Design, Development, and Use of an Exploratory Data Analysis Tool

Tenzin **DOLECK**
Simon Fraser University Burnaby, BC, Canada
tdoleck@sfu.ca

Pedram **AGAND**
Simon Fraser University Burnaby, BC, Canada
pagand@sfu.ca

Dylan **PIRROTTA**
Simon Fraser University Burnaby, BC, Canada
dpirrott@sfu.ca

Data science is a rapidly evolving field and is assuming an increasing role in both academia and industry. As we navigate this landscape, it becomes clear that effective tools are essential for optimizing data science workflows. One such tool holds great potential: Exploratory Data Analysis (EDA) tool. In this paper, we present the design, development, and use of an EDA tool integrated into the DaTu system—a data science educational platform designed to provide personalized guidance for students learning data science concepts. By integrating an EDA tool into the DaTu system, we aim to enhance student learning experiences and promote effective exploration of datasets. This research contributes to the ongoing efforts to develop innovative solutions that support productive data science education and research.

Keywords: Exploratory Data Analysis (EDA); Visualizations; Data Science; Machine Learning; Artificial Intelligence; Programming

Introduction

Technological advancements typically support the development of new skills, rendering some obsolete in the process (Ra et al., 2019). In recent years, the proliferation of data capture and storage capabilities (Power, 2016) has led to an unprecedented surge in data volume (Sarker, 2021). In response, there has been a concerted effort to develop sophisticated computational tools and techniques to help generate insights from complex *big* datasets (Acciarini et al., 2023). This backdrop has significantly elevated the importance of data science (Crisan et al., 2021)—an "interdisciplinary field that combines statistics, data mining, machine learning, and analytics to understand and explain how we can generate analytical insights and prediction models from structured and unstructured big data" (George et al., 2016, p.1493)—for both organizations and individuals across work fields (Coners et al., 2015; Fernandes et al., 2023; Jahani et al., 2023).

The growing significance of data science across various disciplines and industries (Coners et al., 2015; Memarian & Doleck, 2023; Mike & Hazzan, 2023), including education (Ow-Yeong et al. 2023), makes it imperative for preparing individuals with the requisite skills to succeed in a data-driven world (Coners et al., 2015; Zhang et al., 2022). In response, numerous learning environments have been developed to facilitate the acquisition of data science competencies (McKinney et al., 2024). However, a significant challenge within these learning environments is the limited availability of effective tools specifically tailored for exploratory data analysis (EDA) (Jiang et al., 2024)—a crucial aspect of data science that enables learners to develop a deeper understanding of datasets—to gain insights and identify patterns—through interactive and visual data exploration (Peng et al., 2021).

To address this challenge, we have developed a comprehensive learning and research platform called *DaTu* (Doleck et al, 2024), specifically designed for Data Science education and research. By integrating an EDA tool within the *DaTu* system, our objective is to equip users with sophisticated data exploration capabilities that facilitate efficient analysis of complex datasets. This, in turn, will enable them to extract valuable insights, thereby enriching their learning experience and fostering a deeper understanding of data-driven concepts.

Background

Exploratory data analysis (EDA)

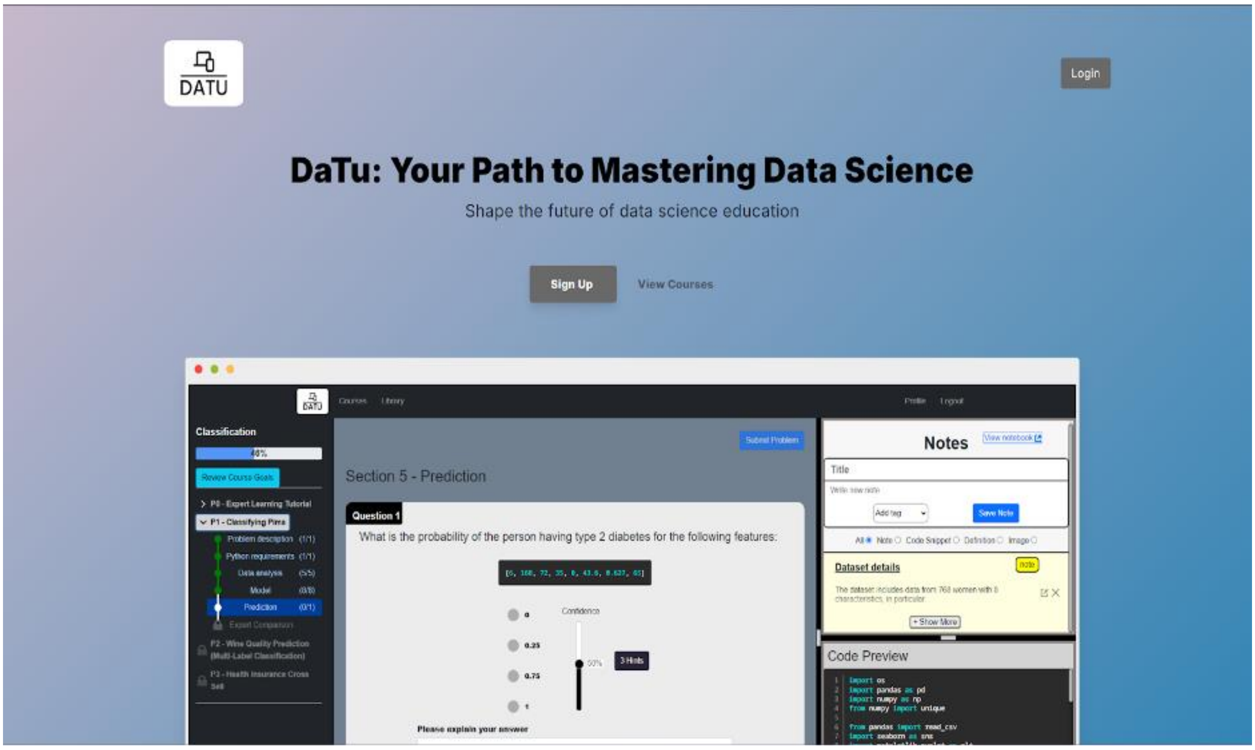
Real-world data is often messy and incomplete (Liu & Demosthenes, 2022), making exploratory data analysis (EDA) an important process for developing a deeper understanding of data (Batch & Elmqvist, 2018; Jebb et al., 2017; Pearson, 2018; Peng et al. 2021). The EDA process requires careful attention to three key areas: data preprocessing, visualization, and modeling (Batch & Elmqvist, 2018; Mike & Hazzan, 2023; Peng et al., 2021; Wongsuphasawat, 2019). EDA begins with careful data preprocessing. Cleaning, transforming, and preparing the data for analysis, helps create a base upon which to build further analysis (Jansen et al., 2023). Visualization provides additional insights in data. By using graphical summaries such as histograms and scatter plots, one can gain a deeper understanding of the distribution of variables and identify patterns (Zietz et al., 2018). Data transformations (e.g., normalization) offer additional channels to prepare the data for subsequent modeling processes (Kamalov et al., 2023).

However, despite its significance within the data analysis workflow, EDA encounters several challenges and limitations. For example, selecting the optimal EDA method for a specific dataset can be challenging due to diverse data types and objectives (Wongsuphasawat, 2019). Furthermore, the absence of definitive guidelines for performing EDA (Zietz et al., 2018) and the focus of EDA tools on plotting capabilities without sufficiently catering to EDA needs (Peng et al., 2021) pose significant obstacles. To address these challenges, it is essential that EDA methods and tools evolve and adapt to maintain their relevance and effectiveness (Ovando-Fuentealba et al., 2024).

Background of the *DaTu* project

The *DaTu* (Figure 1) Intelligent Tutoring System provides a learning and research environment for data science. The system's core components include an emulation of a real-world data science programming setting, which enables learners to engage with authentic exercises that simulate the workflow of data science projects.

Figure 1
DaTu Learning System



DaTu: Your Path to Mastering Data Science

Shape the future of data science education

What is the probability of the person having type 2 diabetes for the following features:

```
[6, 168, 72, 35, 0, 43.6, 0.627, 65]
```

The dataset includes data from 768 women with 8 characteristics, in particular

```
import os
import pandas as pd
import numpy as np
from numpy import unique

from pandas import read_csv
import seaborn as sns
```

In *DaTu*, the problem space serves as a hub for available tasks. As a learner clicks on each task, its description and associated questions unfold in the central pane. Learners can access supplementary information through the Library, while a notepad can be used to record code snippets, definitions, and other pertinent information. A code preview area displays system-generated code for learners' consideration. Three bottom tabs provide access to a code editor, an EDA tool, and an AI-chatbot.

The integration of the EDA tool within *DaTu* offers several advantages. The user-friendly interface empowers learners to delve into their data with ease, fostering hands-on exploration. The provision of authentic datasets enables learners to apply their EDA skills in real-world scenarios, enhancing practical understanding. The EDA tool caters to students' diverse needs by overcoming limitations in existing tools, offering a comprehensive toolkit for learning. Lastly, the capability to save and reuse EDA artifacts encourages interactive analysis, fostering iterative learning processes.

Purpose of the Study

To address the challenge of facilitating effective EDA, we developed an EDA tool within the *DaTu* system that provides users with a user-friendly interface for exploring datasets and visualizing complex data patterns and relationships. This paper describes the features of the developed EDA tool, its implementation within the *DaTu* system, and our experience using it with a group of students.

Method

Our development of the EDA tool was informed by a systematic methodology that integrated theoretical foundations with practical application. The following steps were conducted: (1) literature search relating to EDA; (2) development of the prototype; (3) user study with the tool.

Literature Search

Our comprehensive literature search across multiple databases (Web of Science, Scopus, and Google Scholar) for Exploratory Data Analysis (EDA) revealed several essential steps that underpin effective EDA practices: understanding variables, cleaning the dataset, selecting the right analytical method, feature engineering, and visualizing and analyzing results. We then used this knowledge to design and develop an EDA tool that supports these key elements.

Design and Development of the Tool

In the following section, we will cover the technical details for the structure of the EDA toolbox in *DaTu*.

Architecture

In this section, we will explore the technical underpinnings of our EDA tool, highlighting key design decisions and implementation strategies that have been instrumental in shaping its functionality. We selected Streamlit as the foundation for our EDA application due to its user-friendly interface and ease of integration with various libraries. To augment the functionality of our platform, we incorporated components.v1 and pandas_profiling for additional features.

To optimize overall performance and minimize server queries, we employed caching using Streamlit's @st.cache decorator. This feature is applied to essential functions such as load_image(), show_html() (for Pandas Profiling), and read_csv(). By employing caching, we significantly reduce the number of queries to the server, thereby enhancing overall performance.

For the side menu, we utilize st.sidebar.slider function to add an additional layer to the main app. We created user_input_features() function to handle user inputs and models, which returns features and classifier objects. A summary of the different functions used in our implementation can be found in Table 1.

Table 1
List of functions in the EDA with the I/O and the descriptions

Function name	Inputs	Output	Description
show_html	HtmlFile	Read HTML	Cached read HTML
read_csv	DATA_URL	Read CSV	Cached Read CSV
user_input_features	Features	features, clfs	Create range of features and create clfs as model
compute_predition	clfs, features, labels, df_test	predictions	Append classifier models for prediction

show_scatter_plot	selected_species_df	void	Create scatter plot of the dataframe
select_species	source_df	selected_species_df	A sub dataframe having data for the selected species
show_histogram_plot	selected_species_df	void	Create histogram plot of selected features
_handle_missing	features, labels	new_features, new_labels	Handle missing values by imputing with mean, drop entry or fill with 0
handle_io	source_df	features, labels	Check if the data is numeric, otherwise map category, check null
show_machine_learning_model	source_df	void	Display the performance of a trained ML Algorithm, check balance (apply sampling), compute accuracy

In our predictive modeling process, we employ the `compute_prediction` function to drive the prediction phase. This function plays a pivotal role in implementing machine learning algorithms derived from scikit-learn.

To address various aspects of data exploration, we have engineered a series of functions to facilitate distinct operations: `show_scatter_plot`, `select_species`, and `show_histogram_plot`. `show_scatter_plot` is deployed for generating scatter plots. `select_species` is instrumental in exhibiting the count of unique elements within the output column, thereby enhancing understanding of the data's compositional properties. `show_histogram_plot` serves to construct histograms, providing valuable insights into the distribution and frequency of specific data features.

Upon processing data derived from either the pre-existing repository or imported datasets, potential occurrences of incomplete information may be present. To mitigate potential complications arising from such missing entries, the training/testing datasets are first passed through the `_handle_missing` function. This critical preprocessing step aims to eliminate any existing missing values. We offer three approaches for managing missing data: impute with mean, drop the entry, or replace with zero. The implementation of this approach is contingent upon the existence of missing values within the dataset, and the associated option is displayed accordingly.

To detect missing values or non-numeric values in the dataset, we have engineered a function titled `handle_io`. This crucial component serves the purpose of detecting disparate data types, particularly non-numeric values or missing entries, within the input dataset before it is submitted to the machine learning model. The `handle_io` function screens each datum, and in the case of categorical features, tries to establish a mapping based on the number of distinct categories. Subsequently, these categories are replaced with corresponding numeric values. It is essential to emphasize that this functionality does not extend to continuous string data.

The `show_machine_learning_model` function assumes a pivotal role in facilitating model training. The process initiates with the determination and setting of an optimal train-test ratio. Preceding this stage, the input dataset undergoes preprocessing through the `handle_io` function, which ensures the removal or handling of any non-numeric values and missing entries within the dataset. Subsequently, additional options are presented to users for managing imbalanced datasets via various sampling methods. Following these preparatory steps, the `show_machine_learning_model` function proceeds with model training. Evaluation metrics, including the ROC AUC score, precision score, recall score, and F1 score, are computed and displayed to users. For visualization purposes, this function also incorporates the generation of a Receiver Operating Characteristic (ROC) curve through invocation of the `roc_curve` method.

Our methodology for conducting comprehensive data analysis is structured around four primary tasks (as detailed in Table 2), each distinctly categorized within specific sections to ensure clarity and organization. These tasks are: Data Review for data exploration, Visualization for plotting the results, Classification Modeling for training a classifier, and Classification Prediction for predicting with the trained model. By structuring our methodology around these four primary tasks, we aim to provide users with a clear and organized approach to conducting comprehensive data analysis.

Table 2
List of tasks in the EDA with the utility and the descriptions

Task	Utility	Description
'Data review'	ProfileReport, EDA	Review data, produce report
'Visualization'	Histogram, Pyplot, Scatter	Visualize data
'Classification modeling'	Sklearn, imblearn	Create the classification model, handle imbalance, show measurements
'Classification prediction'	Sklearn, imblearn	Create the new datapoint from same range, learn model, predict

Deployment

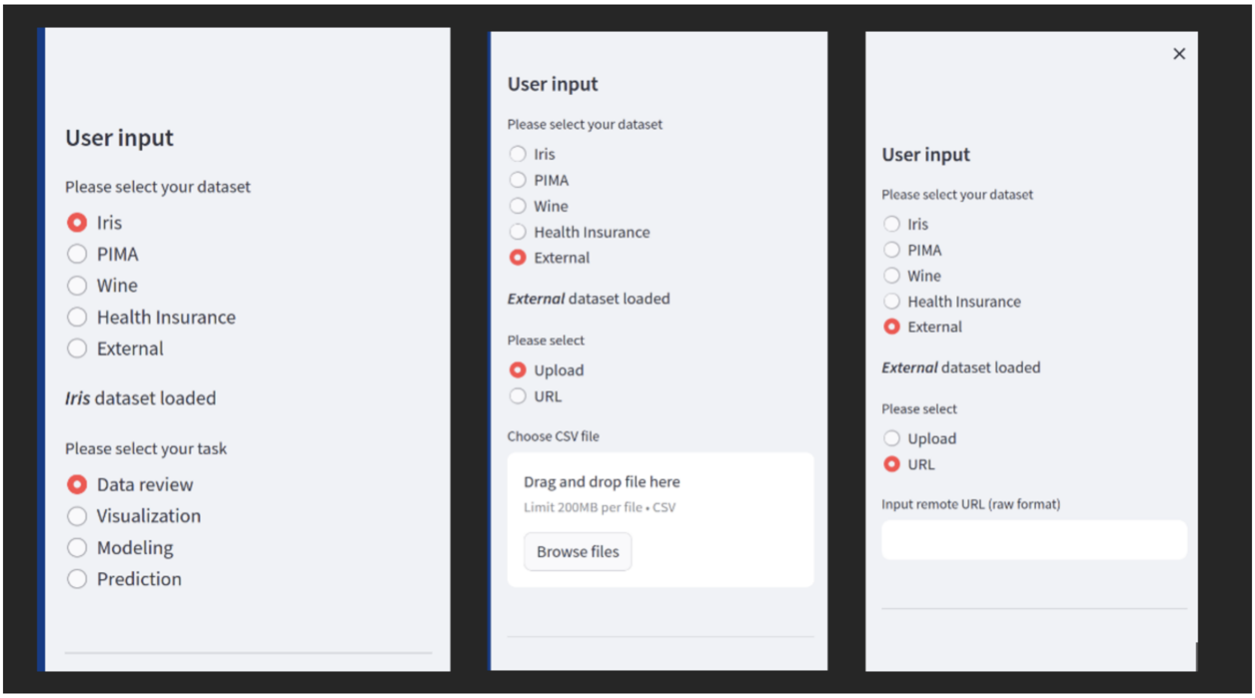
To deploy Streamlit with Nginx, several steps are involved. These include configuring Nginx as a reverse proxy for Streamlit, setting up a SystemD service for Streamlit, and managing Nginx configurations.

Description of the EDA Tool

In this section, we will cover the various components of the EDA tool.

The EDA tool consists of various components that facilitate data exploration and analysis. The sidebar menu (Figure 2) provides users with access to preloaded datasets, including Iris, PIMA, Wine, and Health Insurance. Users can also upload a numeric CSV file or paste the URL for a raw CSV file from the internet.

Figure 2
EDA sidebar: selecting or uploading the data



User input

Please select your dataset

● Iris
○ PIMA
○ Wine
○ Health Insurance
○ External

Iris dataset loaded

Please select your task

● Data review
○ Visualization
○ Modeling
○ Prediction

User input

Please select your dataset

○ Iris
○ PIMA
○ Wine
○ Health Insurance
● External

External dataset loaded

Please select

● Upload
○ URL

Choose CSV file

Drag and drop file here
Limit 200MB per file • CSV

User input

Please select your dataset

○ Iris
○ PIMA
○ Wine
○ Health Insurance
● External

External dataset loaded

Please select

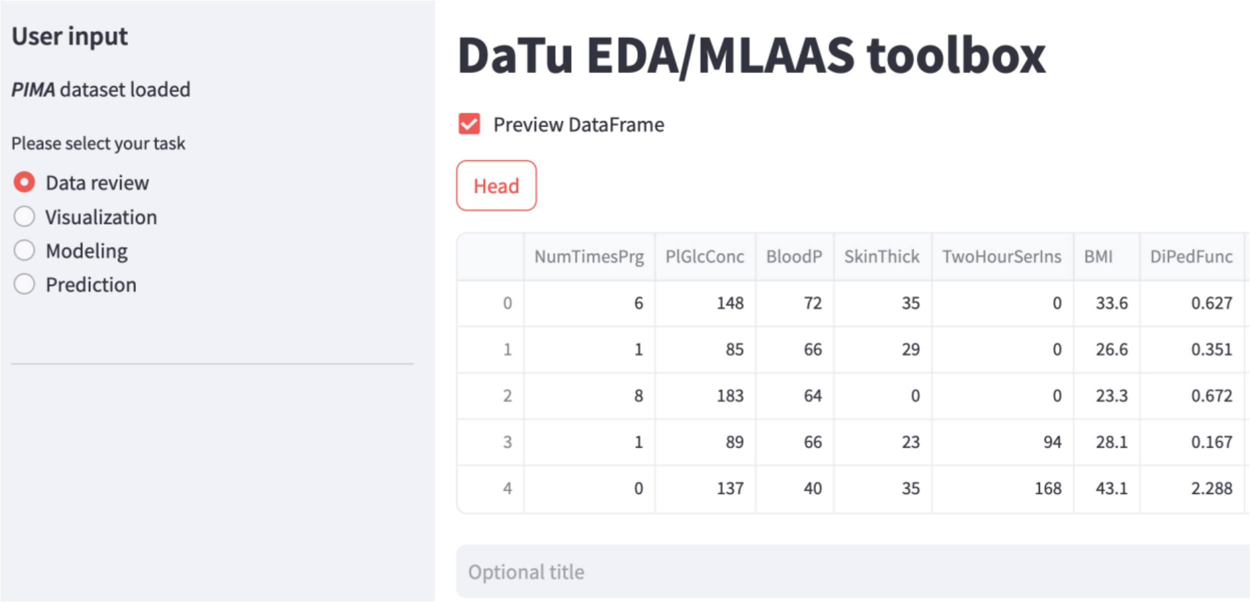
○ Upload
● URL

Input remote URL (raw format)

Upon selecting a dataset, users are presented with four tasks: Data Review, Visualization, Modeling, and Prediction. Each task offers unique functionalities that aid in understanding and analyzing the selected dataset.

The Data Review (Figure 3) task serves as a preliminary stage for gaining insights into the dataset. This includes displaying the initial rows (head) and terminal rows (tail), generating a summary, and employing Pandas Profiling to reveal additional information such as correlations, missing values, distributions, duplicates, interactions, and more.

Figure 3
Data Review



User input

PIMA dataset loaded

Please select your task

● Data review
○ Visualization
○ Modeling
○ Prediction

DaTu EDA/MLAAS toolbox

☑ Preview DataFrame

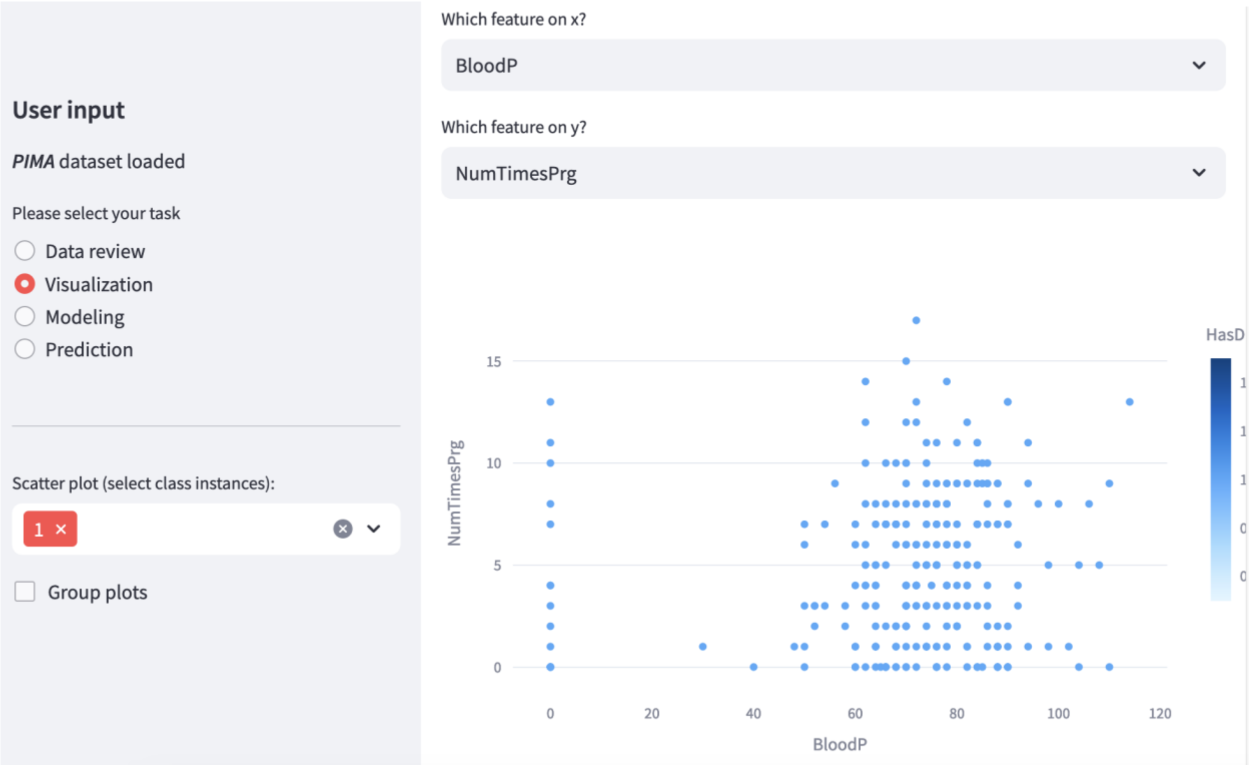
Head

	NumTimesPrg	PlGlcConc	BloodP	SkinThick	TwoHourSerIns	BMI	DiPedFunc
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288

Optional title

In the Visualization task (Figure 4), users are presented with additional options in the left sidebar. One option includes selecting scatter plots for specific classes, which displays histograms and scatter plots for selected features. Additionally, users can opt for group plots, consisting of Matplotlib for individual points (which may be time-consuming for large datasets), box plots, correlation plots, and bar plots. Users are afforded the convenience of saving figures, adjusting their sizes, and employing interactive features such as zooming, labeling, and highlighting to further investigate the data.

Figure 4
Visualization Task



User input

PIMA **dataset loaded**

Please select your task

○ Data review
● Visualization
○ Modeling
○ Prediction

Scatter plot (select class instances): 1 ×

□ Group plots

Which feature on x? BloodP

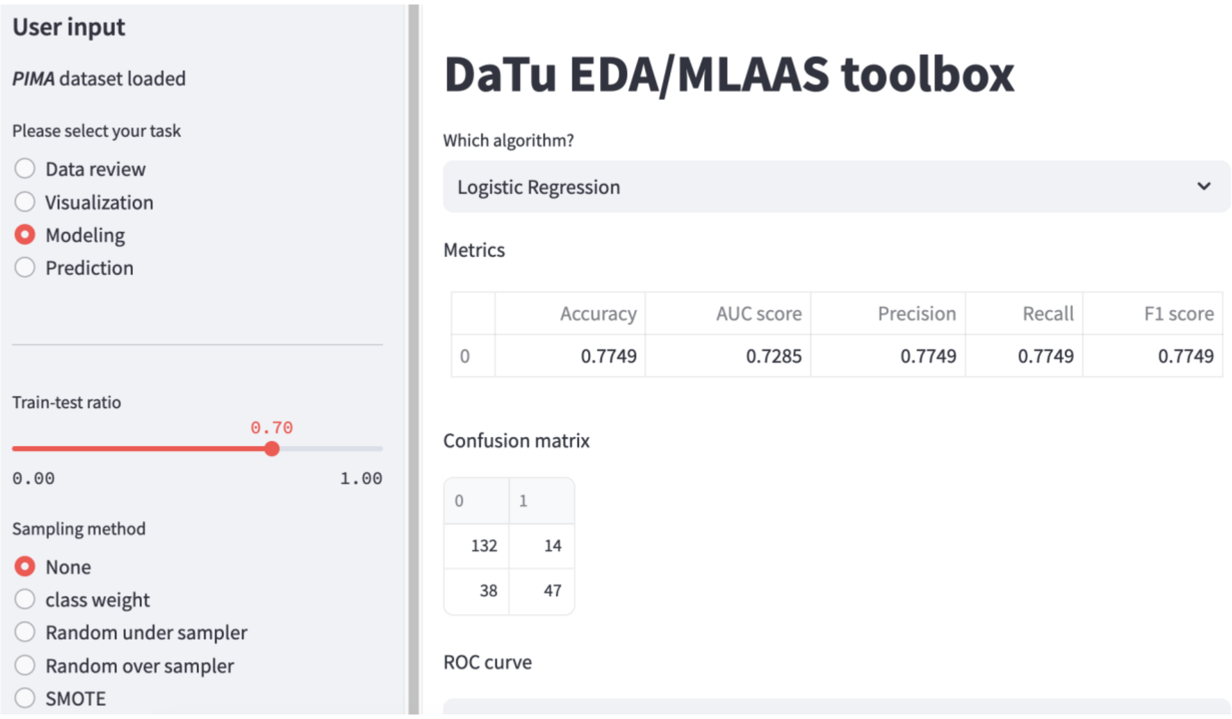
Which feature on y? NumTimesPrg

In the modeling section (Figure 5), on-the-fly learning is made available for different algorithms (logistic regression, random forest, decision tree, support vector machine, naive Bayes, K-nearest neighbors, and linear discriminant analysis). Upon choosing a modeling method from the selection provided, the application produces performance-related metrics such as Accuracy, AUC score, Precision, Recall, and the F1 score. Additionally, it generates the confusion matrix and ROC curve to provide a more comprehensive understanding of the model's efficiency.

The Streamlit application's modeling section grants users the flexibility to experiment with various machine learning models while concurrently providing them with tuning parameters within the sidebar. Users may adjust the train-test ratio, which is set at 0.7 by default, and select a sampling method (None, class weight, Random under sampler, Random over sampler, and SMOTE) for datasets exhibiting class imbalance. This flexibility empowers users to assess model performance and determine which algorithm yields the most promising results while optimizing hyperparameters for enhanced accuracy in crucial metrics.

In dealing with datasets containing missing values, the app identifies such instances and automatically introduces an option in the sidebar for users to manage them. Users may select from available alternatives such as impute with the mean, drop the entry, or replace with zero.

Figure 5
Modeling Task



User input

PIMA dataset loaded

Please select your task

○ Data review
○ Visualization
● Modeling
○ Prediction

Train-test ratio

0.70

0.00 1.00

Sampling method

● None
○ class weight
○ Random under sampler
○ Random over sampler
○ SMOTE

DaTu EDA/MLAAS toolbox

Which algorithm?

Logistic Regression

Metrics

	Accuracy	AUC score	Precision	Recall	F1 score
0	0.7749	0.7285	0.7749	0.7749	0.7749

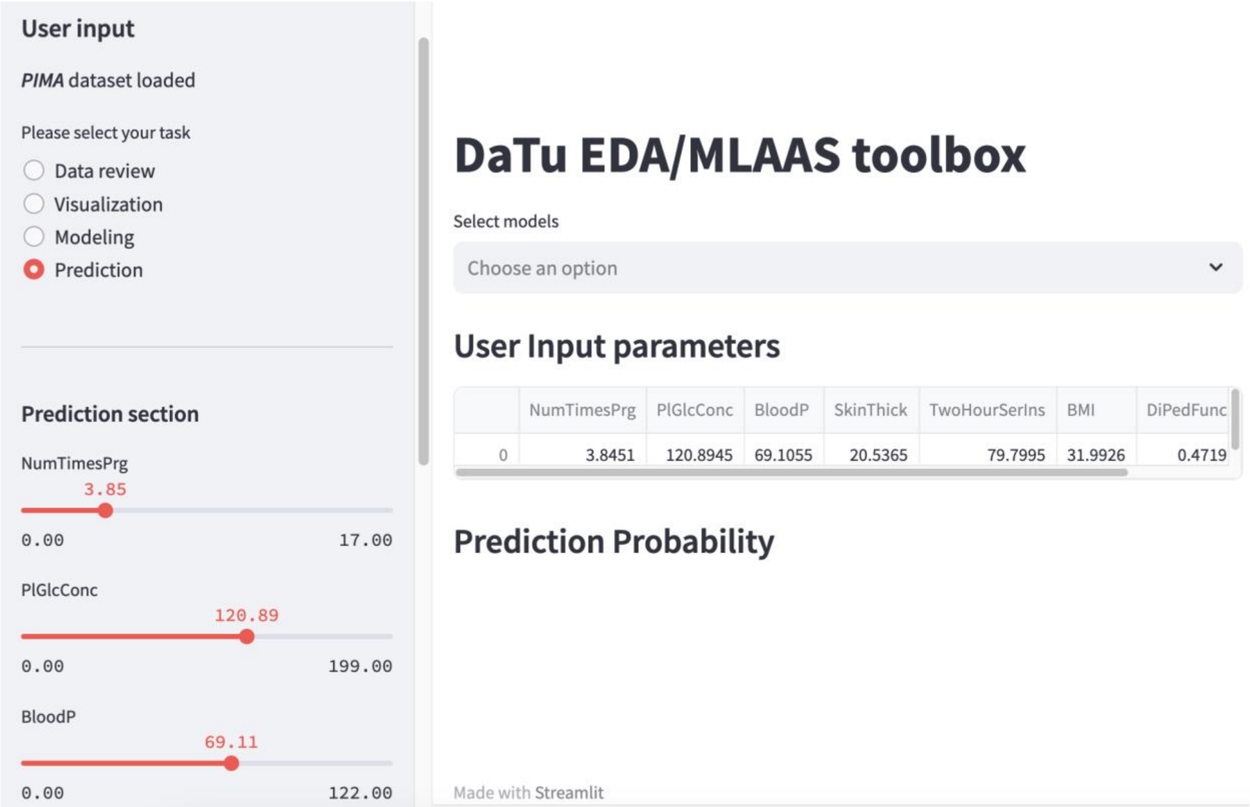
Confusion matrix

0	1
132	14
38	47

ROC curve

The Prediction task (Figure 6) empowers users to compare the performance of different machine learning models by presenting prediction results for specified feature values within the input range. This capability enables the application of ensemble techniques such as voting to determine a final classification or outcome.

Figure 6
Prediction Task



User input

PIMA dataset loaded

Please select your task

○ Data review
○ Visualization
○ Modeling
● Prediction

Prediction section

NumTimesPrg: 3.85 (0.00 – 17.00)

PlGlcConc: 120.89 (0.00 – 199.00)

BloodP: 69.11 (0.00 – 122.00)

DaTu EDA/MLAAS toolbox

Select models

Choose an option

User Input parameters

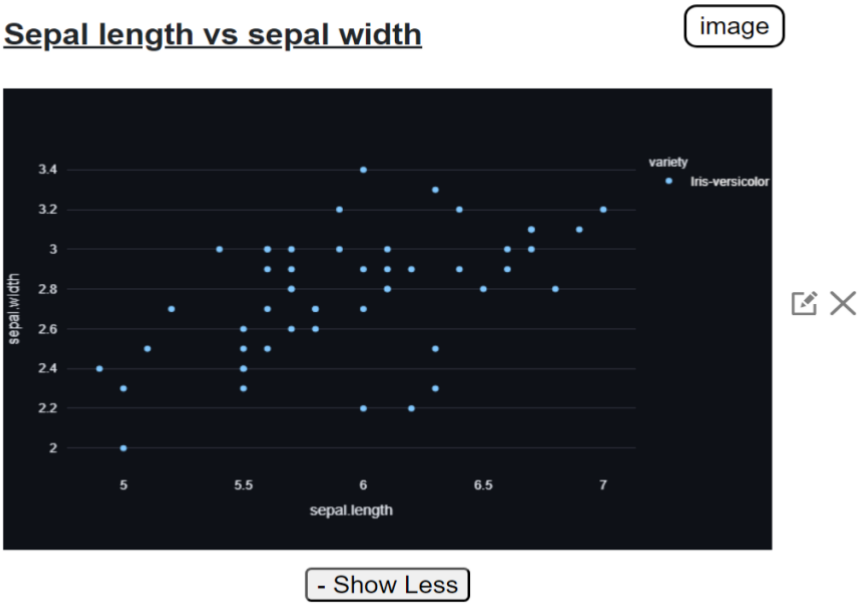
	NumTimesPrg	PlGlcConc	BloodP	SkinThick	TwoHourSerIns	BMI	DiPedFunc
0	3.8451	120.8945	69.1055	20.5365	79.7995	31.9926	0.4719

Prediction Probability

Made with Streamlit

In the EDA toolbox, users can save figures along with an optional title to their notes section; for an example, see Figure 7. This feature allows for seamless integration of visual representations into their learning materials or personal study resources.

Figure 7
Saving Images in Notebook



Sepal length vs sepal width

Streamlit is an intuitive web application framework for Python that empowers data scientists to create dynamic and distributable data applications. By integrating EDA into a Streamlit application, several benefits are realized. Firstly, collaboration is enhanced as team members can engage with the dataset in an uncomplicated manner through Streamlit's user-friendly interface.
Secondly, a Streamlit-driven EDA application simplifies documentation. The transparency gained from recording and reproducing data exploration steps ensures credibility in the data analysis processes. Lastly, Streamlit's real-time interactivity enables adaptability throughout the exploration process.

In summary, deploying EDA with Streamlit transforms the way data exploration is conducted. It renders the process more accessible and engaging, elevating the effectiveness of data science projects.

Use of the Tool: Data Collection

Before embarking on our study, we obtained approval from the Institutional Review Board for our protocol (# 30001688). Participation was voluntary and they were compensated $60. Eligible participants needed knowledge of Python and basic data science. Participants included 89 undergraduates (*N*=89; 68 males and 21 females), with an average age of 21.44 (*SD*=2.64), participating. Participants signed a consent form, completed a demographics questionnaire, and were assigned anonymous IDs. The study used a between-subjects design with participants randomly assigned to use *DaTu* with or without AI chatbot assistance. The study included pre-tests, task-solving, and post-assessment questionnaires on system usability. Participants were tasked with solving three data science problems (classification tasks). The first two problems were structured as multiple-choice-driven tasks focusing on binary and multi-class classification, respectively. In contrast, the third problem was a free-form programming assignment centered on binary classification. Students are provided with the freedom to leverage the EDA tool according to their individual preferences and requirements. This allowed us to examine how students used the EDA tool in various contexts.

We aimed to initially assess the effectiveness of the EDA tool by measuring the time students spent on specific tasks, such as data review and visualization, to gain insight into their engagement with the tool. Our analysis revealed that participants exhibited varied time allocations when using the EDA tool across the three data science programming problems. We calculated the total time spent on each of the four primary tasks— Data Review, Visualization, Modeling, and Prediction— to analyze the distribution of time dedicated to each task in relation to the overall time spent using the EDA toolbox. We found that for all three problems, the majority of the time was devoted to Data Review and Prediction tasks (see Table 3). This suggests that participants primarily focused on understanding the dataset and making predictions rather than extensively exploring visualizations or developing models.

Table 3
Time dedicated to each task in relation to the overall time spent using the EDA toolbox

	Data Review	Visualization	Modeling	Prediction
Problem 1	35.79%	10.77%	9.81%	43.63%
Problem 2	55.44%	6.87%	2.73%	34.96%
Problem 3	70.28%	0.38%	0.00%	29.34%

Furthermore, we calculated the percentage of the total time spent on the EDA toolbox relative to the total problem-solving time for each of the three data science programming problems. Our results indicate that for the first two problems, participants on average devoted 20% of their total problem-solving time to using the EDA tool. This suggests that the EDA tool played an important role in the problem-solving process for these tasks. In contrast, participants spent only 3% of their total time on the EDA tool while solving the third problem. This finding highlights the varying importance of the EDA tool in different data science tasks and underscores the need to consider the specific problem context when allocating time for its use. For problem 3, participants may have found it more challenging to apply the EDA tool effectively in this free-form setting, leading them to rely on other methods that better suited the task requirements.

Overall, our study provides valuable insights into the role of the EDA tool in data science problem-solving. It emphasizes that while the EDA tool is applicable in structured classification tasks, its effectiveness may vary depending on the specific problem characteristics and the level of data complexity. By understanding these findings, students can optimize their time allocation and use the EDA tool effectively to enhance their problem-solving skills.

Discussion

As the field of data science continues to evolve at a rapid pace (Fernandes et al., 2023; Jahani et al., 2023; Mike & Hazzan, 2023), it becomes increasingly clear that developing robust tools for exploratory data analysis is essential—a critical step in any data science project (Peng et al., 2021). In this paper, we presented the design, development, and use of an EDA tool integrated into the *DaTu* system.

Our presentation of the EDA toolbox, which is integrated within the *DaTu* system, provided a detailed overview of its design elements and functionalities. This allowed us to showcase the robust nature of the EDA toolbox, highlighting its versatility and ability to effectively support the various learning activities and tasks within the *DaTu* system.

The user study found that participants allocated varying amounts of time to using the EDA tool in three data science programming problems. We computed the total time spent on specific tasks such as Data Review, Visualization, Modeling, and Prediction, and then performed an analysis on the breakdown of time distribution for each task in relation to overall time spent on using the EDA tool. Results demonstrate that participants predominantly invested time in Data Review and Prediction tasks across all three problems, showcasing a preference for understanding the dataset and making predictions rather than thoroughly exploring visualizations or constructing models. Additionally, the percentage of total problem-solving time for each of the three data science programming problems was computed in relation to the overall time spent utilizing the EDA toolbox. In the first two problems, participants spent an average of 20% of their problem-solving time on the EDA tool, indicating its importance in exploring and understanding the data. However, for the third problem, only 3% of the total time was spent on the EDA tool, suggesting its lesser significance in a free-form programming task. This highlights the importance of considering the specific problem context when using the EDA tool. Understanding these varying patterns of tool usage provides valuable insights into optimizing the workflow for data exploration and analysis in different settings. Moreover, the research results indicate that the inclusion of a more extensive range of data sets derived from practical scenarios across various fields can augment the effectiveness of the EDA tool.

Several limitations and future directions merit consideration to further refine and enhance the functionality and educational value of the EDA tool within the *DaTu* system. While the current implementation allows users to upload datasets, the implementation has only three datasets. To enhance the EDA tool's utility, it would be beneficial to incorporate a wider array of datasets representing real-world applications of EDA techniques in diverse domains such as finance, healthcare, and computer science. Future research should incorporate the collection and consideration of student feedback to provide valuable insights for iterative improvements to enhance the tool's effectiveness in supporting efficient data exploration and analysis. Furthermore, efforts should be made to ensure the accessibility of the EDA tool to accommodate users with diverse needs and abilities. We hope that our EDA tool will spark new ideas and approaches in data science education and research.

References

Acciarini, C., Cappa, F., Boccardelli, P., & Oriani, R. (2023). How can organizations leverage big data to innovate their business models? A systematic literature review. *Technovation, 123*, 102713. https://doi.org/10.1016/j.technovation.2023.102713

Batch, A., & Elmqvist, N. (2018). The interactive visualization gap in initial exploratory data analysis. IEEE Transactions on Visualization *and Computer Graphics, 24*(1), 278–287. https://doi.org/10.1109/tvcg.2017.2743990

Coners, A., Matthies, B., Vollenberg, C., & Koch, J. (2024a). Data Skills for everyone! (?) – an approach to assessing the integration of data literacy and data science competencies in higher education. *Journal of Statistics and Data Science Education*, 1–29. https://doi.org/10.1080/26939169.2024.2334408

Crisan, A., Fiore-Gartland, B., & Tory, M. (2021). Passing the Data Baton : A Retrospective Analysis on Data Science Work and Workers. IEEE Transactions on Visualization and Computer Graphics, 27(2), 1860–1870. https://doi.org/10.1109/tvcg.2020.3030340

Doleck, T., Agand, P., & Pirrotta, D. (2024). Integrating Generative AI in Data Science Programming: Group Differences in Hint Requests. *Computers in Human Behavior: Artificial Humans, 2*(2), 100089. doi: 10.1016/j.chbah.2024.100089

Fernandes, E., Moro, S., & Cortez, P. (2023). Data Science, Machine Learning and Big Data in Digital Journalism: A Survey of state-of-the-art, challenges and opportunities. *Expert Systems with Applications, 221*, 119795. https://doi.org/10.1016/j.eswa.2023.119795

George, G., Osinga, E. C., Lavie, D., & Scott, B. A. (2016). Big Data and Data Science Methods for Management Research. *Academy of Management Journal, 59*(5), 1493–1507. https://doi.org/10.5465/amj.2016.4005

Jahani, H., Jain, R., & Ivanov, D. (2023). Data Science and Big Data Analytics: A systematic review of methodologies used in the supply chain and Logistics Research. Annals of Operations Research. https://doi.org/10.1007/s10479-023-05390-7

Jansen, B. J., Aldous, K. K., Salminen, J., Almerekhi, H., & Jung, S. G. (2023). *Data Preprocessing.* In Understanding Audiences, Customers, and Users via Analytics: An Introduction to the Employment of Web, Social, and Other Types of Digital People Data (pp. 65-75). Cham: Springer Nature Switzerland.

Jebb, A. T., Parrigon, S., & Woo, S. E. (2017). Exploratory data analysis as a foundation of Inductive Research. *Human Resource Management Review, 27*(2), 265–276. https://doi.org/10.1016/j.hrmr.2016.08.003

Jiang, Q., Sun, G., Li, T., Tang, J., Xia, W., Zhu, S., & Liang, R. (2024). Qutaber: Task-based exploratory data analysis with enriched context awareness. *Journal of Visualization.* https://doi.org/10.1007/s12650-024-00975-1

Kamalov, F., Moussa, S., & Reyes, J. A. (2023, November). Data Transformation in Machine Learning: Empirical Analysis. In 2023 International Conference on Innovation and Intelligence for Informatics, *Computing, and Technologies* (3ICT) (pp. 115-120). IEEE.

Liu, F., & Demosthenes, P. (2022). Real-world data: a brief review of the methods, applications, challenges and opportunities. BMC *Medical Research Methodology, 22*(1). https://doi.org/10.1186/s12874-022-01768-6

McKinney, D., Morton, C., Tuohy, B., Berg, S., Karlstad, A., Ortega, C., ... & Kao, Y. (2024, March). Iterative Design of a Socially-Relevant and Engaging Middle School Data Science Unit. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (pp. 826-832).

Memarian, B., & Doleck, T. (2023). Data Science Pedagogical Tools and Practices: A Systematic Literature Review. *Education and Information Technologies*. doi: 10.1007/s10639-023-12102-y

Mike, K., & Hazzan, O. (2023). What is Data Science? *Communications of the ACM, 66*(2), 12–13. https://doi.org/10.1145/3575663

Ovando-Fuentealba, L., Blake, W. H., Taylor, A., Clason, C., & Bravo-Linares, C. (2024). An exploratory data analysis (EDA) tool for tracer selection in sediment and pollution fingerprinting studies. *Environmental Forensics*, 1–13. https://doi.org/10.1080/15275922.2024.2330018

Ow-Yeong, Y. K., Yeter, I. H., & Ali, F. (2023). Learning data science in elementary school mathematics: A comparative curriculum analysis. *International Journal of STEM Education, 10*(1). https://doi.org/10.1186/s40594-023-00397-9

Pearson, R. K. (2018). Exploratory data analysis using R. Chapman and Hall/CRC.

Peng, J., Wu, W., Lockhart, B., Bian, S., Yan, J. N., Xu, L., ... & Wang, J. (2021). Dataprep. eda: Task-centric exploratory data analysis for statistical modeling in python. In Proceedings of the *2021 International Conference on Management of Data* (pp. 2271-2280).

Power, D. J. (2016). Data science: Supporting decision-making. *Journal of Decision Systems, 25*(4), 345–356. https://doi.org/10.1080/12460125.2016.1171610

Ra, S., Shrestha, U., Khatiwada, S., Yoon, S. W., & Kwon, K. (2019). The Rise of Technology and Impact on Skills. *International Journal of Training Research, 17*(1), 26–40. https://doi.org/10.1080/14480220.2019.1629727

Sarker, I. H. (2021). Data Science and Analytics: An overview from data-driven smart computing, decision-making and applications perspective. *SN Computer Science, 2*(5). https://doi.org/10.1007/s42979-021-00765-8

Wongsuphasawat, K., Liu, Y., & Heer, J. (2019). Goals, process, and challenges of exploratory data analysis: An interview study. arXiv preprint arXiv:1911.00568.

Zeitz, J., Self, N., House, L., Evia, J. R., Leman, S., & North, C. (2018). Bringing interactive visual analytics to the classroom for developing EDA skills. *Journal of Computing Sciences in Colleges, 33*(3), 115-125.

Zhang, Y., Wu, D., Hagen, L., Song, I., Mostafa, J., Oh, S., Anderson, T., Shah, C., Bishop, B. W., Hopfgartner, F., Eckert, K., Federer, L., & Saltz, J. S. (2022). Data science curriculum in the field. *Journal of the Association for Information Science and Technology, 74*(6), 641–662. https://doi.org/10.1002/asi.24701