



Research Article

CodelessML: A Beginner's Web Application for Getting Started with Machine Learning

Hanif Noer ROFIQ^{1,*}  Galuh Mafela Mutiara SUJAK² 

¹ Ministry of Finance of Republic Indonesia, Indonesia hanif.noer94@gmail.com

² Ministry of Finance of Republic Indonesia, Indonesia galuhmafela@gmail.com


* Corresponding Author: hanif.noer94@gmail.com

Article Info

Received: 28 June 2024

Accepted: 06 September 2024

Keywords: Machine learning, learning, barrier, software

 10.18009/jcer.1506864

Publication Language: English

Abstract

Building machine learning models requires intensive coding and installation of certain software. This is frequently a barrier for beginners learning about machine learning. To overcome this situation, we present CodelessML, a reproducible web-based application designed for Machine Learning beginners due to its coding-free and installation-free design, published under Code Ocean capsule. It provides a common workflow that eases the process of building Machine Learning models and using the model for predictions. Using the Agile method, CodelessML was successfully built using Python, Anaconda, and Streamlit It. By using CodelessML, users can get a walkthrough and interactive experience of building machine learning through a simplified machine learning process: exploratory data analytics (EDA), modelling, and prediction. The impact of the software was evaluated based on feedback from 79 respondents, which showed that based on a 5-scale Likert, CodelessML received average ratings of 4.4 in accessibility, 4.3 in content, and 4.4 in functionality. CodelessML serves as an accessible entry point for learning machine learning, offering online, free, and reproducible features.



To cite this article: Rofiq, H.N. & Sujak, G.M.M. (2024). CodelessML: A beginner's web application for getting started with machine learning. *Journal of Computer and Education Research*, 12 (24), 582-599. <https://doi.org/10.18009/jcer.1506864>

Introduction

Data analysis has a long history (Tukey, 1962) and has often been used to help carry out tasks and achieve goals (Register & Ko, 2020). In this era, it has evolved into more complex applications, driven by the rising popularity of machine learning, which builds on traditional data analytics by enabling predictive models and automating decision-making processes. The origin history of machine learning in its modern sense is typically associated with Frank Rosenblatt, who created a group that built a machine for recognising the letter of the alphabet Rosenblatt in 1957 (Fradkov, 2020). The machine learning era continued until its turning point at the beginning of the first XXI decade. At that time, the rapidness of machine learning was enabled due to the Big Data trend, reduced cost of parallel computing and

memory, and the new development of deep machine learning algorithms (Fradkov, 2020). With the exponential growth in data volume, machine learning has evolved into a widely utilised tool for data analysis across various fields (Kononenko, 2001; Tetzlaff & Szepannek, 2022). Consequently, Machine Learning has gained significant enthusiasm from professionals and the general public (Fradkov, 2020).

Machine learning is closely related to artificial intelligence (AI). Some countries have begun to chase the opportunity from early education, for example, Hong Kong, America, and Iraq, which are trying to incorporate artificial intelligence-related subjects into their K-12 grade schools (Sallow et al., 2024; Wang & Cheng, 2021; Woodruff et al., 2023). In Indonesia, machine learning is also growing in popularity among vast audiences, including the Ministry of Finance of the Republic of Indonesia. In line with its commitment to becoming a data-driven organisation, the Ministry of Finance has provided its employees with diverse artificial intelligence-related training, including the data analytics subset of training.

Despite its popularity, there is a learning gap for beginners due to machine learning models' rapid development and complexity (Tetzlaff & Szepannek, 2022). As Woodruff et al. (2023) implied, there are learning barriers to learning artificial intelligence, including technical challenges and resource constraints. Wang and Cheng (2021) identified several similar issues for artificial intelligence learning, including the uncertainty of hardware and learning kits and the technical complexity of which can be intimidating for those who are not tech-savvy. Implementing it also requires substantial resources, including financial investment, infrastructure, and training.

Furthermore, people who want to learn about machine learning need to have sufficient knowledge of coding, as the machine learning's algorithms are written in code and implemented through programming. Machine learning itself is an evolving branch of computational algorithms that originated from computer sciences and statistics (Naqa & Murphy, 2015). One of the renowned methodologies, CRISP-DM (Cross Industry Standard Process for Data Mining), which was introduced by Chapman et al. (2000), also showed that most of the cycle in data analytics involves coding and programming stages. Even though the CRISP-DM was proposed as a data mining methodology at the end of the 20th century, it remains widely recognised as the standard framework for organising and managing data mining and machine learning projects (Martinez-Plumed et al., 2019). CRISP-DM depicts the process in reiterative phases: business understanding, data understanding, data preparation,

modelling, evaluation, and deployment. From the stages outlined above, it is discernible that machine learning heavily relies on programming languages, which are essential for data preparation, modelling, evaluation, and deployment phases. However, many algorithms are time-consuming and costly to code one by one (Burscher et al., 2014), making it difficult for beginners to build their models.

Additionally, other requirements must be met to perform machine learning analysis, such as installing and learning the software that will be used. The installation process for these applications can pose challenges for beginners learning machine learning, as it often involves a substantial number of programs and libraries. The foundational languages of machine learning range from programming environments such as Python, C++, R, and Julia (Sarkar et al., 2017). Although there are numerous options available, Python is widely regarded as the most suitable language for teaching introductory statistics in a data-rich environment for machine learning education (Ozgun et al., 2021; Sarkar et al., 2017) and has extensive libraries available for machine learning, such as NumPy, SciPy, TensorFlow, and scikit-learn (Liu, 2020). To further utilise Python in ML learning, we need to install other tools, such as Anaconda and Jupyter Notebook. Minimum computer requirements must also be met to ensure the software operates effectively, which adds additional consideration for beginners learning about machine learning.

Those barriers highlight the challenges that must be addressed to create a more conducive environment for educating someone about machine learning, particularly for adult learners. Many argue that educational machine learning should focus on a broader context than technical issues like coding and programming (Wang & Cheng, 2021). As for novices, especially working employees, the key idea of learning about machine learning is to introduce machine learning to them. Acquiring fundamental coding skills is quite challenging as it requires additional effort and time, which can be particularly demanding for working professionals with no data science and programming background. Therefore, it is essential to introduce user-friendly tools that enable non-technical employees to build their models (Dyck, 2018; Ferguson, 2017; Z.-H. Zhou, 2017).

To fill the capacity and requirement gap to learn the basics of machine learning, we built a website-based application that can be used to learn machine learning without the need for any installation or coding. This application is called CodelessML, which has features that can assist users in doing Exploratory Data Analytics (EDA) and creating

machine learning models (regression and classification). Users can also download predictive models that have been built to be used with new data sets. This application gives a handy workflow that eases the process of learning Machine Learning for beginners. It has been tested in workshops, discussions, and closed training where the participants are all beginners starting to learn machine learning. For further action, it can be widely used to reach wider audiences.

Method

The software was built using the Agile Method. It is a structured approach dividing the project into several continuous phases. We followed the five-step agile method: Plan, Design and Develop, Test, Deploy, and Review, as shown in Figure 1 below. This approach was chosen to facilitate continuous improvement and responsiveness to user needs. For instance, a data splitting option was added based on user feedback, allowing users to adjust the training and testing split ratio. Initially fixed at 70:30, this ratio was made flexible in response to iterative feedback, giving users greater control over the model-building process. The Agile process employed in CodelessML focuses on iterative improvements and adapting to user feedback, ensuring the tool evolves in alignment with user needs and expectations.

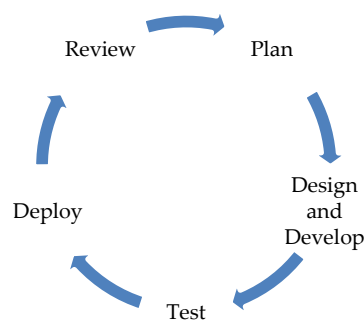


Figure 1. Agile method used in research

Planning CodelessML began when the problem occurred. To construct CodelessML, we used a set of tools: Python, Anaconda Navigator, and Streamlit. The software is licensed under the Apache License 2.0, and we use Git for code versioning. The required environment for compiling and running CodelessML is Python 3.9.16.

The CodelessML's user interface is built with Streamlit, and it uses several libraries for processing and modelling, including Pandas (McKinney, 2010), scikit-learn (Fabian, 2011), XGBoost (Ke et al., 2017), and LightGBM (Chen & Guestrin, 2016). CodelessML has three main menus: EDA, Modeling, and Prediction. The user interface is designed to follow

the general workflow, making it easier for users to learn machine learning. The procedures and the software workflow are illustrated in Figure 2.

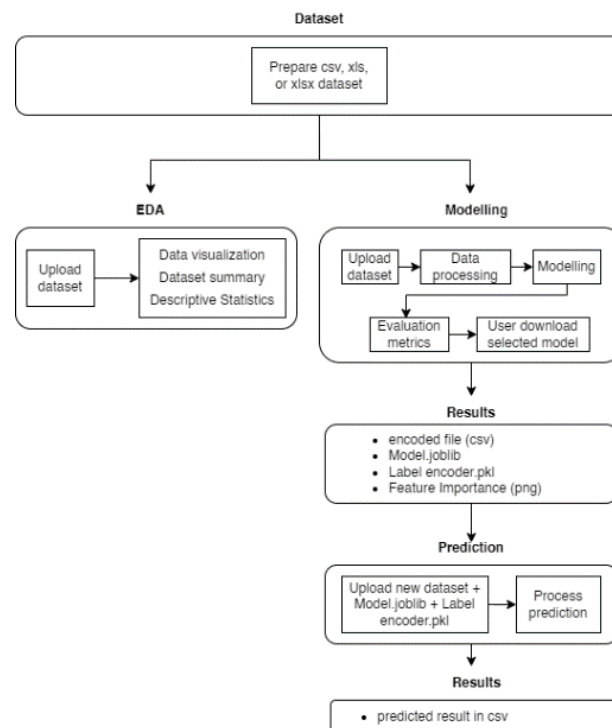


Figure 2. Software workflow

CodelessML was tested on two datasets, the Wine dataset (Aeberhard & Forina, 1991) for classification and the Automobile dataset (Schlimmer, 1987) for regression. The Wine dataset consists of chemical properties and quality ratings of wines from the same region in Italy. It has 13 features and 178 instances, and it is commonly used for classification tasks, particularly in predicting wine quality based on its attributes. The Wine dataset was selected for its simplicity and frequent use in educational contexts to demonstrate classification algorithms. The Automobile dataset contains information about various car models, including attributes such as engine size, horsepower, body style, and fuel system, making it ideal for regression tasks where continuous variables like price prediction are involved. It includes 25 features with 205 instances, making it suitable for testing various machine learning algorithms on a moderately sized dataset. The automobile dataset was chosen for its practical use in predicting car prices, a common application of machine learning in the automotive industry. While CodelessML uses these datasets as examples, users can also upload and work with their own data. These examples were chosen to cover both regression

and classification tasks, allowing users to experiment with real-world data and gain practical, hands-on experience with different machine learning applications.

CodelessML is also equipped with several evaluation metrics to measure its Classification and Regression Model. Machine learning models are generated from iterated and complex model-building processes. Therefore, no single measure can evaluate classifier performance (Mohamed, 2017), so different evaluation metrics are used for different methods of observing the model's performance (Novakovic et al., 2017). Classification is used to classify the object types (Novakovic et al., 2017), and its performance is commonly measured using evaluation metrics such as accuracy, precision, recall, and F1 score (Opitz, 2024). In contrast, regression is used to predict numeric outcomes (Grandini et al., 2020) with performance metrics comparing the predicted values to the actual results (Botchkarev, 2018) such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and R^2 (Botchkarev, 2018; Botchkarev, 2019). CodelessML provides these commonly used evaluation metrics for classification and regression to assess model performance.

To evaluate the usability of CodelessML for beginners without prior knowledge of programming or machine learning, a survey was conducted by the lead author, who occasionally teaches data analytics classes for the Ministry of Finance. CodelessML was made open-source and accessible online, and it was introduced to participants during various events in 2023, including lectures, workshops, and discussions. 87 participants were introduced to CodelessML, and 79 completed the questionnaire, resulting in a response rate of 90.80%. The survey, distributed through Google Forms, provided valuable insights into the effectiveness of CodelessML in assisting beginners with their learning journey. The sample size was limited to employees within the Ministry of Finance of the Republic of Indonesia, with most respondents coming from the Directorate General of State Asset Management (DGSAM), where the authors are employed. Most employees in this organisation are beginners in machine learning due to their primary education and job responsibilities not being related to this field, making it challenging to learn machine learning concepts. Given their heavy workloads, acquiring new skills can be difficult, underscoring the need for innovative tools to facilitate learning. Although CodelessML was primarily introduced internally within the Ministry of Finance, its source code is openly

accessible on platforms such as GitHub and Code Ocean, with the hope that it can also assist other educators facing similar challenges.

The questionnaire was constructed into 8 Likert-scale questions, divided into three 3 sections to understand the Accessibility, Content, and Functionality of CodelessML from the learner's perspectives as shown in Table 1.

Table 1. Questionnaire used in research.

Item Number	Question	Aspects Observed
1.	I don't feel any difficulties when accessing CodelessML	Accessibility
2.	CodelessML is easy to use and has clear navigation	Content
3.	The instructions in CodelessML helped me in operating the Application	Content
4.	CodelessML was useful for me in understanding the basics of Machine Learning	Functionality
5.	I did not experience any difficulties when using the EDA menu	Functionality
6.	I did not experience any difficulties in using the Modeling menu	Functionality
7.	I did not experience any difficulties in using the Prediction menu	Functionality
8.	I would recommend CodelessML to others who want to start learning Machine Learning	Functionality

The survey is tested using Pearson Correlation to prove its validity and Cronbach Alpha to measure its reliability. The validity test is carried out by using Pearson Correlation test to assess whether or not data is correlated with another. The validity test result using Pearson Correlation is shown in Table 2. r_{table} is given for 5% significance. The results showed that $r_{xy} > r_{table}$ indicates that all items in the questionnaire are valid.

Table 2. Validity test result

Item Number	r_{xy}	r_{table}	Result (valid if $r_{xy} > r_{table}$)
1	0,812	0.1841	Valid
2	0.667	0.1841	Valid
3	0.678	0.1841	Valid
4	0.705	0.1841	Valid
5	0.828	0.1841	Valid
6	0.737	0.1841	Valid
7	0.693	0.1841	Valid
8	0.757	0.1841	Valid

Meanwhile, the Cronbach Alpha score for the survey is 0.892. Cronbach Alpha is a measurement to determine reliability and consistency inside the survey items. The closer the

Cronbach Alpha value to 1 means it has greater internal consistency and reliability. The value is usually acceptable between 0.70 and 0.90 or higher depending on the research type (Adeniran, 2019). From the analysis, the given Cronbach Alpha score is greater than the lowest acceptable reliability limit of 0.6 or 0.7 (Hair et al., 2019), so the survey is considered reliable.

Result

CodelessML was successfully built using Python and deployed using Streamlit. It is a reproducible open-access software published under Code Ocean capsule (<https://codeocean.com/capsule/5407148/tree/v1>). The final deployment of the software is shown in Figure 3. CodelessML features five main menus: "About," "EDA," "Modelling - Classification," "Modelling - Regression," and "Prediction." The "About" menu provides an application overview and includes user instructions. The "EDA" (Exploratory Data Analysis) menu offers descriptive statistics, dataset summaries, and various visualizations of uploaded datasets to give users a comprehensive overview of their data. The "Modelling - Classification" and "Modelling - Regression" menus are used to build machine learning models, while the "Prediction" menu allows users to make predictions based on the trained models.

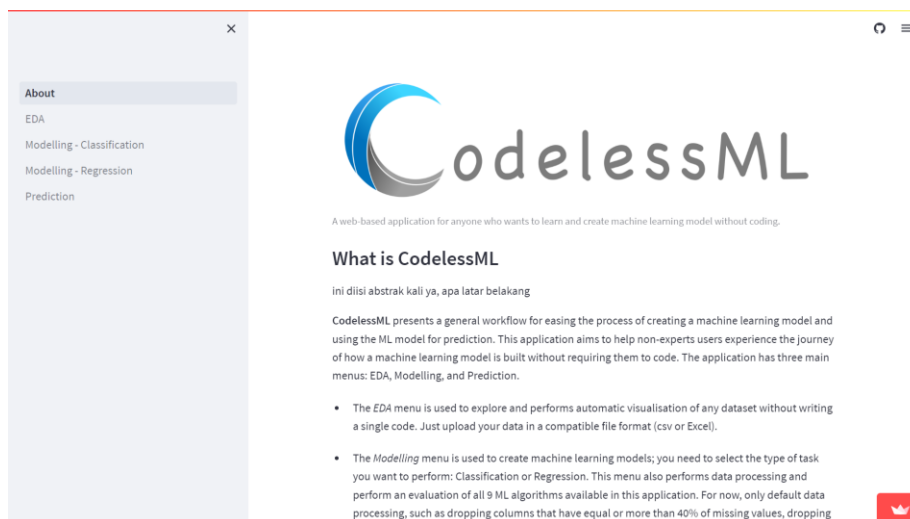


Figure 3. CodelessML main menu

The EDA (Exploratory Data Analysis) menu is designed to help users explore and analyse datasets to gain insights into data distribution and relationships between variables, identify patterns, detect anomalies, test hypotheses, and verify assumptions. Users can upload a CSV, XLSX, or XLS file by dragging it into the interface or using the "browse file" button. Once uploaded, the menu displays a sample of the dataset, a summary, descriptive

statistics, visualizations for numerical and categorical data distributions, a correlation matrix, and options for custom plots like boxplots or scatterplots. This step can be skipped if users have already processed the data in another software or are familiar with the dataset.

The modelling menu consists of Modelling-classification for the classification task and Modelling-regression for the regression task. To use this menu, users must upload a CSV, XLSX, or XLS file by dragging it into the interface or clicking the "browse file" button. If users have already utilized the EDA menu, they can skip the upload step and directly select the target column, set the data splitting ratio, and click "Submit" to automatically train the data using nine different machine learning algorithms. During this process, CodelessML applies default data processing, such as dropping columns with 40% or more missing values, removing columns that are 100% unique, and imputing missing values with the mean for numerical data or mode for categorical data. The modeling menu also divides the dataset into training and test data with a default size of 70% for training data and 30% for test data. The amount of distribution in this dataset can be changed based on the user's preferences. Once training is complete, the evaluation metrics for each model are displayed, allowing users to choose and download their preferred model.

The output from the modelling menu on CodelessML is a compressed zipped folder that contains coded input data, the results of a classification or regression model (model.joblib), the save of the encoder label, which functions to convert non-numeric data into a numeric format that matches the training data (label encoder.pkl), and Feature Importance of selected model in png format. The Feature Importance contains each feature's contribution to the outcome (Nohara et al., 2022) and is only available in Regression Models other than Support Vector Regression and K-Nearest Neighbors. The input file that has been processed using the encoder label is returned to the user so that the user can understand what kind of file is ready for modelling.

The Prediction menu is linked to the Modelling menu, as it relies on the output generated from the previous Modelling menu. In the Prediction menu, users must upload two output files from the Modeling menu: the encoder (.pkl) and the model (.joblib), along with the new, unseen data to be predicted. The encoder (.pkl) file is necessary to convert non-numeric data into a numeric format that matches the training data. The model (.joblib) file is the machine learning model created in the Modelling menu, which will be used for prediction. Once all required files are uploaded, the option to select columns to ignore will

show up, and there is a preview of the unseen data that will be predicted to make sure that every column name matches the data used in the Modelling. After confirming the column match, the tool will display a sample output, and the user can download the full prediction results in a CSV file named prediction.csv.

CodelessML Illustrative Example

We demonstrate the capabilities of CodelessML using two datasets: the Wine dataset (Aeberhard & Forina, 1991) for classification and the Automobile dataset (Schlimmer, 1987) for regression. In this example, we focus on the regression analysis using the Automobile dataset.

The process begins with the Exploratory Data Analysis (EDA) menu, where the user uploads the Automobile training dataset. Upon upload, CodelessML automatically provides a sample of the dataset (Figure 4), allowing users to understand the dataset column and properties.

The screenshot shows the CodelessML application interface. On the left is a navigation menu with options: About, EDA (selected), Modelling - Classification, Modelling - Regression, and Prediction. The main area displays the EDA workflow. At the top, it shows a cache object reference: `<streamlit.runtime.caching.cache_data_api.CacheDataAPI object at 0x7fb318d0a2b0>`. Below this is an upload section with the text "Upload your data here" and a "Drag and drop file here" area with a "Browse files" button. A file named "automobile_train.csv" (21.4KB) is shown as uploaded. Below the upload section, it states "uploaded automobile_train.csv with the shape of (164, 26)". A table titled "dataset sample" is displayed, showing a preview of the data. Below the table is a "Summary of the dataset" section with a table of statistics.

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-v
16	2	104	saab	gas	std	four	sedan	fwd
156	1	154	plymouth	gas	std	four	hatchback	fwd
80	0	91	toyota	diesel	std	four	sedan	fwd
85	3	194	nissan	gas	turbo	two	hatchback	rwd
73	2	94	volkswagen	diesel	turbo	four	sedan	fwd

Summary of the dataset

Column	dtypes	non-null	Missing	Missing (%)	Uniques
--------	--------	----------	---------	-------------	---------

Figure 4. Sample of automobile dataset in CodelessML application

Below the dataset sample table, summary statistics (Figure 5) and descriptive analytics (Figure 6) are displayed.

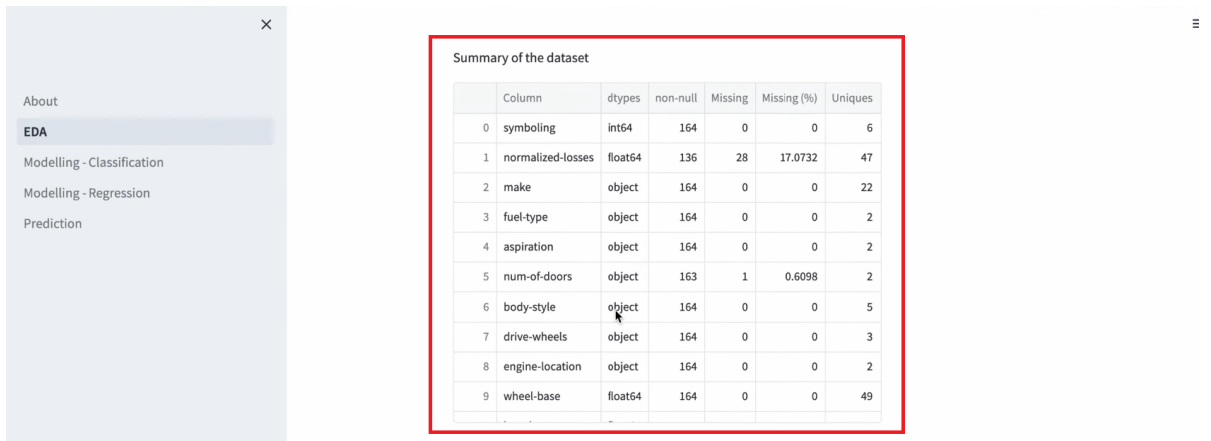


Figure 5. Summary of the automobile dataset in CodelessML

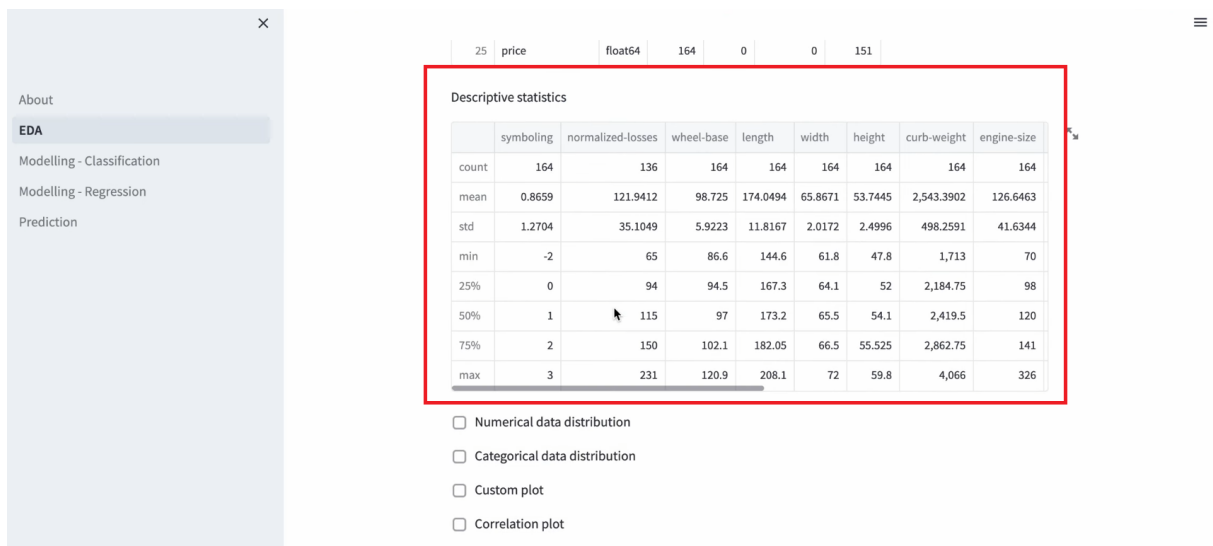


Figure 6. Descriptive analytics of automobile dataset in CodelessML application

Users can select options to visualise numerical data distributions, categorical data distributions, custom plots, and correlation plots. Here are some graph examples: Figure 7 below depicts the numerical data distribution from the dataset; Figure 8 shows the scatterplot; Figure 9 shows the categorical data distribution; and Figure 10 shows the correlation plot.

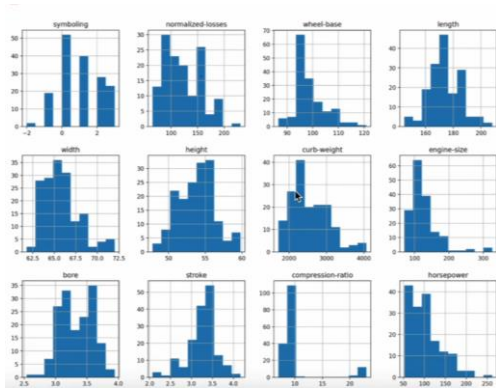


Figure 7. Numerical data distribution graph in CodelessML

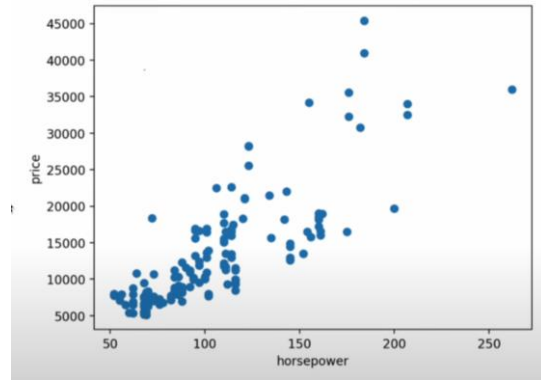


Figure 8. Scatterplot in CodelessML

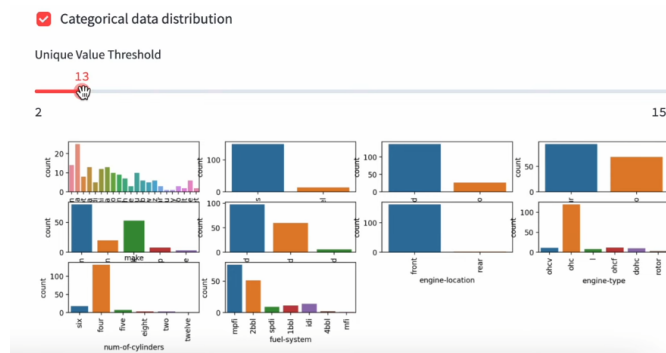


Figure 9. Categorical data distribution in the CodelessML

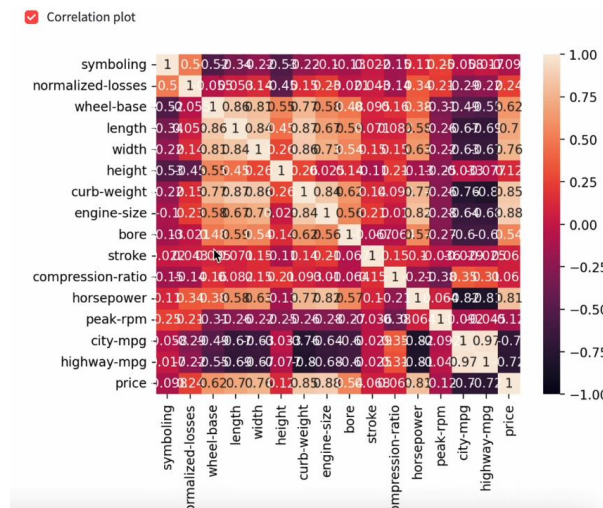


Figure 10. Correlation plot in CodelessML application

In the Modelling - Regression menu, we set "price" as the targeted column, aiming to predict automobile prices. The data was split into training and testing sets using an 80:20 ratio. After processing, CodelessML generated models using nine different algorithms, with each performance metric as shown in Figure 11. Users can sort and select the best model

based on their preferred metrics. In this case, the Random Forest algorithm was chosen with an R^2 score of 0.928767, MSE of 2,538,853.761265, RMSE of 1,593.378098, MAE of 1,167.756414, and MAPE of 0.095964. The Random Forest model was subsequently downloaded, consisting of encoded data, a .joblib model file, and a .pkl encoder file, as shown in Figure 12.

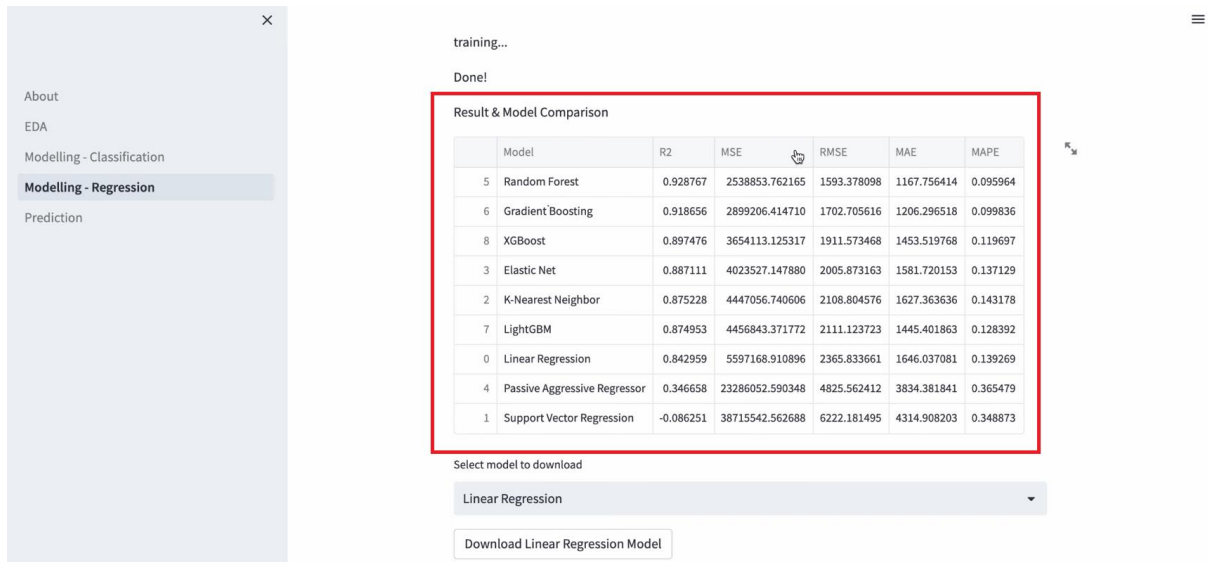


Figure 11. Regression result and model comparison

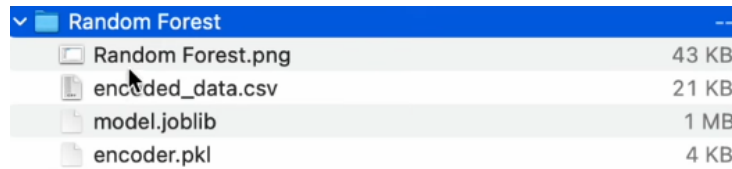


Figure 12. Regression result and model comparison

Finally, in the Prediction menu, the user uploads the test dataset, along with the .pkl encoder file and the .joblib model file. CodelessML then provided a table containing the predicted and actual automobile prices, as shown in Figure 13.

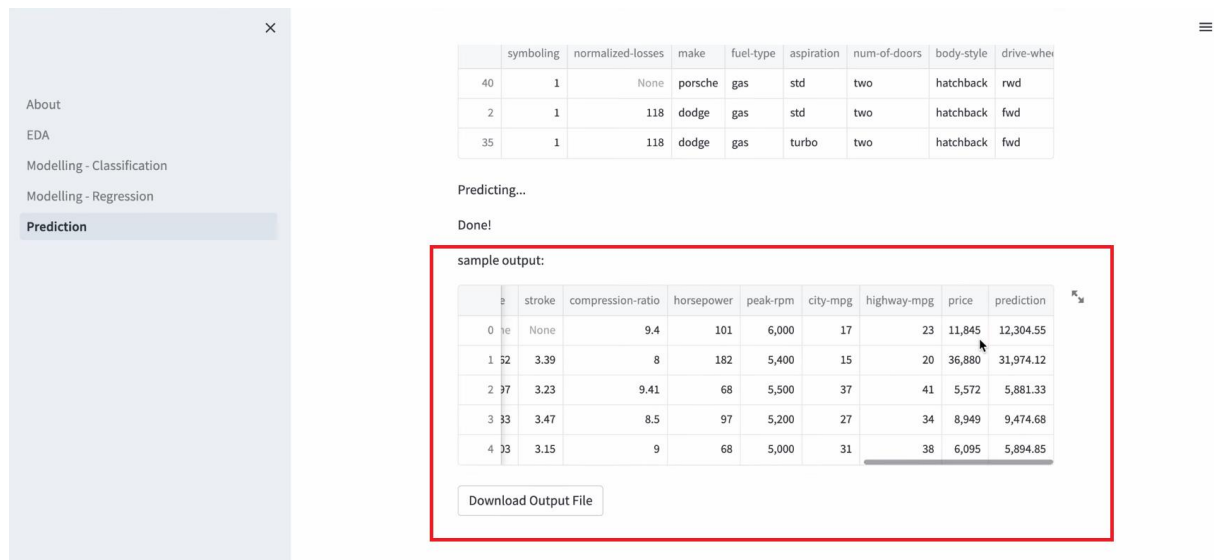


Figure 13. Regression result and model comparison

CodelessML Impact

The impact of CodelessML was measured through a survey that was conducted on 79 respondents. The questionnaires are given to determine three quality factors of visitor perspective: accessibility, content, and usability (Nabil et al., 2011). Calculated from the Likert scale average, the results are shown in Table 3 below:

Table 3. Likert average score based on each aspect

Number	Aspect	Likert Average Score
1.	Accessibility	4.4
2.	Content	4.3
3.	Functionality	4.4

In addition to the questionnaire, the respondents were also presented with a tick box and open-ended questions, asking their opinion about the CodelessML. The complete questionnaire is attached in the supplementary materials. From the calculated average scale and the questions' answers, it can be concluded that the respondents Agree (point 4 in the Likert Scale) that CodelessML is accessible and easy to use, has clear navigation and documentation, and has an impact on helping to understand machine learning. Respondents also would recommend CodelessML for other people starting to learn Machine Learning.

Conclusion

In this work, we introduce CodelessML, a tool designed for anyone interested in learning about and creating machine learning models without needing programming skills or specific software installations.

The software possesses three functionalities: EDA, Modeling, and Prediction:

- *EDA*: CodelessML provides data exploration tools, allowing users to summarize and visualize the uploaded dataset and provide the dataset's descriptive statistics.
- *Modelling*: CodelessML comes with preconfigured models for classification and regression. Users can freely configure the models they want and download the selected model to be used in the Prediction Menu.
- *Prediction*: Users can upload the selected downloaded model from the Modelling Menu along with a new dataset to yield new prediction results.

A survey of 79 beginners with no prior experience in machine learning concluded that respondents rated CodelessML as accessible and easy to use, with clear navigation and documentation. Each of the following criteria scored 4.4, 4.3, 4.4 (5-scale likert) for accessibility, content, and functionality. Additionally, the majority indicated that CodelessML significantly aids in understanding machine learning concepts and expressed willingness to recommend it to others starting in the field. The results demonstrate that CodelessML effectively facilitates learning machine learning without the need for coding skills or additional software installation. From the survey, CodelessML has proven to help the respondents learn machine learning easily without coding skills and installing certain software. Users can also compare three Machine Learning models at once without coding each individually. Finally, CodelessML can help users gain an initial understanding of Machine Learning before mastering it further.

Study Limitation and Future Development

CodelessML offers limited customization in data processing, with fixed options for handling missing values, dropping columns, and imputing using mean or mode. While this simplicity benefits non-expert users, advanced users may find it restrictive, especially when dealing with complex datasets that require specific imputation techniques or transformations. The CodelessML supports only 9 machine learning algorithms, allowing beginners to easily explore models without coding one by one. However, advanced users

may feel constrained by the lack of hyperparameter tuning, which can limit model optimization. Additionally, CodelessML's performance with large datasets depends on server capacity in the hosted version or the user's hardware when installed locally. Handling large datasets demands significant computational resources, which CodelessML cannot influence. As a result, users with large datasets may experience slow performance or memory issues, particularly on less powerful machines. However, small to medium datasets typically perform well without significant performance issues.

CodelessML's future development will be driven by user feedback and specific requirements, primarily focusing on helping inexperienced users understand machine learning and data science workflows. Planned feature updates may enhance the EDA feature and add additional machine learning models while maintaining simplicity and accessibility. As CodelessML is designed solely for educational purposes, its development will remain targeted toward non-expert users without integrating external platforms to ensure a focused and streamlined experience.

Acknowledgement

Due to the scope and method of the study, ethics committee permission was not required.

Author Contribution Statement

Hanif Noer ROFIQ : *Conceptualization, coding, survey question design, data collection, data analysis, implementation, writing, and translation.*

Galuh Mafela Mutiara SUJAK: *Conceptualization, literature review, survey question design, data collection, implementation, and writing.*

References

- Adeniran, A. O. (2019). Application of Likert scale's type and Cronbach's alpha analysis in an airport perception study. *Scholar Journal of Applied Sciences and Research*, 2(4), 1-5.
- Aeberhard, S., & Forina, M. (1991). Wine [Data set]. UCI Machine Learning Repository, 10, C5PC7J.
- Botchkarev, A. (2018). Performance metrics (Error Measures) in machine learning regression, forecasting and prognostics: Properties and typology. *ArXiv*.
- Botchkarev, A. (2019). A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information Knowledge and Management*, 14, 045–076. <https://doi.org/10.28945/4184> .

- Burscher, B., Odijk, D., Vliegthart, R., De Rijke, M., & De Vreese, C. H. (2014). Teaching the computer to code frames in news: comparing two supervised machine learning approaches to frame analysis. *Communication Methods and Measures*, 8(3), 190–206.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. (2000). CRISP-DM 1.0 - Step-by-step data mining guide. *CRISP-DM Consortium*.
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794. <https://doi.org/10.1145/2939672.2939785>.
- Dyck, J. (2018). *Machine learning for engineering*. In: Proceedings of the 23rd Asia and South Pacific Design Automation Conference. IEEE Press, pp. 422–427.
- Fabian, P. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, 2825-2830, <https://doi.org/10.1145/3369834>.
- Ferguson, A. L. (2017). Machine learning and data science in soft materials engineering. *Journal of Physics: Condensed Matter* 30(4).
- Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, 53(2), 1385–1390. <https://doi.org/10.1016/j.ifacol.2020.12.1888>.
- Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: An overview. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2008.05756>.
- Hair, J. F. J., Black, W. C., Babin, B. J., Anderson, R. E., Black, W. C., & Anderson, R. E. (2019). *Multivariate data analysis. Cengage Learning*.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, 30.
- Kononenko, I. (2001). Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1), 89-109.
- Liu, Y. (2020). *Python machine learning by example - Third Edition*.
- Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J., & Flach, P. (2021). CRISP-DM Twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3048–3061.
- McKinney, W. (2010, June). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 445(1), 51-56.
- Mohamed, A. E. (2017). Comparative study of four supervised machine learning techniques for classification. *International Journal of Applied*, 7(2), 1-15.
- Nabil, D., Mosad, A., & Hefny, H. A. (2011). Web-Based applications quality factors: A survey and a proposed conceptual model. *Egyptian Informatics Journal*, 12(3), 211-217. <https://doi.org/10.1016/j.eij.2011.09.003>.
- Naqa, I. E., & Murphy, M. J. (2015). What is machine learning? In *Springer eBooks* (pp. 3–11). https://doi.org/10.1007/978-3-319-18305-3_1

- Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*, 214, 106584.
- Novakovic, J. Dj., Veljovic, A., S. Ilic, S., Papic, Z., & Tomovic, M. (2017). Evaluation of classification models in machine learning. *Theory and Applications of Mathematics & Computer Science*, 7(1), 39–46.
- Opitz, J. (2024). A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *arXiv (Cornell University)*.
- Ozgur, C., Colliau, T., Rogers, G., & Hughes, Z. (2021). MatLab vs. Python vs. R. *Journal of Data Science*, 15(3), 355–372. [https://doi.org/10.6339/jds.201707_15\(3\).0001](https://doi.org/10.6339/jds.201707_15(3).0001).
- Register, Y., & Ko, A. J. (2020, August). Learning machine learning with personal data helps stakeholders ground advocacy arguments in model mechanics. *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 67-78,
- Sallow, A. B., Asaad, R. R., Ahmad, H. B., Abdulrahman, S. M., Hani, A. A., & Zeebaree, S. R. (2024). Machine learning skills to K–12. *Journal of Soft Computing and Data Mining*, 5(1), 132-141.
- Sarkar, D., Bali, R., & Sharma, T. (2017). The python machine learning ecosystem. In *Apress eBooks* (pp. 67–118). https://doi.org/10.1007/978-1-4842-3207-1_2.
- Schlimmer, J. (1987). Automobile [Data set]. UCI machine learning repository. DOI, 10, C5B01C. <https://doi.org/10.24432/C5B01C>.
- Tetzlaff, L. M., & Szepannek, G. (2022). mlr3shiny—State-of-the-art machine learning made easy. *SoftwareX*, 20, 101246. <https://doi.org/10.1016/j.softx.2022.101246>.
- Tukey, J. W. (1962). The future of data analysis. *The annals of mathematical statistics*, 33(1), 1-67.
- Wang, T., & Cheng, E. C. K. (2021). An investigation of barriers to Hong Kong K-12 schools incorporating Artificial Intelligence in education. *Computers and Education Artificial Intelligence*, 2, 100031. <https://doi.org/10.1016/j.caeai.2021.100031>.
- Woodruff, K., Hutson, J., & Arnone, K. (2023). Perceptions and barriers to adopting artificial intelligence in k-12 education: A survey of educators in fifty states. In *IntechOpen eBooks*. <https://doi.org/10.5772/intechopen.1002741>.
- Zhou, Z.-H. (2017). “Machine learning challenges and impact: an interview with Thomas Dietterich.” *National Science Review* 5(1), 54–58.