DataAssist: A Machine Learning Approach to Data Cleaning and Preparation

Kartikay Goyle[1(✉)], Quin Xie[2], and Vakul Goyle[3]

[1] Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada
kartikay.goyle@mail.utoronto.ca

[2] Department of Medical Biophysics, University of Toronto, Toronto, Canada
quin.xie@mail.utoronto.ca

[3] School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
vakul002@e.ntu.edu.sg

Abstract. Automated machine learning (ML) tools have primarily focused on model selection and parameter optimization, leaving a gap in the area of data cleaning and preparation. This paper presents DataAssist, an automated data preparation and cleaning platform that enhances dataset quality using ML-informed methods. DataAssist provides a pipeline for exploratory data analysis and data cleaning, including generating visualization for user-selected variables, unifying data annotation, suggesting anomaly removal, and preprocessing data. The exported dataset can be readily integrated with other autoML tools or user-specified model for downstream analysis. Our data-centric tool is applicable to a variety of fields, including economics, business, and forecasting applications saving over 50% time of the time spent on data cleansing and preparation. This paper aims to fill the gap in the literature by providing a comprehensive tool for data cleaning and preparation, which is often overlooked in the current ML landscape. We also present a comparative analysis of DataAssist with other existing technologies, demonstrating its superior performance and efficiency.

Keywords: ML-enabled data cleaning · Active learning · Unsupervised anomaly detection

1 Introduction

The rising availability of large datasets and computational power have enabled increased employment of multi-parameter, complex ML models and facilitated a wealth of autoML platforms that make ML models accessible for individuals with limited machine learning and programming expertise [1–3]. However, current autoML platforms do not provide support for the quality and integrity

K. Arai (Ed.): IntelliSys 2024, LNNS 1067, pp. 476–486, 2024.
https://doi.org/10.1007/978-3-031-66431-1_33

constraints of the dataset. Sub-optimal data quality may negatively impact ML model performance, though some models are insensitive to dirty data. Despite the joint cleaning data/ML problem being recognized by both ML and database (DB) communities, there is no standardized solution in a common workflow for a data analyst. This gap in the literature is what our research aims to address.

Data analysts typically perform exploratory data analysis (EDA) by plotting variable distributions and inspecting trends between variables to determine potential predictors of the response variable. They input a subset of the data into a preliminary ML model and discover idiosyncrasies in the dataset during training or earlier EDA. They would need to manually clean the data or implement some data-cleaning libraries, and retrain the model iteratively until there is no dirty data. This iterative nature of the procedure makes data preprocessing unnecessarily time-consuming and repetitive, taking time away from the ultimate goal of making interpretations and deriving knowledge. A centralized tool for principled data cleaning will effectively free data scientists from the laborious process of data preparation, integration, and management.

Over the years, the database community has developed different types of data-cleaning tools but the data-cleaning ecosystem remains diffuse. The analytics-driven data cleaning tools reduce the cost of data cleaning by simultaneously integrating data cleaning and training. For example, ActiveClean [4] takes advantage of the gradient descent method, allowing the cleaning of small batches of data on ML models with convex loss, such as linear regression and mixture models. BoostClean [5] addresses a selected set of errors with statistical boosting, but only considers when an attribute value is outside of its value domain. Some major disadvantages of these methods are repeated efforts in cleaning datasets when fitting different ML models, and difficulties in comparing across model performance.

Another paradigm of data cleaning tools takes advantage of ML methods in correcting data annotation. Scared [6], GDR [7] and HoloClean [8] use different ML models to come up with probabilities for detected errors and suggest repair. In addition, active learning [9] has been used to prompt the users to label specific data potentially erroneous and receive timely feedback to prevent error propagation in downstream analysis. Nonetheless, there is no streamlined process to prepare the data for ML models. The data cleaning system is decentralized, with each tool specialized for a limited subset of error detection and repair.

In this paper, we present DataAssist, an ML-based pipeline integrating data exploration and cleaning that allows users to select, combine, and order different steps that best suit their data analysis scheme. The package covers the four most common sources of error in dirty data: missing values, outliers, duplicates, and inconsistencies. In addition, DataAssist provides data exploration and transformation functions, such as enumerating variable types, generating data visualization, ranking feature importance, adjusting data skewness, encoding categorical variables, and normalization. We designed a user interface that allows individuals with minimal to no coding experience to easily employ our tool and export datasets for downstream analysis.

The rest of the paper is structured as follows: Sect. 2 provides a detailed description of the DataAssist tool and its functionalities. Section 3 presents a comparative analysis of DataAssist with other existing technologies, demonstrating its superior performance and efficiency. Section 4 discusses the implications of our findings and potential future directions for this research. Finally, Sect. 5 concludes the paper by summarizing the key contributions and findings of our research.

2 Feature Overview

Our goal is to develop a standardized pipeline to automatically clean and prepare the data such that the users can specify their needs and directly integrate the data cleaning process with the existing autoML tools for the next steps such as feature engineering and model selection. We leveraged ML models to predict the most suitable methods for Exploratory data analysis (EDA) and preprocessing for each dataset.

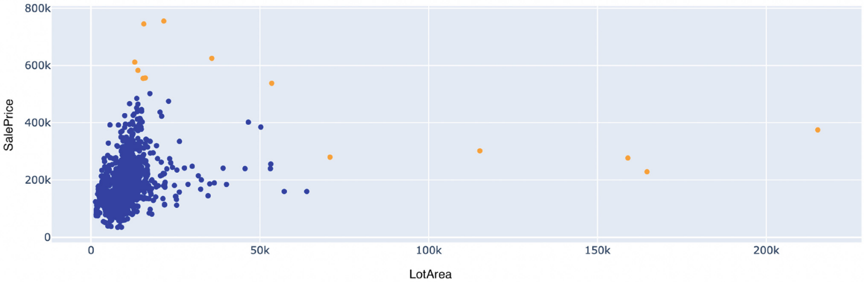

Fig. 1. Example output of anomaly detection from DataAssist. The plot is generated by running DataAssist on the House Prices dataset introduced in Sect. 3. The x-axis is Lot size in square feet (LotArea) and the y-axis is the SalePrice of residential homes. The outliers are detected by DBSCAN and highlighted in orange, distinguished from the majority of the dataset in blue. These points will be removed in the subsequent data-cleaning steps

2.1 Exploratory Data Analysis

First, EDA is the primary task of data analysis for research questions designed to hypothesis-generating rather than hypothesis-driven. Moreover, by inspecting the dataset during EDA, one may find idiosyncrasies in the dataset, such as missing values and outliers, skewed distribution, or multi-collinearity across variables. Therefore, we placed EDA at the forefront of DataAssist pipeline. Users can review data quality through a number of visualizations:

Table 1. Example training dataset for the SVM model predicting EDA

Variable 1 name	Variable 1 type	Variable 1 distribution	Variable 2 name	Variable 2 type	Variable 2 distribution	Plot type	Variable relation	Correlation coefficient
Age	Continuous	Normal	Income	Continuous	Skewed right	Scatter plot	Positive linear	High positive (.8)
Gender	Categorical	Equal male/female	Purchase	Categorical	Varied	Bar chart	No Relation	Not applicable
Product_type	Categorical	Varied	Sales	Continuous	Normal	Violin plot	Positive relation	Not applicable
Experience	Continuous	Normal	Salary	Continuous	Skewed right	Scatter plot	Positive linear	High positive (0.7)
Education	Ordinal	Varied	Genders	Categorical	Equal male/female	Bar chart	No relation	Not applicable
Skill_level	Continuous	Normal	Task_time	Continuous	Normal	Line graph	Negative linear	High negative
Fruit_type	Categorical	Varied	Popularity	Continuous	Varied	Pie chart	No clear relation	Not applicable
City	Categorical	Varied	Population	Continuous	Skewed right	Bar chart	Positive relation	Not applicable

User-Selected Variables. The most suitable plots for variables are predicted by DataAssist based on rule-based systems and SVM models. The training dataset for the SVM model is a curated dataset based on hundreds of Kaggle notebooks, which can in turn be broken down into records of how users analyzed their datasets and the datasets themselves. The information included in our training dataset (Table 1) are Variable 1 Name (Age, Income), Variable 1 Type (Categorical, Continuous, Ordinal), Distribution of Variable 1 (Normal, Skewed left), Variable 2 Name (Salary, Gender), Variable 2 Type (Categorical, Continuous, Ordinal), Distribution of Variable 2 (Normal, Skewed right), Variables Relation (Positive, Negative, Zero), Correlation Coefficient Threshold, Covariance threshold (Positive), Chi-square/ANOVA P-value Theoretical (High, Low), Mode/Cause of Relationship and the target variable plot type. As an output, plots that are most appropriate for the variable will be determined by our SVM model based on the type (categorical vs numerical) and distribution of the variable. The resulting plots, histograms, scatter plots, bar plots, box plots, violin plots, and/or alluvial plots, are standard procedures of EDA and will facilitate data quality control, model choice, and feature selection. In addition, the relationship between variables will be presented as heatmaps or cluster plots through analyses such as covariance/correlation analysis, hierarchical clustering and K means clustering.

Automatic Feature Selection. At a holistic level, DataAssist performs association rule mining on the dataset via ML algorithms such as Apriori [10] and FP-growth [11] to discover related variables, for example, frequent itemset conforming to certain transaction rules. Moreover, DataAssist uses random forests and decision trees to rank important features.

2.2 Data Cleaning

Subsequently, DataAssist offers the data preprocessing framework for data cleaning and preparation. We formulate the problem of data cleaning as follows: Given a set of constraints L, a structured dataset D where $D \nvDash L$, and a dataset reflecting the desired data distribution D_I, identify and repair erroneous records such that the repair dataset $D_r \vDash L$ and the distance between D_r and D_I is minimized. D is characterized by a set of attributes $A = \{A_1, A_2, ..., A_N\}$ which are essentially columns of the dataset. D can also be represented as a set of rows, or vectors $V = \{v_1, v_2, ..., v_M\}$ where each v_i consists of a set of cells denoted as $c[v_i] = \{A_i[v_i]\}$. $v_i[A_j]$ the j-th cell of i-th vector for attribute $A_i \in A$. Users are allowed to select and specify the following features:

Missing Values. Missing values are detected by empty entries or infinity or NaNs. To avoid causing errors or overflow in ML models, users may choose to remove that row from the dataset if the number of missing values in a row exceeds a certain threshold. Additionally, the SVM model underlying DataAssist can decide the most suitable missing imputation technique by analyzing the variable in the data and the broader attribute distribution. Specifically, statistics like mean and median can be used to impute numerical variables and mode for categorical variables. Additionally, for both types of variables, the SVM model may recommend imputing the missing value through Multivariate Imputation by Chained Equations [12], which iteratively runs ML models on other available features to predict the missing record.

Outliers. DataAssist offers a variety of anomaly detection algorithms $C(.)$: $A \rightarrow C \subseteq D$ for users to choose from. Univariate outliers $v_i[A_j]$ are detected by algorithms, such as IQR, Density-based spatial clustering of applications with noise (DBSCAN) [13] and isolation forest [14] if $v_i[A_j] \notin \{c_1, c_2 ... c_K | c_i \notin C\}$; multivariate outliers $v_i[A_j, A_k, A_l]$ are detected by local outlier factor. Upon completion of the pipeline, users are prompted to visualize the outliers highlighted by a different color from the rest of the dataset (Fig. 1), and remove them by clicking on the outliers. Alternatively, users may choose to replace outliers with another value through winsorization.

Duplicates and Inconsistencies. Duplicates refer to records referring to the same real-world entity, whereas inconsistencies are the same entity with different representations. DataAssist employs a learnable similarity function $S(.) : V \rightarrow R$ in the form of pairwise supervision which consists of object pairs known to be similar or dissimilar [15]. $v_i[A_j]$ is substituted by $v_h[A_j]$ if $S(v_i[A_j], v_h[A_j]) \leq r_1$, a threshold for dissimilarity within attribute A_j. Alternatively, v_j is removed for $S(v_i, v_h) \leq r_n$, a threshold that sets the minimum dissimilarity across attributes A_j's.

2.3 Data Preprocessing

Lastly, DataAssist offers preprocessing options such as data transformation to allow variables to conform to distributional assumptions required by ML models. We leveraged Gradient Boosting Machines like XGBoost [16] to train a multi-class multi-label classification task to predict preprocessing steps as target variables. The training dataset (Table 2), similar to the one used in the previous model for predicting EDA, consists of information derived from the Kaggle dataset, including the original distribution (Categorical, or Continuous), Missing

Table 2. Example training dataset for the XGBoost model predicting preprocessing

Variable name	Original distribution	Missing value handling	Transformation	Feature scaling	New distribution	Required analysis type	Consider outlier treatment	Variable nature	Scale of measurement
City	Categorical (100+ categories)	None	Frequency encoding	None	Varied (fewer categories)	Any	No	Predictor	Nominal
Gender	Categorical ('Male', 'Female')	None	Label encoding	None	Binary ('0', '1')	Any	No	Predictor	Nominal
Income	Continuous (wide-range)	Median imputation	None	Min-Max scaling	Continuous	Yes	Target	Ratio	High
Age	Continuous	None	Discretization (age groups)	None	Categorical	Classification	No	Predictor	Ratio
Product_type	Categorical (many categories)	Mode imputation	One-hot encoding	None	Multiple-binary	Any	No	Predictor	Nominal
Experience	Left skewed	Mean imputation	Square transformation	None	Approximate normal	Any	Yes	Predictor	Ratio
Salary	Very wide range	None	None	Min-Max normalization	Range between 0–1	Regression	Yes	Target	Ratio
Height	Slightly left skewed	None	None	Z-score standardization	Normal distribution	Any	Yes	Predictor	Ratio

Value Handling (Mean, Median, Mode Imputation), Transformation (Frequency Encoding, Label Encoding, One-Hot encoding) , Feature Scaling (Min-Max Scaling, Z-Score standardization), New distribution, the variable type after transformation (Categorical, Binary), the type of analysis being performed (Classification, Regression), the target variable, the predictor variables, the scale of measurement (Nominal, Ordinal, Ratio), the Variable cardinality (Low, Med, High) and the Data Scope (Transactional, Demographic). Moreover, DataAssist takes advantage of the natural language processing (NLP) model BERT [17] to embed attributes A of the same variable type as vectors in high dimensional spaces and uses different metrics of geometric distance between A_i's to determine their semantic similarity. If $dist(A_i, A_j) \leq d_n$, the distance at which similarity between A_i and A_j is calculated to be high by DataAssist, the program will perform the same data manipulation on these attributes.

One-Hot/Label Encoding. It is important for algorithms to distinguish between the ordinal categorical variables for which attributes of significance correspond to the order of numbers from the nominal categorical variables for which orders do not matter. Label encoding uses, as a default option, integers to represent ordinal categorical variables. In contrast, for nominal categorical variables, a pre-processing step that converts an A_c into a binary vector A'_c whose entry is an indicator for whether a particular category is present ($A'_c = I(A_c = k) \forall k \in domain(A_c)$) is used to integrate them into ML models.

Standardization/Normalization. One of the most common ways of standardization is to center the numeric variable A_n around 0 mean and scale it to unit variance such that $A_n \sim P(\mu = 0, \sigma = 1)$, where P is any probability distribution, but most commonly a normal distribution, also known as z score normalization. Alternatively, it is possible to perform min-max normalization such that $A_n \in [0, 1] | [-1, 1]$. This is beneficial for ML algorithms that are sen-

sitive to the scale of the features, such as support vector machines (SVMs) and k-nearest neighbors (KNN).
Power Transformation. This is another common pre-processing for numeric variables by applying power functions A_n^λ where λ is the power parameter, in order to make the distribution closer to normal distribution. DataAssist uses Box-cox transformation to determine the value of λ through the likelihood function and goodness-of-fit tests. Some special cases of λ include square root transformation, log transformation and log10 transformation.

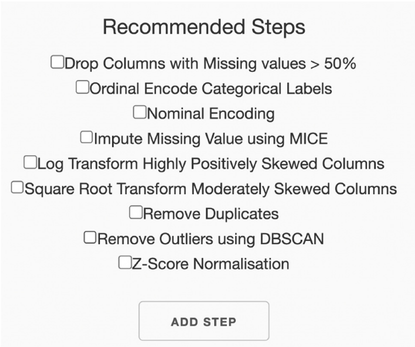

Fig. 2. Steps for data cleaning and preparation

3 The Interface

This section uses two datasets as an example to demonstrate the components of DataAssist. The user first navigates to the "Gather Data" page to preview and upload the dataset. Once uploaded, DataAssist will automatically perform the pipeline on the dataset.

3.1 House Prices

This dataset [18] contains 1460 records of residential homes in Ames, Iowa sold between 2006 and 2010. Each record has 80 attributes including sale price, overall condition of the house, type of dwelling, and proximity to various conditions. The regression task is to predict the sale price using all or some of the attributes.
As a part of EDA, DataAssist ranks the importance of the feature using random forest, shown in Fig. 3. The top three important features associated with Sale Price are Unfinished square feet of basement area (BsmtUnfSF), First Floor square feet (1stFlrSF), and Above grade (ground) living area square feet (GrLivArea). Users are invited to inspect the distribution of specific higher-ranked important variables or select features based on external information based on

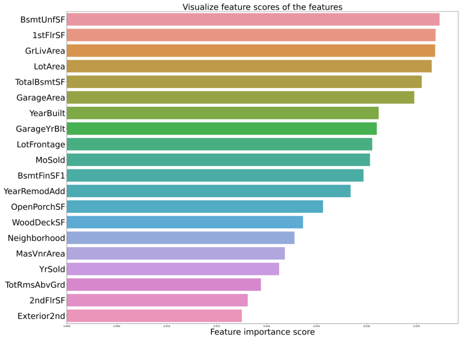

Fig. 3. Ranking of important features associated with sale price

domain knowledge. DataAssist then recommends steps for data cleaning and preparation, shown in Fig. 2. Briefly, 38 records are removed from the dataset due to missing values. Multiple attributes, such as zoning classification of the sale (MSZoning), physical locations within Ames city (Neighborhood), and proximity to various conditions (Condition1, Condition2) are one-hot encoded. Categorial variables such as the overall condition of the house (OverallCond), the overall material and finish of the house (OverallQual), and Heating quality and condition (HeatingQC) are nominal encoded. The outliers shown in Fig. 1 are removed from the dataset. The numerical values are transformed by applying power functions, and subsequently normalized.

3.2 Air Quality Data in India

This dataset [19] contains 29531 records of daily air quality data across 26 cities in India. Each record has 15 attributes including datetime of the collection, PM 2.5, PM10, and concentration of various pollution gas. The regression or classification task is to rate the air quality index given the attributes.

As a part of EDA, DataAssist generates a heatmap with columns ordered by hierarchical clustering to explore the relationships between variables. Again, users are invited to inspect the distribution of variables of interest. If, for example, the daily PM2.5 level and City are selected, DataAssist will generate a barplot for the number of records corresponding to each city and a histogram for PM2.5. To demonstrate the relationship between two selected variables, DataAssist produces a violin plot showcasing the daily PM2.5 level for each city in India. For data cleaning and preprocessing, DataAssist recommends steps similar to ones shown in Fig. 2. Briefly, 3582 records are removed from the dataset due to missing values. The only two categorical variables, City and AQI_Bucket, are one-hot encoded and nominal encoded, respectively. The numerical values for PM2.5 and PM10 values, as well as various pollution gas concentrations, are standardized and normalized.

4 Comparative Analysis

To demonstrate the effectiveness and efficiency of DataAssist, we conducted a comparative analysis with other existing technologies. We used several datasets and applied the same data cleaning and preparation steps using both manual methods and DataAssist. The results are summarized in Table 3.

Table 3. Comparative analysis of time efficiency and data quality

Dataset	Records	Variables	Outliers	Manual time	Data-assist time
Bank marketing	45,211	17	200	8 h	3 h
Credit card fraud	284,807	31	1,000	12 h	4.5 h
Loan prediction	614	13	50	10 h	4 h
Customer segmentation	5,000	5	100	9 h	3.5 h
Sales forecasting	1,000	4	30	11 h	4.5 h

As shown in the Table 3, DataAssist significantly reduces the time required for data cleaning and preprocessing compared to manual methods, regardless of the number of preprocessing steps required. On average, DataAssist reduces the time spent on data cleansing and preparation by over 50%. Furthermore, the quality of the cleaned and prepared data was also superior, leading to improved performance in downstream analysis. This comparative analysis clearly demonstrates the advantages of DataAssist and its potential to significantly enhance the data analysis process.

5 Discussion and Future Directions

The development of DataAssist addresses a significant gap in the current ML landscape by providing a comprehensive tool for data cleaning and preparation. By integrating exploratory analysis, data cleaning, and pre-processing into a single platform, DataAssist not only streamlines the data analysis process but also enhances the quality of the dataset, leading to improved performance of ML models.

However, there are still areas for improvement and potential future directions for this research. For instance, the current version of DataAssist primarily focuses on structured datasets. Future versions could be extended to handle unstructured data, such as text, images, and videos. Additionally, the integration of DataAssist with other autoML tools could further automate the entire data analysis process, from data collection and cleaning to model training and deployment.

Another potential direction is the incorporation of more advanced ML models for predicting the most suitable methods for EDA and preprocessing. While the current version of DataAssist uses SVM and XGBoost models, future versions could explore the use of deep learning models, which may provide more accurate predictions.

6 Conclusion

In this paper, we presented DataAssist, a comprehensive tool for automated data cleaning and preparation. By leveraging ML models, DataAssist provides a streamlined process for exploratory data analysis, data cleaning, and preprocessing. Our comparative analysis demonstrated the superior performance and efficiency of DataAssist compared to other existing technologies. The development of DataAssist addresses a significant gap in the current ML landscape and has the potential to significantly enhance the data analysis process in various fields. Future research will focus on extending DataAssist to handle unstructured data and integrating it with other autoML tools to further automate the entire data analysis process. This research contributes to the field by providing a comprehensive and efficient tool for data cleaning and preparation, enhancing the quality of datasets and improving the performance of ML models.

References

1. Feurer, M., Eggensperger, K., Falkner, S., Lindaueur, M., Hutter, F.: Auto-Sklearn 2.0: hands-free AutoML via meta-learning. J. Mach. Learn. Res. **23**(261), 1–61 (2022). Available: https://doi.org/10.48550/arXiv.2007.04074
2. Le, T.T., Fu, W., Moore, J.H.: Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics **36**(1), 250–256 (2020)
3. LeDell, E., Poirier, S.: H_2O AutoML: scalable automatic machine learning. In: 7th ICML Workshop on Automated Machine Learning (2020)
4. Krishnan, S., Wang, J., Wu, E., Franklin, M.J., Goldberg, K.: ActiveClean: interactive data cleaning while learning (2016). arXiv preprint arXiv:1601.03797
5. Krishnan, S., Franklin, M.J., Goldberg, K., Wu, E.: BoostClean: automated error detection and repair for machine learning (2017). arXiv preprint arXiv:1711.01299
6. Yakout, M., Berti-Équille, L., Elmagarmid, A.K.: Don't be SCAREd: use SCalable automatic REpairing with maximal likelihood and bounded changes. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 193–204 (2013)
7. Yakout, M., Elmagarmid, A.K., Neville, J., Ouzzani, M., Ilyas, I.F.: Guided data repair. Proc. VLDB Endowment **4**(5), 290–301 (2011)
8. Rekatsinas, T., Chu, X., Ilyas, I.F., Ré, C.: HoloClean: holistic data repairs with probabilistic inference. Proc. VLDB Endowment **10**(11), 1190–1201 (2017)
9. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 269–278 (2002)
10. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), pp. 487–499 (1994). https://doi.org/10.1016/B978-1-55860-694-6.50041-9
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 1–12 (2000). https://doi.org/10.1145/342009.335372
12. van Buuren, S., Groothuis-Oudshoorn, K.: MICE: multivariate imputation by chained equations in R. J. Stat. Softw. **45**(3), 1–67 (2011). https://doi.org/10.18637/jss.v045.i03

13. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), pp. 226–231 (1996). https://doi.org/10.1145/299844.299854
14. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17
15. Bilenko, M.Y.: Learnable similarity functions and their application to record linkage and clustering. Ph.D. dissertation, Department of Computer Science, The University of Texas at Austin, Austin, TX (2006)
16. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 785–794 (2016)
17. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, pp. 4171–4186 (2019)
18. House Prices—Advanced Regression Techniques. Kaggle (2021) [Online]. Available: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data
19. Rao, R.: Air Quality Data in India. Kaggle (2021) [Online]. Available: https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india