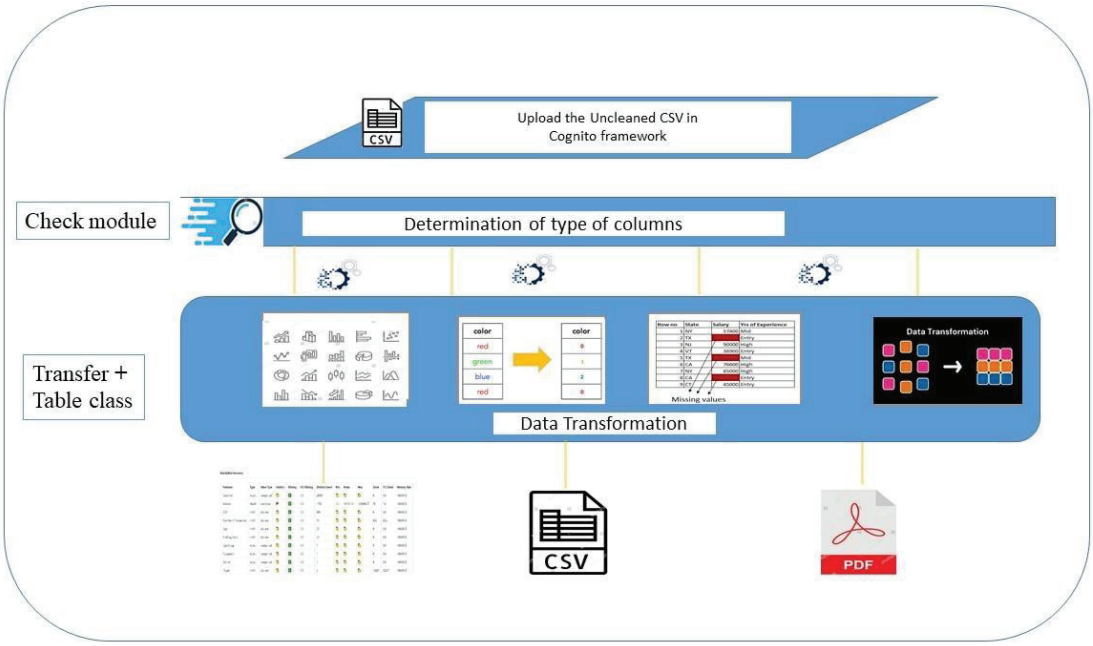

Fig. 1. COGNITO Base Architecture

Following steps needs to be followed to analyse the data using COGNITO

1. Install COGNITO using the command ‘pip install COGNITO’
2. Pass the uncleaned CSV file to COGNITO library using the command ‘COGNITO prepare -m -ml -imp filename.csv -out outpit_filename’
3. The uncleaned data gets processed through 3 modules, check module, transfer module and table module. Check module checks the type of columns of the dataset. Transfer module converts the datasets in usable format by handling missing values, outlier, categorical column encoding, removing identifiers and many more. Table module performs basic data analysis of cleaned data.
4. The COGNITO gives a CSV file with a cleaned dataset along with a PDF file with basic summary of the data using the command ‘COGNITO audit -i newly_created_filename.csv -0 report.
5. It also returns an HTML file which includes basic possible questions from the dataset which are auto answered by COGNITO using audio command.

IV. INSURANCE DATASET PREPROCESSING

This section of the paper included the description of data used for the analysis and the problem statement related to the data..Data used for analysis in the depicted case has been taken from the Kaggle website. This dataset contains ten columns and 20228 rows. A concerned dataset has been used for the risk profile analysis of customers of insurance companies. It captures the customers' demographic details like Cust id, Age, and the number of insurance claims. The dataset contains numerical and categorical variables. The given dataset is likely to help insurance companies to profile their existing customers into potential high-risk and low-risk categories. The same dataset will be used in forecasting the potential losses for the insurance companies.

A. Problem Description:

Data preprocessing aims to reduce the data size,find the relations between data, normalize data, remove outliers and extract features for data.It requires several techniques like data cleaning, integration, transformation and reduction.. Upon uploading the caselet data in the COGNITO framework, descriptive report gets generated which automatically flaggs the instances of missing data and outlier present in the given dataset. Figure 3 gives the overview of data summary generated by the COGNITO.

TABLE II. UNCLEAN VARIABLES OF INSURANCE DATASET

Variable Name	Problems
Target	Level reduction and One hot encoding
Customer ID	Duplicate records and an identifier variable
Age	Missing Values
Score	Outlier Values

Figure 2 includes the summary of the Insurance data before pre-processing. Table II lists four variables identified as problem areas to be treated using COGNITO as a part of data pre-processing. Applying appropriate pre-processing solutions on corresponding columns reduces the inefficiency of these variables in the prediction mode

COGNITO helps in providing the automated solution to the above problem. Upon looking at Fig.2, it has been observed that COGNITO provides a descriptive summary of data. It automatically shows the value types, object type and provides the statistical summary of the data. Given below are the instances where COGNITO has provided the quick and effective solution for the given case study.

Fig. 2. Ssummary data analysis given by COGNITO Framework

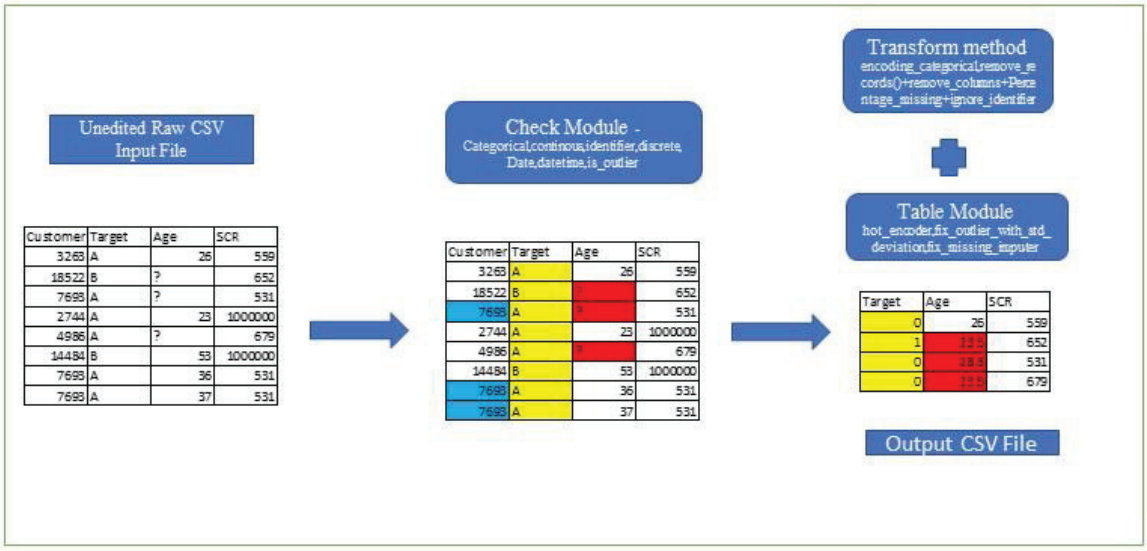

Fig. 3. Underlying Process running in background of COGNITO Framework

B. *Automation Process:*

The overall description of the process performed by COGNITO along with the type of problem getting solved is also described in this section. Upon processing the raw file, COGNITO framework through *check* module has detected the missing values and id variable (Customer ID) in the initial screening. Afterwards, through *transform* and table module one hot encoding and detection and removal of duplicate records and ID variables take place. Fig 3. summarizes the underlying process running in the background of COGNITO framework. Below the details related to subprocesses carried out by COGNITO framework has been discussed.

1.Missing value: By definition, missing data is the absence of the observation value stored under a variable of interest. Although missing data is a commonly recurring problem in the data cleaning process, it can significantly impact the conclusion drawn from the data if it is not treated. As shown in Figure 3 and discussed in Table 3, COGNITO has automatically identified and replaced the 58 missing values for '*Age*' variables with the mean values.

2.Dimensionality reduction: It is the process of transforming data from high dimensional space to low dimensional space. It helps in retaining the meaningful properties of the original data close to its intrinsic dimension. In the dataset provided above,ID variable (*policy no.*) has been automatically removed from the data file which is prepped for the model building.

***3.Duplicate Records*:** Amidst the large dataset, maintaining necessary duplication and removing unnecessary duplicate records is a complex task. Through an automated method of duplication detection and removal, data can be processed in a lesser timeframe while maintaining a valuable relationship and features of a dataset. In the given example, entire 228 instances of duplication of the variable *Customer ID* were identified and successfully remove.

4. One hot encoding: While dealing with a binomial/polynomial classification problem, the multitude of machine learning algorithms is unable to operate on label data directly. All input and output attributes should be numeric in nature. This limitation hinders the efficient implementation of the algorithm. In the current instance which is running, COGNITO has replaced alphabetical labels A(Associated) and E(Ex) customers with 0 and 1 for *target* variable.

5.Outlier Detection: Outliers can mislead the statistical diagnostics due to skewness, leading to wrong interpretation of underlying data and relationships. Outlier treatment is a critical step in the data cleaning process. Its removal results in a better fit for data and more accurate predictions. In the present instance, outlier values(score value more than 1200) which were present in the variable *score* were removed automatically by the COGNITO.

6. Data Storytelling: One unique feature of COGNITO is its data storytelling capability. It is a query-based voice recognition tool that answers the initial queries that can solve researchers' basic question on voice commands. Figure 4 shows graphical representation of three auto generated queries along with the corresponding answers. COGNITO processes and answers the business-related queries in the natural language(English) of the end-user.

V. RESULT AND DISCUSSION

Upon passing the prepped file again through the COGNITO framework we can see the list of transformed variables. Missing values under *Age* variables has been removed and replaced with mean values. *Balance* variable values are normalized. Also, outliers are also removed from the *score* column. Alphabetical labels under target columns is replaced with numerical values 0 and 1. Similarly duplicate values under variable *policy number* has been removed. Thus, COGNITO library has helped the analyst in data cleaning, transformation, and reduction. So it can be concluded otherwise the process which might have taken 2 hours if data was cleaned using manual solution was processed by COGNITO framework in 0.162 seconds.

Table 2 shows the solution obtained by COGNITO over the problems mentioned in Table 1. There are 4 problems encountered and solved in the experimental dataset. From figure 4 it can be seen that identifiers are removed; missing values and outliers are identified and imputed and duplicated values are removed. Table 3 shows the comparison between manual process and automated process.

TABLE III. LIST OF SOLUTIONS OBTAINED AFTER DATA PRE-PROCESSING.

Variable Name	Problems	Solution Obtained
Target	One hot encoding	COGNITO has replaced alphabetical labels A(Associated) and B(Ex) customers with 0 and 1 for *target* variable.
Customer ID	Duplicate records and an identifier variable	In the dataset provided above, duplicate records and id variable presents in variable (Customer ID) has been automatically detected and removed from the data file.
Age	Missing Values	In the given example, entire 228 instances of duplicated were identified and successfully removed.
Score	Outlier Values and level reduction	Outlier values (score value more than 1200) which were present in the variable score were removed automatically by the COGNITO thus inducing the level reduction.

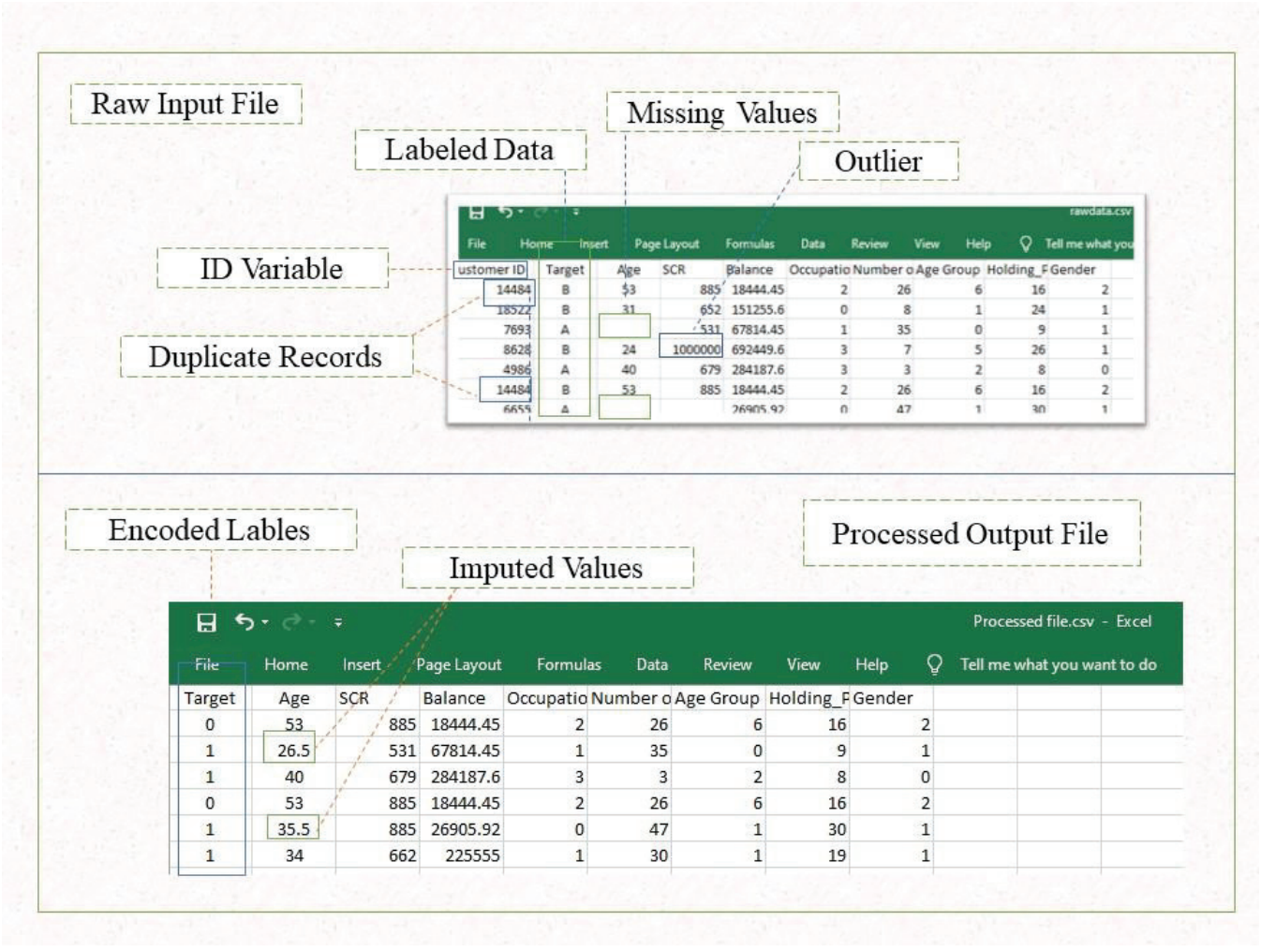

Fig. 4. State of raw input file and processed output file

TABLE IV. COMPARATIVE ANALYSIS OF MANUAL PROCESS AND AUTOMATED PROCESS.

Sr No	Anomalies	Manual processing to handle anomalies	COGNITO
1	One hot encoding	1. Finding number of classes in the given column (Associated (A) , Ex (B). 2. Setting specific encoded value for each class (A – 0, B – 1). 3. Replacing class with defined encoded value in the data column.	Automated
2	Identifier Removal	1. Finding the types of all the column in the dataset. 2. Counting total number of records in the dataset. 3. Finding number of unique values in each column. 4. Comparing records count and unique record counts and decide the identifier column. 5. Removing the identifier column from the dataset	Automated
3	Duplicate records Removal	1. Identifiers are considered to be used to check uniqueness of the record. 2. Finding duplicated entries of identifier 3. Removal of duplicated values from the dataset.	Automated
4	Missing Values Imputation	1. Identifying the columns containing missing values 2. Finding the percentage of missingness 3. If missing percentage is less than 10%, the records are be removed. 4. If the percentage of missing value is more than 10% , then impute it with mean / median / mode	Automated
5	Identify and remove outliers	1. Finding the columns containing outliers 2. Finding percentage of outliers. 3. Check the number of outliers present in the dataset. 4. If the count is less than 10 % then removal of the complete record.	Automated

VI. CONCLUSION AND FUTUREWORK

Data Processing is a vital technique to improve prediction models' performance. Its automation can reduce the efforts of data engineers. Processing of the raw data follows standard procedures and templates according to the problems that exist in the variables. It is possible to automate these standard procedures by checking the problem's existence in a corresponding variable. COGNITO, a Python auto analysis and storytelling library, has automated the pre-processing of raw data for performance improvement. The current paper has used Insurance data to show the capability of COGNITO to successfully preprocess the data and provided a fundamental analysis of it. This library has reduced the majority of data analysts' work and enabled them to concentrate more on prediction strategies.

The future scope of COGNITO is very vast, where the plethora of processing tasks related to machine learning algorithms will be automated, and a data pipeline can be generated to provide end to end solution for machine learning. Further work can be based on automated machine learning, the deep-learning process, and automated hyper parameters' optimization using the COGNITO library. Authors intend to branch out their work in automation of following

- Time Series Analysis
- Clustering
- Estimation
- Prediction
- Classification
- Descriptive Analysis

ACKNOWLEDGMENT

COGNITO is developed by the authors of the paper itself as CMD line library. It can be assessed here at https://pypi.org/project/cognito

REFERENCES

[1] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Anal.*, vol. 1, no. 1, pp. 1–22, 2016, doi: 10.1186/s41044-016-0014-0.

[2] S. B. Kotsiantis and D. Kanellopoulos, "Data preprocessing for supervised leaning," *Int. J. ...*, vol. 1, no. 2, pp. 1–7, 2006, doi: 10.1080/02331931003692557.

[3] Y. Zhao, Z. Qiao, C. Xiao, L. Glass, and J. Sun, "PyHealth: A Python Library for Health Predictive Models," pp. 1–7, 2021, [Online]. Available: http://arxiv.org/abs/2101.04209.

[4] M. Zanfardino *et al.*, "MuSA: a graphical user interface for multi-OMICs data integration in radiogenomic studies," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, 2021, doi: 10.1038/s41598-021-81200-z.

[5] N. R. Gupta *et al.*, "dame-flame: A Python Library Providing Fast Interpretable Matching for Causal Inference," pp. 1–5, 2021, [Online]. Available: http://arxiv.org/abs/2101.01867.

[6] L. Neto, N. Pinto, A. Proença, A. Amorim, and E. Conde-Sousa, "4specid: Reference dna libraries auditing and annotation system for forensic applications," *Forests*, vol. 12, no. 1, pp. 1–15, 2021, doi: 10.3390/genes12010061.

[7] V. Bhagat, B. Robins, and M. O. Pallavi, "Sparx - Data Preprocessing Module," *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 1–6, 2019, doi: 10.1109/I2CT45611.2019.9033938.

[8] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, and R. G. Mark, "tableone: An open source Python package for producing summary statistics for research papers," *JAMIA Open*, vol. 1, no. 1, pp. 26–31, 2018, doi: 10.1093/jamiaopen/ooy012.

[9] Y. A. De Montjoye, L. Rocher, and A. S. Pentland, "Bandicoot: A python toolbox for mobile phone metadata," *J. Mach. Learn. Res.*, vol. 17, pp. 1–5, 2016.

[10] W. McKinney, "pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Perform. Sci. Comput.*, no. December, pp. 1–9, 2011.

[11] J. Bastin Robins, "COGNITO - Intuitive Auto Data Exploratory Toolkit," *2020 IEEE Int. Conf. Innov. Technol. INOCON 2020*