RESEARCH PAPER

FAIR Machine Learning Model Pipeline Implementation of COVID-19 Data

Sakinat Folorunso[1†], Ezekiel Ogundepo[2], Mariam Basajja[3], Joseph Awotunde[4], Abdullahi Kawu[5], Francisca Oladipo[6,7,8], Abdullahi Ibrahim[5]

[1]Department of Mathematical Sciences, Olabisi Onabanjo University, P.M.B 2002, Ago-Iwoye, Ogun State, Nigeria 120005, Nigeria

[2]Data Science Nigeria, Lagos 105102, Nigeria

[3]Leiden University, 1011NC, Amsterdam, the Netherlands

[4]Department of Computer Science, University of Ilorin, Ilorin, Kwara State, 240103, Nigeria

[5]Department of Computer Science, Ibrahim Badamosi University, Lapai, Niger State, 911101, Nigeria

[6]Kampala International University, 260101, Uganda

[7]Federal University Lokoja, Nigeria

[8]Virus Outbreak Data Network-Africa

Keywords: FAIRification; Semantic data model; Cluster analysis; FAIR data; Metadata; Machine learning model

Citation: Folorunso, S., Ogundepo, E., Basajja, M., Awotunde, J.B., Kawu, A.A., Oladipo, F.O., Ibrahim, A.: FAIR machine learning model pipeline implementation of COVID-19 data. Data Intelligence 4(4), 971–990 (2022). doi: 10.1162/dint_a_00182
Submitted: March 10, 2021; Revised: June 10, 2022; Accepted: July 15, 2022

ABSTRACT

Research and development are gradually becoming data-driven and the implementation of the FAIR Guidelines (that data should be Findable, Accessible, Interoperable, and Reusable) for scientific data administration and stewardship has the potential to remarkably enhance the framework for the reuse of research data. In this way, FAIR is aiding digital transformation. The 'FAIRification' of data increases the interoperability and (re)usability of data, so that new and robust analytical tools, such as machine learning (ML) models, can access the data to deduce meaningful insights, extract actionable information, and identify hidden patterns. This article aims to build a FAIR ML model pipeline using the generic FAIRification workflow to make the whole ML analytics process FAIR. Accordingly, FAIR input data was modelled using a FAIR ML model. The output data from the FAIR ML model was also made FAIR. For this, a hybrid hierarchical *k*-means

† Corresponding author: Sakinat Folorunso, Olabisi Onabanjo University (Email: sakinat.folorunso@oouagoiwoye.edu.ng; ORCID: 0000-0002-7058-8618).

(HHK) clustering ML algorithm was applied to group the data into homogeneous subgroups and ascertain the underlying structure of the data using a Nigerian-based FAIR dataset that contains data on economic factors, healthcare facilities, and coronavirus occurrences in all the 36 states of Nigeria. The model showed that research data and the ML pipeline can be FAIRified, shared, and reused by following the proposed FAIRification workflow and implementing technical architecture.

ACRONYMS

API	application programming interface
BSSE	between-group sum of squares
DSW	Data Steward Wizard
FAIR	Findable, Accessible, Interoperable, Reusable
FDP	FAIR Data Point
HHK	hybrid hierarchical *k*-means
ML	machine learning
PC	principal component
PCA	principal component analysis
RDF	Resource Description Framework
TWSS	total within the sum of squares
UPGMA	unweighted pair group method with arithmetic mean
VODAN	Virus Outbreak Data Network
WCSS	within-cluster sum of square
WSSE	within-group sum of squares

1. INTRODUCTION

Data FAIRification and its process workflow is a well-researched area. In 2020, Jacobsen et al. [1] described a generic stepwise FAIRification workflow for a team of multidisciplinary skilled persons steered by FAIR data stewards. This FAIRification process is applicable to all kinds of data as well as strange diseases resources. The progressive stages of the workflow are to detect the FAIRification goal, analyse the data and metadata, define the semantic data and metadata model, make data and metadata linkable, host FAIR data, and assess the FAIRness of the data. The authors also describe data processing methods, expertise requirements, which procedures and tools can be used, and which FAIR Guidelines they relate to. In addition, Weigel et al. [2] propose methods to improve the findability of workflows by extending machine capabilities, particularly by exploiting the Digital Object Architecture, common object operations, and machine learning (ML) techniques.

Research and development are gradually becoming data-driven and implementing the FAIR Guidelines (that data be Findable, Accessible, Interoperable, Reusable) [3] in relation to scientific data management and stewardship has the potential to remarkably enhance the reuse of research data [4]. At the FAIRification

stage of the generic workflow presented by Jacobsen et al. [1], semantic models for the study of data and metadata are generated. These models are templates for converting data and metadata into a machine-readable format. These newly-created semantic models can be made available for reuse over time. Some of the gains that will be made by implementing FAIR Guidelines could result in multiple improvements through machine readability (of data and metadata), which will enable the reuse of data and scalability. The short-term advantages, based on time and cost, include the improved findability of existing data, faster access to the data at scale, and easy selection of standardised and high-quality data for analytics such as ML [4].

In addition to research data, the FAIR Guidelines could also be applied to workflows, tools and algorithms that lead to optimum outcomes [5]. A workflow is an exact report of a multi-step process that synchronises many tasks and their data dependencies. A task could mean the execution of a computational process (running a code), the call for a service, the invocation of a command line tool, access to a database, or the implementation of a data processing script or workflow [6]. Hence, computational workflows represent the multi-step process used for data collection and preparation, analytics, modelling, and simulation that lead to new data products. They can inherently contribute to the FAIR Guidelines: by processing data according to established metadata; by creating metadata themselves during the processing of data; and by tracking and recording data provenance. An ML pipeline consists of a series of ordered steps used to automate the ML workflow [5]. Furthermore, ML models could visit FAIR data to detect hidden patterns and discover useful insights.

There are two main ML models: supervised and unsupervised learning models. The major task of supervised learning is classification and regression. Datasets belonging to this category are usually characterised by labelled data, while datasets belonging to unsupervised learning are unlabelled. The tasks of models belonging to this category are clustering, projection and visualisation. Clustering attempts to group unlabelled data into homogeneous subgroups by categorising related objects in a collection, which is used to aggregate the initial dataset into two or more subgroups [5]. In clustering modelling, *k*-means and hierarchical clustering algorithms are the most popular. The *k*-mean is computationally faster and produces tighter clusters than the hierarchical clustering algorithm. However, hierarchical clustering algorithms estimate a complete hierarchy of clusters and, hence, give more information than *k*-means. Yet, both of these algorithms have some drawbacks. The performance of *k*-means clustering largely depends on how effectively the initial count of clusters (i.e., the value of *k*) is determined, and the advantage of hierarchical clustering comes at the cost of low efficiency.

In 2015, Chen et al. [6] proposed a hybrid clustering model that combined the merits of two popular and efficient algorithms, called the hybrid hierarchical *k*-means (HHK) clustering model, for Eisen's yeast microarray datasets. This method produces a better-quality cluster, compared to already existing hybrid clustering and *k*-means clustering, based on two different distance metrics on microarray data. In addition, the HKK model has the capacity to treat outliers in the dataset. In 2011, Murugesan and Zhang [7] presented a hybrid of the bisecting *k*-means and unweighted pair group method with arithmetic mean (UPGMA) agglomerative hierarchical clustering model for documents. This hybrid initially clusters the documents with

the *k*-means algorithm to produce the total count of clusters, then the UPGMA is applied to these clusters to identify a homogeneous group of documents. The proposed model is subsequently compared with the standard bisecting *k*-means model, based on different clustering assessment metrics. The result showed that the model gave a superior performance to *k*-means.

This research aimed to extend the generic FAIRification workflow proposed by Jacobsen et al. [1] to an ML pipeline. This extension is valid as the FAIR Guidelines can also be applied to workflows, tools, pipeline and algorithms that lead to results, and not only to research data. This FAIRification will improve transparency and reproducibility, which is equal to the interoperability and reusability of the ML pipeline [8]. As the ML pipeline will be executed in a notebook (Jupyter), ProvBook [9] is used and the REPRODUCE-ME ontology [10] adopted. ProvBook can generate the executions in Resource Description Framework (RDF) using the following queries, among others: Which hyperparameters of the ML model were used in one particular run? Which libraries and corresponding versions were used to validate the ML model? What is the execution environment of the ML pipeline? How many training runs were performed in the ML pipeline? What is the train-test-validation percentage split of the data sample? What evaluation method was used to assess the model?

The input data used in this research consisted of social-economic, health, and COVID-19 incidence cases in Nigeria, which were analysed using a HHK clustering ML model [11]. The reason for using the hybrid approach was to combine the advantages of the two classic methods and minimise the disadvantages. The goal of the clustering analysing was to identify the set of Nigerian states with similar COVID-19 cases, identify underlying structures in the social-health-economic data, and summarise the relevant behaviours and characteristics. One reason for adopting the generic FAIRification workflow was that it is applicable to all types of data, even non-clinical data. It also addresses the four components of the FAIR Guidelines, which depend greatly on metadata.

This study is much needed due to the limited reporting of COVID-19 prevalence in Africa [12]. FAIRification includes the process of making data findable on the Internet by machines (and humans) across different locations. The nature of the FAIR data, held in residence and reachable through a FAIR data host, changes the nature of the data, in term of ownership and control, as well as the knowledge of the provenance of such data, requiring distributed data visiting. While the latter is not the explicit purpose of this study, carrying out an ML analysis of the FAIRified data can provide some insight into the potential of running algorithms over the Internet as computational machines to analyse data held in residence. The FAIRified data with rich metadata may provide new avenues to perform ML on data-sets held in residence. This study investigates the first step towards such an architecture.

2. METHODOLOGY

The methodology adopted for this study was divided into phases. The first phase applied the FAIRification workflow to the study dataset in order to make it 'FAIR'. Phase two involved the FAIRification of the

exploratory and pre-process task. This involved activities like data cleaning, wrangling, feature extraction and selection. The third phase applied the FAIRification workflow to the actual execution of the experiment and its environment to make it FAIR. In this context, the actual implementation of the ML model is regarded as an experiment and involved the type of ML algorithm used for analytics, the implementation language used (Python, R Language, C++, JAVA etc.), and the parameters of the particular algorithm for the generic FAIR model. The last phase applied the FAIRification workflow to the evaluation of the model after execution. This step looked at the particular metric used for the algorithm assessment and the result of the assessment. The whole process constitutes a FAIRification pipeline for the ML analysis of data (see Figure 1). The detailed generic FAIRification workflow process is illustrated in Figure 2.

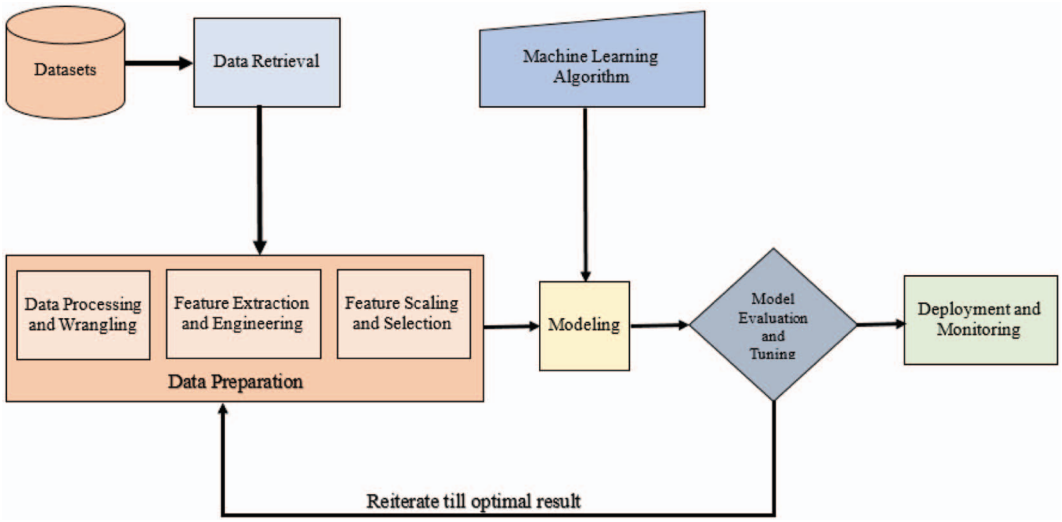

Figure 1. The ML pipeline for FAIRification workflow.

2.1 FAIRification

The object described in the FAIRification process could be: data, an algorithm, tools, a workflow or a pipeline. The process of making the object 'FAIR' involves two major steps, as suggested by Jacobsen et al. [1]. Firstly, the semantic data model for both the object and its corresponding metadata are defined and the objects and their metadata linked. Then, a licence is assigned and the FAIR object is published in a FAIR Data Point (FDP). The FAIRification process is thereby extended from data to object. The extended main generic workflow, as suggested by Jacobsen et al. [1], is shown in Figure 2.

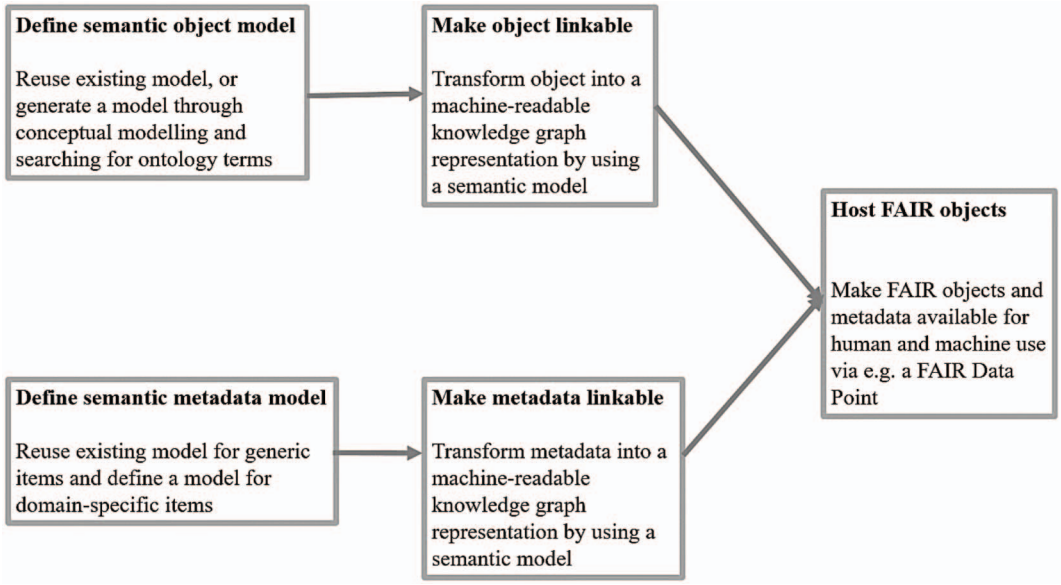

Figure 2. Extended generic FAIRification process workflow [1].

2.1.1 Preliminary FAIRification stage

2.1.1.1 Raw un-FAIR object

The initial stage is to determine the aim of FAIRification. This step requires access to the object, as well as an extensive knowledge and understanding of the object and the FAIR Guidelines. The workflow can be reiterated many times to accommodate more object elements. For the purpose of this article, the aim of FAIRification is to build a FAIR ML model pipeline and implement it using data on COVID-19 cases in Nigeria.

2.1.1.2 Analyse object

The next action is to analyse the object to prepare for future FAIRification. This analysis includes examining the object form, format and semantics, confirming that the object contains FAIR components by using FAIRness assessment tools [13, 14]. The object used in this study is presented in a standard format. There is no existing model for both input and output data, so a new model was built; however, there is an existing semantic model for ML pipeline that can be reused.

2.1.1.3 Analyse metadata

The next actionable step is to analyse the obtainability of metadata with reference to FAIR. This process flow may include examining the metadata, defining the object or redefining what metadata is to be gathered, and confirming that the metadata already contains FAIR components by using FAIRness assessment

tools [13, 14]. Improving metadata in terms of findability, accessibility, and reusability requires details such as: licence attribution, object versioning, indexing, aggregation [15], copyright, contribution statements (e.g., funders, data set creator, publisher), and description of use conditions and access to data.

2.1.2 Main FAIRification stage

2.1.2.1 Define semantic object and metadata model

The first step is to generate a semantic model for both the object and the metadata within the FAIRification stage of the workflow. The semantic model transforms the object and metadata into a machine-readable format. Although creating a semantic model is laborious, less effort will be required over time as more models are made available for reuse, especially if such models are treated as FAIR digital objects themselves. For the ML pipeline, there is an existing semantic model [16] and ontology [10] that can be reused. The concepts and relationship between the object elements are replaced with the machine-readable classes and properties from ontologies, vocabularies and thesauri. The comprehensive semantic object model differentiates between the object instances and their values and classes. This model is a representation of the data and exposes the meaning of the data in machine-readable terms. This enables the transformed FAIR object to be efficiently incorporated into other systems, the analysis workflow, and unforeseen future applications. The metadata semantic models relating to generic items are accessible to be reused, e.g., Data Catalog Vocabulary (DCAT). Domain-specific items, as defined by FAIR Guidelines F2 and R1.3, are decided by domain experts and later added to a semantic metadata model.

2.1.2.2 Make objects and metadata linkable

The second step is to transform objects and metadata into a FAIR format using a linkable machine-readable global framework like RDF. RDF provides a common and straightforward underlying model to create a powerful global virtual knowledge graph. The RDF of the pipeline was generated as a Turtle document in the Jupyter Notebook. This is achievable because the stages in the pipeline were implemented in the Notebook in the R programming language. The RDF of the input data was achieved with the FDP Data Steward Wizard (DSW), but could also be completed in semantic data platforms such as CEDAR or ELIXER.

2.1.2.3 Host FAIR data

The third step, is to host the FAIR object through the FDP [17]. This permits use by humans and machines through different interfaces like application programming interfaces (API), RDF Triple Store, and Web applications. The FDP machine interface will return a machine-readable RDF document.

2.1.3 Assess FAIR data

The final phase is to assess the FAIR object based on the original objectives and cross-check the FAIR status of the object and metadata using the FAIRness assessment tools [14, 18]. The FAIR object is kept in

RDF format, stored in RDF data stores (Triple Stores) and queried with SPARQL. The data is assessed through data visiting, as data are held in residence, but reachable through the FAIR data host.

2.2 REPRODUCE-ME Ontology [10]

The REPRODUCE-ME ontology [19, 20] was adopted for this study with the objective to capture the provenance of the experiment, execution environment and agents responsible for making the experiment FAIR. The PROV-O [21] and P-PLAN [22] ontologies were extended to form the REPRODUCE-ME ontology, which describes microscopy experiments, procedures, instruments used, people involved and results [23]. The prefix '*repr:*' is used to indicate the terms. The main concepts in the REPRODUCE-ME ontology were extended from the starting point terms of PROV-O, which include *prov:Entity*, *prov:Agent* and *prov:Activity*. An 'entity' is an item that can be physical, digital, or conceptual, with some fixed aspects, while an 'activity' is an event occurring over a period of time and acts upon or with entities. An activity may include consuming, processing, transforming, modifying, relocating, using, or generating entities. An 'agent' takes responsibility for an activity occurring, for another agent's activity taking place or for the existence of an entity. The main entity, *repr:Experiment*, which extends from *prov:Entity*, connects various concepts and relations in the ontology.

This study also made use of ProvBook [21], which is an extension of the Jupyter Notebook, to implement the ML pipeline. It captured and tracked the provenance of the cells over the course of time. The provenance data — which includes the runs of the cells in the Notebook, the start and end time of the execution, the length of time of running each cell, the source and output of the cells — are stored in the metadata of the cell in the Notebook whenever the code cell is executed. The provenance of Jupyter Notebook executions is downloaded in machine-readable format (RDF in Turtle document), along with the provenance traces and execution environment attributes. The major features provided by the ProvBook are the provenance, machine readability, and the difference of runs of Jupyter Notebook.

Figure 3 shows the provenance of code cells alongside their input and output for the study dataset and ML algorithm. The study ProvBook, its metadata, input data and RDF file are stored in the project's GitHub[24] and VODAN-Africa and Asia FDPs. Figure 4 shows a representation of the provenance of the Notebook using the REPRODUCE-ME ontology. The Notebook is described as a *p-plan:Plan* with the cells as *p-plan:Step* of the plan. Every run of a cell is represented as *CellExecution*, which is a subclass of *p-plan:Activity*. The *CellExecution* uses input and generates output with a start and end time. The Notebook is attributed to the authors using the object property *prov:wasAttributedTo*. The provenance information, including the kernel (the Interactive Notebook computation engine that executes the code written in a programming language [scripting language] like Python, R or Java) and its version (property describing the differences with reference to its previous form), is available in the downloaded version of RDF.

Figure 3. Code cells with provenance data executions.

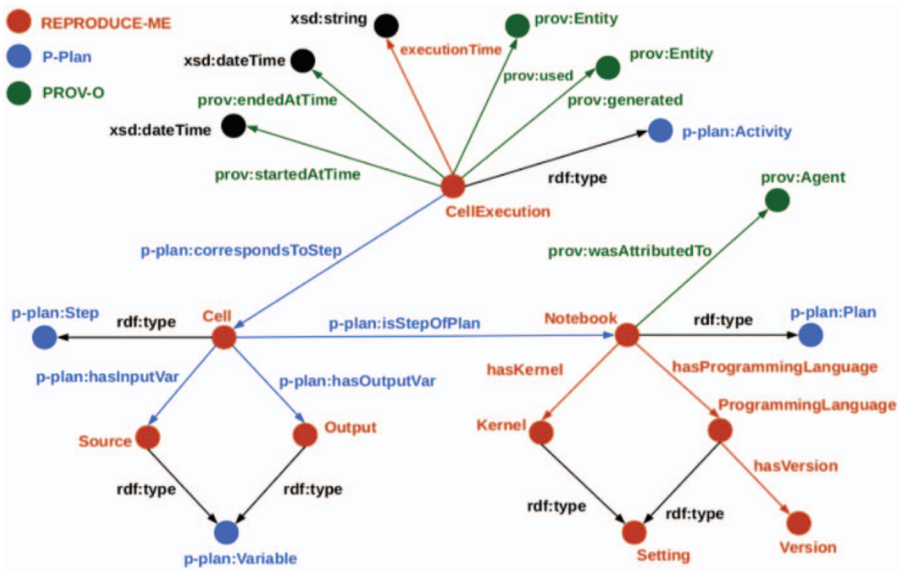

Figure 4. The Notebook provenance presented using the REPRODUCE-ME ontology [21].

2.3 Study Dataset Description and Semantic Modelling

The study dataset [12], which is the input data for the model, is on the COVID-19 pandemic and covers all states in Nigeria. The data were collected from different sources including the National Center for Disease Control (NCDC), Geo-Referenced Infrastructure and Demographic Data for Development (GRID3), Geographic Population and Demographic Data (GeoPoDe), and National Bureau of Statistics (NBS). The variables in the data consist of geo coordinates, population disaggregated by gender, COVID-19 cases, healthcare facilities, COVID-19 government laboratories, and budget allocation for each state. Some additional variables were created from the existing features such as discharge rate and COVID-19 fatality rate, as presented in Table 1.

Table 1. The study dataset description.

Variable	Description	Data type
State	Name of each state in Nigeria	String
Geo-political zone	One of the six geopolitical zones	String
Geographical coordinates: • Latitude • Longitude	Coordinates of each state in Nigeria	Numeric
Population estimate: • Female • Male	Population by gender for each state in Nigeria	Numeric
COVID-19 cases: • Confirmed • Admitted • Discharged • Death • Discharge rate • Fatality rate	Total count incidence of COVID-19 cases in each state in Nigeria $Fatality\ rate = \frac{COVID-19 death cases}{COVID-19 confirmed dases}$ The fatality rate is the proportion of deaths from a certain disease compared to the total number of people diagnosed with the disease for a particular period.	Numeric
State budget in (billion Nigerian Naira): • 2019 • 2020 • 2020 revised • % reduction	Amount of budget in billion Nigerian Naira available to each state in Nigeria for 2019–2020	Numeric
Total revenue: • Federation Account Allocation Committee (FAAC) 2019 • Internal Generated Revenue (IGR) 2019 • Revenue 2019 • Revenue 2018	FAAC is the total revenue in billion Nigerian Naira available to each state in Nigeria.	Numeric
Healthcare: • Primary • Secondary • Tertiary	Number of healthcare facilities available in each state	Numeric
COVID-19 laboratory	Number of COVID-19 government laboratories available in each state	Numeric

The semantic modelling of the input dataset is shown in Figure 5. The data schema shows the classes and relationships between the classes. The schema contains classes representing different aspects of the COVID-19 dataset [24].

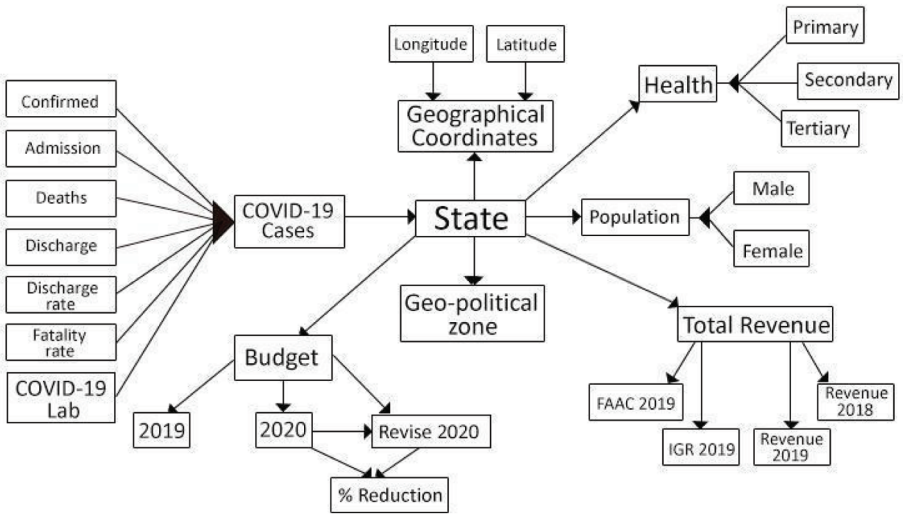

Figure 5. Core classes in data schema.

Note: Boxes indicate classes; arrows signify properties, subclass relations and part-of relations.

2.4 Clustering Analysis

Clustering is a sort of unsupervised ML model in which different groups of objects are separated into homogeneous subgroups. Assuming a task of clustering p objects to x_j, $j \in \{1, \ldots, p\}$ and that each object x_j is an n-dimensional matrix $(x_{1j}, \ldots, x_{nj})^T$. Let $d(x_j, x_{j'})$ represent the dissimilarity between objects x_j and x_j and let D be the $p \times p$ symmetrical vector of dissimilarities. A typical measure of dissimilarity includes distance, correlation, absolute correlation and cosine-angle. Clustering routines (either implicitly or explicitly) map a distance vector D into p cluster labels [25]. Hierarchical and k-means clustering are the two most prevalent clustering methods. Yet, they both have their drawbacks. Hierarchical clustering cannot represent unique clusters with similar expression patterns. Moreover, the real expression patterns become less relevant as the cluster grows in size. In contrast, k-means clustering is sensitive to outliers, requires the count of clusters (k) be predefined, and the initial centroid is chosen randomly [26].

2.4.1 Hierarchical k-means clustering

The hybrid process is a combination of the hierarchical and k-means clustering models. The hierarchical clustering algorithm uses the average linkage method and is checked at the distance between the two consecutive nodes of the hierarchy that represents the maximum. Using this information, the value of k is determined, which is then fed into the k-means clustering algorithm to produce the final clusters. In both

algorithms, the Pearson correlation coefficient (r) was used as the similarity metric between two samples and ($1-r$) to measure the distance. Hierarchical clustering creates a tree of clusters called a dendrogram by splitting/merging each cluster on each level until an optimum number of clusters is created.

The steps in the HHK clustering algorithm are as follows:

- Step 1: Perform a total agglomerative hierarchical clustering on the dataset and record the count of clusters created during the process.
- Step 2: Repeat the run on the hierarchical clustering and cut the trees until optimum count of clusters is created.
- Step 3: Execute the k-means clustering with the group of cluster centres in the previous step as the initial cluster centres.

2.4.2 Model measure

2.4.2.1 Within-cluster sum of square

For a given set of clusters $S = (s_1, s_2 \ldots s_k)$, with centres $(\mu_1, \mu_2 \ldots \mu_k)$, the within-cluster sum of squares (WCSS) is defined as equation (1):

$$WCSS = \sum_{i=1}^{k} \sum_{x \in S_i} \left\| x - \mu_i \right\|^2 \tag{1}$$

Where $\|Z\|^2 = \sum z_i^2$ is squared Euclidian distance of $Z = (z_1, z_2 \ldots z_n)$.

2.4.2.2 Determining the optimal number of clusters

The optimal number of clusters was determined using the elbow method. The basic idea is to define clusters such that the within-cluster variation is minimised, as observed in equation (2):

$$WSS = min\left(\sum_{k=1}^{k} W\left(C_k\right)\right) \tag{2}$$

Where C_k is the k^{th} cluster and $W(C_k)$ is the within-cluster variation.

Thus, the steps in the algorithm to define the optimal clusters are as follows:

- Step 1: Compute clustering algorithm for different values of k, for instance, by varying k from 1 to 10 clusters.
- Step 2: For each k calculate the total WCSS.
- Step 3: Plot the curve of WCSS according to the number of clusters k.
- Step 4: The location of a bend (elbow) in the plot is generally considered to be an indicator of the appropriate number of clusters.

3. RESULTS

This section presents the result obtained from clustering the social-economic-health data in Nigeria based on the incidence of COVID-19. A hybrid model of the hierarchical and *k*-means models was used to combine the merits of the two models.

3.1 Clustering Modelling

The HHK clustering modelling and analysis were carried out in R programming language (RJupyter Notebook) and the project script can be found at GitHub [24]. Subsequently, HHK was initially performed by splitting the original, but FAIR, data into *k* homogeneous groups. The elbow method was applied to choose an optimum value for *k* [27]. According to the elbow method, the total within the sum of squares (TWSS) was defined for each value of *k*.

Figure 6 displays the values of TWSS for different values of *k*. In Figure 6, the count of clusters increases from 1 to 3 and the value of TWSS suddenly decreases and continues onwards. The results recommend three (3) as the optimum clustered number [27]. The dendrogram for the three clusters, separated by three colours based on which state the data relates to, is shown in Figure 7. The input variables for the hybrid clustering model were: state, population_female, population_male, #Confirmed, #Admission, #Deaths, discharge_rate, fatality_rate, 2020_initial_budget, 2020_revised_budget, primary_health_care, secondary_health_care, and tertiary, covid_19_lab. All variables in the model were normalised and the three clusters that were generated are distributed across the states in Nigeria.

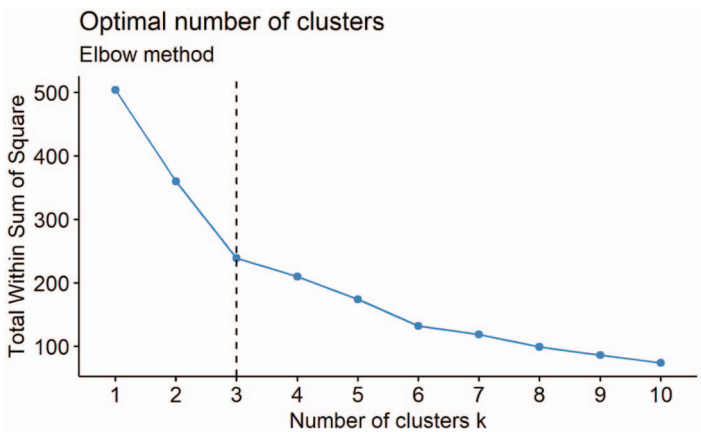

Figure 6. Defining the optimum value for *k*.

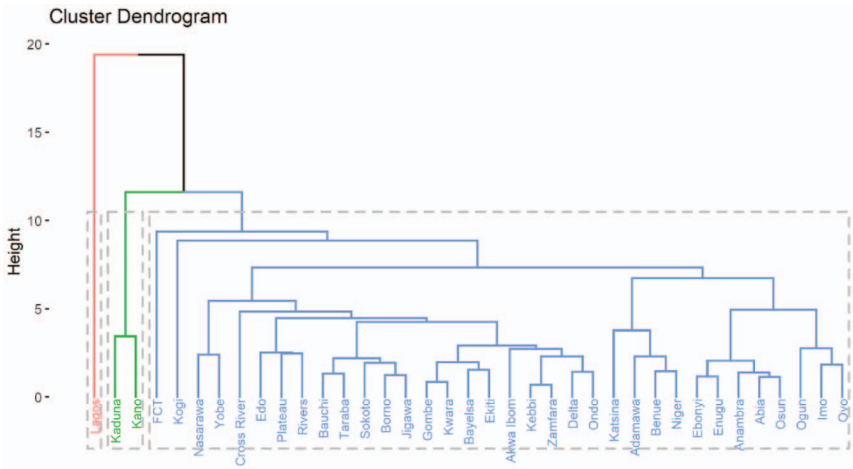

Figure 7. Cluster dendrogram of HHK for $k = 3$.

The cluster plot is a graphical display that simultaneously describes the objects alongside their interrelations and corresponding clusters. This enables us to picture the size and shape of each of the clusters and their position. Because there are more than two variables in the dataset, a principal component analysis (PCA) was performed to reduce the dimension [28] and plot the data points according to the first two principal components (PCs) that explain 100% of the point variability. Hence, the plot displays the data points relative to the PCs, instead of the original axes. Component 1, which consists of the largest dispersion, is plotted on the x-axis, while component 2 is plotted on the y-axis (Figure 8).

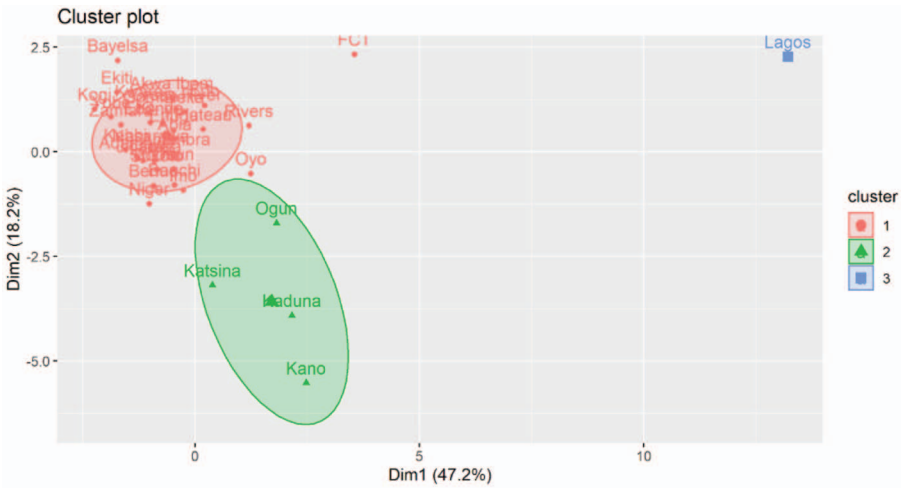

Figure 8. Cluster plot of the Nigerian COVID-19 FAIR data by HHK.

In Figure 8, the cluster plot presents the three clusters of sizes 32, 4 and 1. These clusters are represented by ellipses and distinguished by the colours red, green and blue, with 47.2% variability. The ellipses are based on the average and the covariance matrix of each of the three clusters. The size of each ellipse spans all points in the respective cluster.

3.2 Output Data

The output data is the result of the analysis of the ML model. This could also be represented in a machine-actionable format. In this study, using a hybrid clustering technique, three different clusters were generated consisting of Nigerian states with similar COVID-19 cases. Table 2 shows the description of the dataset.

Table 2. Output dataset description.

Cluster	Description
Cluster 1 Cluster 2 Cluster 3	States in Nigeria with similar COVID-19 cases

The semantic modelling for the output dataset is shown in Figure 9. The data schema shows the classes and the relationships between the classes. The schema contains classes representing different clusters of the COVID-19 dataset from the result analysis.

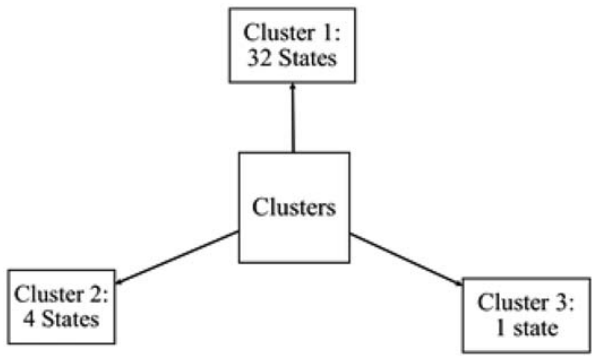

Figure 9. Core classes in the data schema.
Note: Boxes indicate classes; arrows signify properties and relations

3.3 Cluster Table

The detailed results of the HHK clustering evaluation of the three clusters are presented in Table 3. Column 1 lists the three clusters and column 2 describes the number of items (states) in individual clusters. The 'Centres' column shows the mean values within each cluster. The between-group sum of squares (BSSE) is the error between the different clusters, while the within-group sum of squares (WSSE) is the error within

the different clusters. The total sum of squares error column is the sum of the BSSE and WSSE. The variance of the model, which is the fraction of BSSE and total sum of squares, is 0.526. This value indicates that it is a good model, as it is close to 1.

Table 3. Hybrid hierarchical *k*-means cluster table.

Clusters	Vectors	Centers														
		Pop_female	population_male	Confirmed	Admission	Discharged	Deaths	disc_rate	fatality_rate	2020_ini_budget	2020_rev_budget	pry_health_care	sec._health_care	tertiary	covid_19_lab	State
1	32	-0.319	-0.318	-0.182	-0.122	-0.185	-0.174	-0.017	0.062	-0.152	-0.197	-0.177	-0.101	-0.280	-0.197	Abia
2	4	2.098	2.065	0.100	0.127	0.090	0.176	0.009	-0.368	0.040	0.281	1.261	-0.289	2.237	0.720	Kaduna
3	1	1.803	1.911	5.420	3.401	5.548	4.867	0.500	-0.500	4.689	5.189	0.627	4.376	0.006	3.435	Lagos

For example, in Table 3, for Cluster 3, one item was grouped together with variables like state, population_female, population_male, #Confirmed, #Admission, #Deaths, discharge_rate, fatality_rate, 2020_initial_budget, 2020_revised_budget, primary_health_care, secondary_health_care, tertiary, and covid_19_lab. The mean values for these variables are 1.803, 1.911, 5.420, 3.401, 5.548, 4.867, 0.500, –0.500, 4.689, 5.189, 0.627, 4.376, 0.006 and 3.435, respectively. The WCSS for Clusters 1, 2 and 3 are 212.786, 26.315 and 0.00, respectively.

4. DISCUSSION

In this study, the generic FAIRification workflow [1] was applied to build a FAIR ML model pipeline, which was implemented using Nigeria's daily COVID-19 incidence cases. A semantic model of both the input and output data and ML models was presented. We also extended the concept to object, which was defined as either data, the ML model or the workflow. Fortunately, an existing semantic model for ML was reused for FAIRification. The generic workflow for the object FAIRification process is described in this article. The goal of this generic workflow was to ease the FAIRification process.

However, there are detailed decisions that are far from this generic workflow that call for the attention of stakeholders involved in the FAIRification. Hence, a FAIR data and ML model pipeline was created. A hybrid of clustering ML model was applied to the FAIR data for analysis. The elbow method was used to determine the optimum value of *k*. A HHK with *k*=3 clusters was obtained from analysis. A dendrogram of the hierarchical clustering showing the clustered states was produced (Figure 6) and clusters plotted for each state based on PCA and the variation (Figure 7).

The results show the evaluation of the hybrid clustering algorithm and hidden pattern in the data (Table 3). Three clusters were identified (Figure 7); this outcome categorises the states into clusters of high (Cluster 3), medium (Cluster 2) and low (Cluster 1) values for COVID-19 incidence cases. Cluster 3 contains only one item (Lagos state), which is characterised by high values for lab-confirmed, discharged and death for COVID-19 incidence cases. It also has the highest number of secondary health facilities with a highly-revised budget to accommodate the planning and managing COVID-19 cases.

One of the reasons for the high incidence could be that Lagos is a major financial centre and economic hub for all other states in Nigeria and Africa. This megacity has the fourth-highest GDP in Africa [29] and houses one of the largest and busiest seaports on the continent [30]. Cluster 2 contains four items, namely: Ogun, Kano, Kaduna and Katsina states. They are characterised by moderate values for COVID-19 incidence cases. Kaduna State, which is the centre of this cluster is bordered by Kano and Katsina. Kaduna is an industrial centre in Northern Nigeria; it has an international airport, major train station and road transportation hub servicing surrounding agricultural areas and connecting traders all over the country [31]. Ogun state is the only state that shares a border with Lagos state. It hosts many industrial estates and is a major manufacturing hub in Nigeria. Its high population is due to the influx of people working in Lagos, but who live in Ogun. Cluster 1 contains the largest number of items (34) states. The centre of this cluster is Abia State, which is the fifth most industrialised state in Nigeria. Abia also has the fourth highest index of human development, with many economic activities and a fast-growing population [32]. One of the reasons for the low incidence of the COVID-19 data in Cluster 1 could be incorrect data recorded or, because it is removed from the capital and, consequently, access to testing for COVID may have been less, suppressing the figures on incidence.

5. CONCLUSION

This study investigated the clustering of COVID-19 data following a hybrid clustering ML model. The aim was to perform clustering on data that were curated according to the FAIR Guidelines, which require data to be Findable, Accessible, Interoperable and Reusable. A FAIRification pipeline was implemented using ProvBook and REPRODUCE-ME ontologies, including creating machine-readable semantic data, which served as input for the execution of the task, defined as clustering the incidence of COVID-19. While the process of creating semantic and linked data may be laborious at first, it is expected to become easier with time. The FAIRification procedure has the additional potential of visiting data held in residence, with the related advantage of the retention of data ownership during the execution of data analytics. The FAIRification of data was carried out through the DSW, producing machine-readable data as input. The output identified three clusters, with the first cluster showing a high incidence of COVID-19 infections in Lagos, the capital of Nigeria and a busy financial centre and seaport. The second and third clusters each had a lower incidence of COVID-19.

The findings of this study, if replicated regularly, would provide good insight into the evolution of COVID-19 prevalence in Nigeria. This is highly relevant, as COVID-19 data from Africa are limited and underreported. The method tested in this study shows that this situation could be improved using the combined methods of data FAIRification, data visiting and ML based analytics. It also found that the FAIRification of data does not affect the ML analysis result. In the future, the REPRODUCE-ME ontology needs to be expanded to accommodate the ML pipeline.

ACKNOWLEDGMENTS

We thank Mirjam van Reisen and Erik Schultes for their comments and contributions to the article. We are also grateful to the VODAN-Africa Nigeria team at Olabisi Onabanjo University, Data Science Nigeria (DSN), and Ibrahim Badamosi University (IBBUL) for providing data stewards. We would like to thank Misha Stocker for managing and coordinating this Special Issue (Volume 4) and Susan Sellars for copyediting and proofreading. Finally, we acknowledge VODAN-Africa, the Philips Foundation, the Dutch Development Bank FMO, CORDAID, and the GO FAIR Foundation for supporting this research.

AUTHORS' CONTRIBUTIONS

Sakinat Folorunso (sakinat.folorunso@oouagoiwoye.edu.ng, ORCID: 0000-0002-7058-8618): Conceptualization, Data curation, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing. **Ezekiel Ogundepo** (ogundepoezekiel@gmail.com, ORCID: 0000-0003-3974-2733): Data curation, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing. **Mariam Basajja** (mariam.basajja@gmail.com, ORCID: 0000-0001-7710-8843): Methodology, Resources, Software, Validation, Writing – original draft, Writing – review & editing. **Joseph Awotunde** (awotunde.jb@unilorin.edu.ng, ORCID: 0000-0002-1020-4432): Investigation, Resources, Validation, Writing – original draft, Writing – review & editing. **Abdullahi Kawu** (abdullahikawu@ibbu.edu.ng, ORCID: 0000-0003-2531-9539): Methodology, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing. **Francisca Oladipo** (francisca.oladipo@kiu.ac.ug, ORCID: 0000-0003-0584-9145): Formal Analysis, Investigation, Resources, Supervision, Writing – original draft, Writing – review & editing. **Ibrahim Abdullahi** (ibrojay01@ibbu.edu.ng, ORCID: 0000-0002-3467-1203): Formal Analysis, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing.

CONFLICT OF INTEREST

All of the authors declare that they have no competing interests.

ETHICS STATEMENT

Tilburg University, Research Ethics and Data Management Committee of Tilburg School of Humanities and Digital Sciences REDC#2020/013, June 1, 2020–May 31, 2024 on Social Dynamics of Digital Innovation in remote non-western communities

Uganda National Council for Science and Technology, Reference IS18ES, July 23, 2019–July 23, 2023

REFERENCES

[1] Jacobsen, A., Kaliyaperumal, R., da Silva Santos, L.O.B., Mons, B., Schultes, E., Roos, M., Thompson, M.: A generic workflow for the data FAIRification process. Data Intelligence 2, 56–65 (2020)

[2] Weigel, T., Schwardmann, U., Klump, J., Bendoukha, S., Quick, R.: Making data and workflows findable for machines. Data Intelligence 2, 40–46 (2020)
[3] Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3(1), 1–9 (2016)
[4] Wise, J., de Barron, G.A., Splendiani, A., Balali-Mood, B., Vasant, D., Little, E., et al.: Implementation and relevance of FAIR data principles in biopharmaceutical R & D. Drug Discovery Today 24(4), 933–938 (2019)
[5] Samuel, S., Löffler, F., König-Ries, B.: Machine learning pipelines: Provenance, reproducibility and FAIR Data Principles [Online]. Cornell University, arXiv, 2006.12117v1 [cs.LG] (22 June 2020). Available at: https://arxiv.org/abs/2006.12117. Accessed 4 January 2022
[6] Goble, C., Cohen-Boulakia, S., Soiland-Reyes, S., Garijo, D., Gil, Y., Crusoe, M.R., Peters, K. Schober, D.: FAIR computational workflow. Data Intelligence 2, 108–121 (2020)
[7] Nguyen, H., Bui, X.-N., Tran, Q.-H., Mai, N.-L.: A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical K-means clustering and cubist algorithms. Applied Soft Computing Journal 77, 376–386 (2019)
[8] Chen, B., Tai, P.C., Harrison, R., Pan, Y.: Novel hybrid hierarchical-k-means clustering method (H-K-means) for microarray analysis. In: Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference Workshops (CSBW'05) (2005)
[9] Murugesan, K., Zhang, J.: Hybrid hierarchical clustering: An experimental analysis. Technical Report, University of Kentucky, Lexington (2011)
[10] GitHub: ProvBook [Online]. GitHub (2021). Available at: https://github.com/Sheeba-Samuel/ProvBook. Accessed 1 March 2021
[11] Samuel, S., K'onig-Ries, B.: REPRODUCE-ME: Ontology-based data access for reproducibility of microscopy experiments. In: The Semantic Web: ESWC 2017 Satellite Events, LNCS 10577, pp. 17–20 (2017)
[12] Hasan, M.S., Duan, Z.-H.: Hierarchical k-means: A hybrid clustering algorithm and its application to study gene expression in lung adenocarcinoma. In: Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology, Elsevier Inc., pp. 51–67 (2015)
[13] Ogundepo, E., Folorunso, S., Adekanmbi, O., Akinsande, O., Banjo, O., Ogbuju, E., et al.: An exploratory assessment of a multidimensional healthcare and economic data on COVID-19 in Nigeria. Data in Brief 33, 106424 (2020)
[14] De Miranda Azevedo, R., Dumontier, M.: Considerations for the conduction and interpretation of FAIRness evaluations. Data Intelligence 2, 285–292 (2020)
[15] Wilkinson, M.D., Sansone, S.A., Schultes, E., Doorn, P., da Silva Santos, L. O. B., Dumontier, M.: Comment: A design framework and exemplar metrics for FAIRness. Scientific Data 5, 1–4 (2018)
[16] Sinaci, A.A., Núñez-Benjumea, F.J., Gencturk, M., Jauer, M.-L., Deserno, T., Chronaki, C., et al.: From raw data to FAIR Data: The FAIRification workflow for health research. Methods of Information in Medicine 59, e21–e32 (2020)
[17] Publio, G.C., Esteves, D., Ławrynowicz, A., Panov, P., Soldatova, L., Soru, T., Vanschoren, J., Zafar, H.: ML-schema: Exposing the semantics of machine learning with schemas and ontologies. In: Enabling Reproducibility in Machine Learning MLTrain@RML (co-located with ICML 2018) (2018)
[18] Thompson, M., Burger, K., Kaliyaperumal, R., Roos, M., da Silva Santos, L.O.B.: Making FAIR easy with FAIR tools: From creolization to convergence. Data Intelligence 2, 87–95 (2020). doi: 10.1162/dint_a_00031
[19] Wilkinson, M.D., Dumontier, M., Sansone, S.A., da Silva Santos, L.O.B., Prieto, M., Batista, D., et al.: Evaluating FAIR maturity through a scalable, automated, community-governed framework. Scientific Data 6(1), 174 (2019)

[20] GitHub: REPRODUCE-ME [Online]. GitHub (2021). Available at: https://github.com/Sheeba-Samuel/REPRODUCE-ME. Accessed 9 April 2021
[21] BioPortal: Library of ontologies [Online]. BioPortal (2021). Available at: https://bioportal.bioontology.org/ontologies. Accessed 8 April 2021
[22] Samuel, S., König-Ries, B.: ProvBook: Provenance-based semantic enrichment of interactive notebooks for reproducibility. In: The 17th International Semantic Web Conference (ISWC) 2018 Demo Track (2018)
[23] Garijo, D., Gil, Y.: The P-Plan ontology [Online]. Vocab (2014). Available at: http://vocab.linkeddata.es/p-plan/index.html. Accessed 22 March 2021
[24] Samuel, S.: Integrative data management for reproducibility of microscopy experiments. In: E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, O. Hartig (eds), The Semantic Web. ESWC 2017. Lecture Notes in Computer Science, Vol. 10250, Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58451-5_19
[25] GitHub: Clustering-analysis [Online]. GitHub (2021). Available at: https://github.com/sakinatfolorunso/Clustering-analysis. Accessed 7 March 2021
[26] Van der Laan, M.J., Pollard, K.S.: A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap. Journal of Statistical Planning and Inference 117, 275–303 (2003)
[27] Chen, B., He, J., Pellicer, S., Pan, Y.: Using hybrid hierarchical k-means (HHK) clustering algorithm for protein sequence motif super-rule-tree (SRT) structure construction. International Journal of Data Mining and Bioinformatics 4(3), 316–330 (2010)
[28] Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in K-means clustering. International Journal of Advanced Research in Computer Science and Management Studies 1, 90–95 (2013)
[29] Pison, G., Struyf, A., Rousseeuw, P.J.: Displaying a clustering with CLUSPLOT. Computational Statistics & Data Analysis 30, 381–392 (1999)
[30] Jacobs, F.: These cities are the hubs of Africa's economic boom [Online]. Big Think (4 October 2018). Available at: https://bigthink.com/strange-maps/richest-cities-in-africa. Accessed 3 March 2021
[31] Businesstech: Africa's biggest shipping ports [Online]. Businesstech (8 March 2015). Available at: https://businesstech.co.za/news/trending/81995/africas-biggest-shipping-ports/. Accessed 1 April 2021
[32] Wikipedia: Kaduna [Online]. Wikipedia (2021). Available at: https://en.wikipedia.org/wiki/Kaduna. Accessed 21 March 2021
[33] Wikipedia: Abia State [Online]. Wikipedia (2021). Available at: https://en.wikipedia.org/wiki/Abia_State#cite_note-www.nddc.gov.ng-5. Accessed 21 March 2021