Mini-Project Report On

No-code Machine Learning Platform

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Gokul Baburaj (U2003087)
Joel Manuel C. J. (U2003106)
Maria Sabi (U2003127)
Merene Benson (U2003132)

Under the guidance of
Dr. Saritha S

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***"No-code machine learning platform"*** *is a bonafide work done by* ***Mr. Gokul Baburaj (U2003087), Mr. Joel Manual C. J. (U2003106), Ms. Maria Sabi (U2003127), Ms. Merene Benson (U2003132)****, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2022-2023.*

Dr. Preetha K. G.
Head of Department
Dept. of CSE
RSET

Mr. Uday Babu P.
Mini-Project Coordinator
Asst. Professor
Dept. of CSE
RSET

Dr. Saritha S.
Mini-Project Guide
Associate Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards **Dr. P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K. G.**, Head of Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini-project, "No-code machine learning platform".

We are highly indebted to our mini-project coordinators, **Mr. Uday Babu P.**, Assistant Professor, Department of Computer Science and Engineering and **Ms. Tripti C.**, Assistant Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our mini-project guide **Dr. Saritha S.**, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Gokul Baburaj

Joel Manuel C J

Maria Sabi

Merene Benson

ABSTRACT

There is a need for a comprehensive learning experience that combines the simplicity of a No Code environment with an in-depth understanding of ML algorithms and concepts. By bridging this gap, the platform would offer a faster and more accessible way for beginners to learn ML, encouraging them to dive deeper into the field and eventually transition to more advanced coding techniques when they are ready.

The project is a No Code Machine Learning platform called Codio that generates Python code for various machine learning algorithms. Through this project, we aim to reduce the learning curve for anyone who wishes to learn machine learning by making it easier to experiment with the concepts without knowing the complicated syntax first.

Users can register themselves on the website and then log in to start generating code. They can either specify the path of their input file or choose from the inbuilt datasets. They select which algorithm to implement from the GUI and the parameters associated with the algorithm can be inputted/chosen from the menu. The code corresponding to the algorithm is generated and on clicking the download button, the code along with the machine learning model will be downloaded. Users also have the option to view information on these algorithms and their parameters for better understanding. An admin can log in and view the number of users and times the code has been generated. He/she can also update the information corresponding to the algorithms.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The history of AI has been a remarkable journey of human ingenuity. It all began in the 1950s when computer scientist John McCarthy coined the term "Artificial Intelligence" and sparked the birth of a formal field of study. However, early progress faced significant hurdles, leading to periods of disappointment known as "AI winters." It wasn't until the 2010s that a resurgence occurred, fueled by improved computing power, vast datasets, and breakthroughs in algorithms, particularly in the domain of deep learning. This resurgence brought AI into the mainstream, with applications like virtual assistants, recommendation systems, and autonomous vehicles becoming part of everyday life. The field continues to evolve rapidly, with ongoing research focusing on new frontiers like reinforcement learning and ethical AI, cementing AI and machine learning as transformative forces in modern society.

Machine learning can be challenging, as it involves understanding complex mathematical concepts and algorithms, as well as the ability to work with large amounts of data. However, with the right resources and support, it is possible to learn and become proficient in machine learning. It also depends on the individual's background and experience. Some people with a strong mathematical or programming background may find it easier to learn, while others may find it more difficult. Machine learning training and instruction might call for a great deal of commitment, in-depth understanding, as well as meticulousness.

Mastering programming language, honing your algorithmic skills, and paying close attention to artificial intelligence applications for merchandise and services are all methods for getting started with machine learning. A machine learning breakthrough can be found in everything from the technology of a Tesla car to Netflix's recommendation engines to verbal identification on any iPhone.

It is challenging to analyze algorithmic machine learning because the code has multiple implications wherein knowledge might even be inaccurate.

- Strong understanding of coding: You require a solid grasp of sophisticated programming languages like Python, Julia, and others in order to apply machine learning algorithms.

- Advanced knowledge A deep neural network is used in the machine learning subfield of deep learning to create programs capable of carrying out complicated tasks at such a conscious level. You'll require advanced programming abilities in addition to a firm grasp of math and statistics to study deep convolutional neural networks.

- Computerized sharing: When dispersed among a large number of computers during the training phase, machine learning algorithms often expand. You will require a basic understanding of software engineering and cloud computing when you desire to specialize in cloud applications.

- Arduous algorithms: Understanding machine learning algorithms may be challenging, especially for newcomers. Before applying an approach, you must learn all of its many elements. Even though, testing is necessary to determine the best strategy because not all algorithms will be effective given specific data collection or commercial challenges.

- Math abilities: To comprehend machine learning algorithms, it'd be beneficial to be familiar with some of these fundamental mathematical ideas, such as probability, statistics, and linear programming. Owing to its intricacy, understanding those ideas can sometimes be challenging. Additionally, students need to learn how to apply every machine learning concept, that necessitates a deeper understanding of such subjects beyond simply the fundamentals.

1.2 Existing System

Machine learning can be learned through various online courses on platforms such as Coursera[15]. But these systems require the user to learn the syntax of the code first providing a chance to try out the concepts. Some existing code generation platforms are Apple's Create ML[11], Google Cloud's AutoML[12], DataRobot[14], etc. making it easier for users to create ML models without any machine learning knowledge. These platforms have implemented an automated approach to machine learning which doesn't add an understanding of machine learning algorithms, concepts, or code.

1.3 Problem Statement

The aim of the project is to design and develop a web-based application that generates code for various ML algorithms, making it easier to experiment with concepts of machine learning.

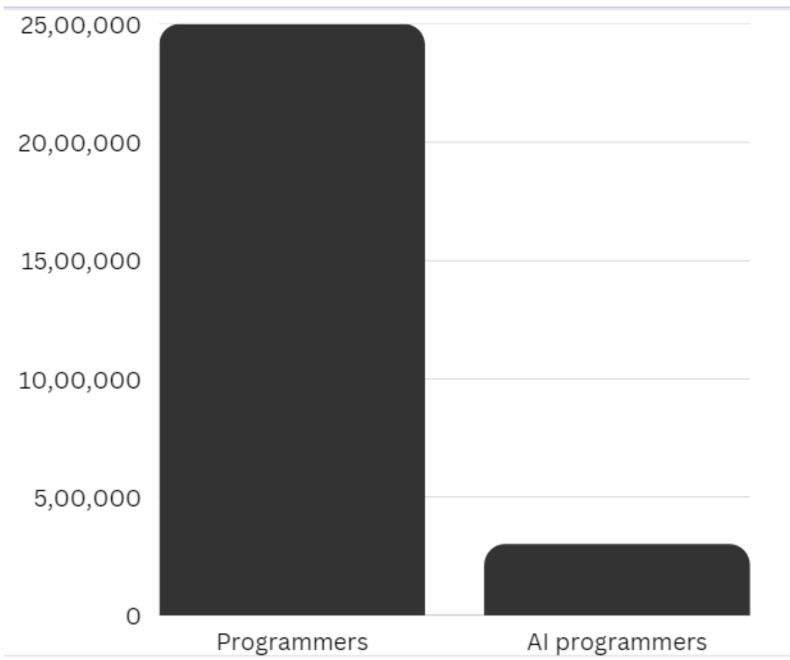


Figure 1.1: Graph showing the number of programmers in the world compared to the number of AI/ML developers

The demand for AI developers is surging at an unprecedented rate, far surpassing the current supply in the job market. While many programmers possess a basic understanding of AI concepts, a significant number opt not to explore and utilize powerful AI/ML frameworks like sklearn, TensorFlow, and others. This creates a critical gap between the potential of AI technology and its actual implementation. AI/ML developers play a crucial role in bridging this divide by harnessing the capabilities of these frameworks to build intelligent, data-driven applications that can revolutionize industries and enhance human experiences. Their expertise is instrumental in designing and deploying cutting-edge algorithms, predictive models, and neural networks to extract valuable insights from vast datasets, drive decision-making processes, and automate complex tasks. As industries increasingly rely on AI-driven solutions, the demand for skilled AI/ML developers continues to grow, underlining the pressing need for professionals who can leverage these frameworks to unlock the full potential of artificial intelligence and machine learning in our rapidly evolving world.

As industries embrace digital transformation and seek competitive advantages, the demand for machine learning (ML) continues to soar. However, the perceived complexity of ML programming, the lengthy code required, and the intricate syntax involved make programmers hesitate to enter into the ML domain. Furthermore, the majority of existing ML learning platforms assume a certain level of coding knowledge, which acts as a barrier for novice programmers. This requirement for extensive coding expertise limits the accessibility of ML learning and prevents aspiring programmers from harnessing the potential of ML techniques in their work.

To address this challenge, there is a need for user-friendly ML learning platforms that cater to the needs of novice programmers. These platforms should simplify the complexity of ML programming, automate the generation of code for various ML algorithms, and provide intuitive interfaces that require minimal coding knowledge. By reducing entry barriers and making ML more accessible, such platforms can empower novice programmers to leverage ML techniques effectively and foster innovation in various industries.

1.4 Objectives

The objective is to simplify the process of experimenting with ML concepts by automating the generation of code, thereby reducing the need for manual coding and streamlining the learning and implementation process.

The proposed application should possess the following key features:

- **Algorithm Selection**- The application should offer a comprehensive library of ML algorithms, encompassing popular models such as linear regression, decision trees, support vector machines, neural networks, and more. Users should be able to select algorithms based on their specific requirements and problem domains.
- **Customization Options** -The application should provide flexibility for users to customize the generated code. This includes specifying input data sources, feature selection techniques, hyperparameter settings, and evaluation metrics. By allowing customization, users can tailor the code to their specific use cases and gain a deeper understanding of the underlying ML concepts.
- **Code Generation** -The core functionality of the application is to automatically generate code based on the selected ML algorithm and customization options. The generated code should be clear, well-documented, and compatible with commonly used programming languages, such as Python, to ensure ease of integration and execution.
- **Interactive Environment** - The application should offer an interactive environment where users can execute the generated code and observe the results in real time. This allows users to experiment with different algorithms, parameters, and datasets, gaining hands-on experience with ML concepts without the need for extensive coding knowledge.
- **Educational Resources**- To aid beginners in understanding ML concepts, the application should provide educational resources such as explanations of algorithmic principles and their respective parameters. These resources can help users grasp the fundamental concepts behind each ML algorithm and guide them in utilizing the generated code effectively.

1.5 Scope

A no-code ML learning platform has the opportunity to revolutionize the way individuals, including non-technical professionals, approach and leverage ML techniques.

One of the key aspects of such a platform is its ability to enable rapid prototyping and experimentation. Users can quickly generate code for different ML algorithms, allowing them to iterate, refine, and experiment with various models.

Beyond its practical applications, the no-code ML learning platform also serves as an educational tool. It provides insights into the inner workings of ML algorithms, allowing users to gain a better understanding of the generated code, the underlying principles of ML, and the impact of different algorithmic choices. This promotes learning and empowers users to make informed decisions when applying ML techniques

This platform can bridge the gap between ML algorithms and individuals without extensive coding expertise. It enables professionals from various domains to easily explore and apply ML techniques without the need for deep programming knowledge. This expands the potential user base and encourages a wider range of individuals to adopt ML in their work.

Chapter 2

Literature Review

2.1 'Using No-code AI to Teach Machine Learning in Higher Education' by Leif Sundberg and Jonny Holmström

As ML is commonly associated with technical professions, such as computer science and engineering, incorporating training in the use of ML into non-technical educational programs, such as social sciences courses, is challenging. This paper [1] presents an approach to address this challenge by using no-code AI in a course for students with diverse educational backgrounds. The approach was tested in an empirical, case-based educational setting, in which students engaged in data collection and trained ML models using a no-code AI platform. In addition, a framework consisting of five principles of instruction (problem-centered learning, activation, demonstration, application, and integration) was applied. This paper contributes to the literature on IS education by providing information for instructors on how to incorporate no-code AI in their courses, and insights into the benefits and challenges of using no-code AI tools to support the ML workflow in educational settings.

No-code solutions for software development have been subject to previous research as they enable non-programmers with little or no coding experience to produce various applications. As noted by Sundberg and Holmström, a new generation of 'lightweight' no-code AI platforms—also known as AI as a service or simply AI service platforms—enables non-data scientists to train ML models to make predictions. For example, new 'drag-and-drop' interfaces enable anyone to develop, train and test AI algorithms in a few hours.

As noted by Holmström, rapid technological developments create challenges for maintaining up-to-date curricula for educating professionals who will work in environments with high levels of technology. As AI is being increasingly adopted in diverse domains, most, if not all, professionals will engage with or be affected by intelligent systems in their

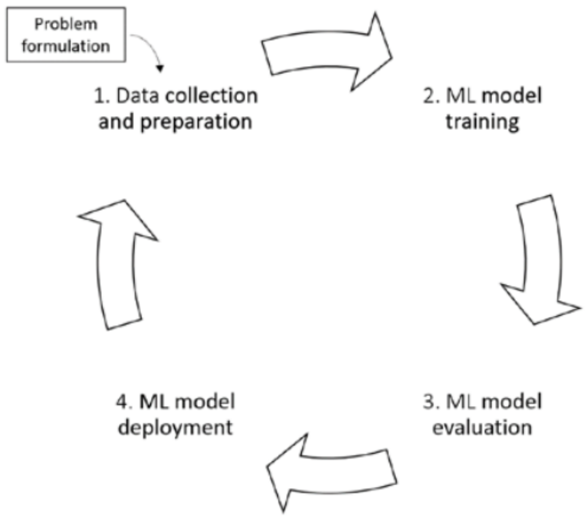


Figure 2.1: The general ML workflow [1]

careers. However, as mentioned, AI is associated with the need to understand algorithms and hence skills rooted in computer science and engineering. This poses challenges for professionals rooted in other disciplines, not because they have nothing to contribute to AI or gain from its use, but because of a lack of fundamental knowledge of how, for example, an ML system works. A potential remedy, also already mentioned, is to use 'lightweight' AI in the form of AI service platforms, which are easy to use with little to no installation requirements (as they are cloud-based) and have graphical interfaces that help users to train ML models.

Hence, in this paper, we pose two research questions (RQs):

- RQ1: How can no-code AI be used to teach ML in non-technical educational programs?
- RQ2: What are the benefits and challenges of using no-code AI in education?

The paper describes a course module on using no-code AI platforms to teach machine learning in higher education. The module was conducted over a little over a month, including Christmas holiday breaks, and started with a 3-hour workshop session that introduced machine learning and demonstrated the functionalities of the no-code AI platform. The module emphasized supervised learning and focused on providing students with hands-on experience.

Principles	Description
Problem centered learning	Humans learn better when they are solving problems, so learning is promoted when learners acquire skills in contexts of real-world problems.
Activation	Learning is promoted when learners activate existing knowledge and skills as foundations for a new skill. An important step here is to start at the learner's level. Activation requires learning activities that stimulate the development of mental models and schemes that can help learners to incorporate new knowledge or skill into their existing knowledge framework.
Demonstration	Learning is promoted when learners observe a demonstration of the skill to be learned, e.g., by exposure to examples of good and bad practice.
Application	Learning is promoted when learners apply new skills they have acquired to solve problems. Applying new knowledge or skills to real-world problems is treated as almost essential for effective learning.
Integration	Learning is promoted when learners reflect on, discuss, and defend knowledge or skill they have acquired. The effectiveness of a course is enhanced when learners are provided opportunities to discuss and reflect on what they have learned in order to revise, synthesize, recombine and modify their new knowledge or skills.

Figure 2.2: Principles of educational approach[1]

The workshop covered topics such as the current status of ML, increases in dataset scales, and ML applications. It introduced the differences between supervised, unsupervised, and reinforced ML, and emphasized the importance of data labeling. The lecture provided a checklist for determining the suitability of ML applications and outlined various ML problems and use cases.

After the theoretical part, the workshop demonstrated the ML workflow using the no-code AI platform. It covered data collection, preparation, model training, evaluation, and deployment. Students were then assigned a problem-centered task to help a fictional company, "WeldCorp," use ML for quality assurance of welding joints.

The students were divided into two groups and provided access to the no-code AI platform. They were encouraged to collect and upload data for training their ML models. The students engaged in data processing, training, and evaluation. During a Q and A session, they discussed their queries and challenges related to data collection and model evaluation.

The final seminar, held via Zoom, allowed students to present their results and propose operationalization ideas for their work. The teachers played a facilitating role during the seminar, encouraging student reflections on the ML process.

The teaching activities in the module were aligned with five instruction principles: problem-centered learning, activation, demonstration, application, and integration. The students received pass or fail grades based on the coherence of their suggestions for WeldCorp and the formulation of results and discussions.

Principles	Description
Problem centered learning	The students were presented with a case of a welding company, WeldCorp, seeking to expand and scale up its business while improving quality control. To help these efforts they were encouraged to apply ML to differentiate between good and bad weld points.
Activation	Since the students had diverse educational backgrounds (business and administration, computer science, and behavioural science), we chose to use a no-code AI platform. This enabled them to incorporate previous skills and work during the course, even if they lacked previous experience of data science.
Demonstration	We showed the students several examples of ways to train ML models via the no-code AI platform. Students were encouraged to take tutorials and experiment with different types of open datasets (e.g., table-, text-, and image-based), and problems that can be accessed through the platform.
Application	The students were divided into two groups and each student was given access to an enterprise account enabling them to use the no-code AI platform to address a new type of problem by applying the previously demonstrated procedures.
Integration	Students were encouraged to reflect on their learning during the final seminar in both a survey and the course evaluation. During the final seminar they were also expected to learn from each other by preparing questions for the other group.

Figure 2.3: Activities that the students engaged in, linked to the five principles of instruction[1]

ML Workflow	Role of no-code AI
Data collection / preparation	Provision of a graphical interface for visualization, uploading and processing data.
Model training	Access to a portfolio of pre-trained models, tutorials and datasets, as well as automatic selection and fine-tuning of one or more algorithm(s) for training.
Evaluation	Visual interface for evaluating and comparing the performance of models (e.g., through ROC curve- and confusion matrix-based analyses).
Deployment	API-interfaces with complementary plugins to aid integration in organizational settings

Figure 2.4: Ways that no-code AI can facilitate learning about ML[1]

Benefits of Using No-Code AI in Education:

1. Visualization of data and provision of a graphical interface for uploading data.
2. Access to a portfolio of pre-trained models, tutorials and datasets, as well as automatic selection and fine-tuning of algorithm(s) for training.
3. Visual interface for evaluating and comparing the performance of models (e.g., through ROC curve- and confusion matrix-based analyses).

The paper answered RQ1 by proposing a problem-centered approach to using no-code AI in higher education, with instruction to teachers. Regarding RQ2, the course shows how no-code AI can help to guide students through the ML workflow (data processing, model training, evaluation and deployment), and present important challenges (ML case construction, platform selection and user management, and student group composition) that was encountered during the course.

2.2 'Study of Machine Learning No-Code Platform' by Kartikey Ahlawat

This paper [2] discusses the development and implementation of ML Modeller, a web application based on a no-code platform aimed at democratizing machine learning. The paper highlights the significance of no-code platforms in the current technological landscape and its potential to empower individuals with no coding knowledge.

ML Modeller provides a user-friendly interface that allows users to upload their datasets effortlessly. Once the data is uploaded, the platform automatically trains various machine learning models, including classification and regression algorithms, on the provided data. Users can then obtain accurate predictions based on their input dataset, eliminating the need for manual data analysis and coding efforts. ML Modeller is invented to solve:

1. Saving time: Saves time, as users do not have to write code.

2. Data: The website is not storing any user data.

3. No need to have programming knowledge as the process is automated, you only have to check predictions and metrics for evaluation (MSE, RMSE, and MAE) and accuracy.

4. No need to know how to code as it is done at the backend for you.

5. No need to have prior knowledge about ML models.

6. No need to go through data manually, as it might give false insight and eventually result in making bad decisions.

The paper discusses the potential applications of ML Modeller across various professions and industries. For educators, the platform can streamline tasks such as classroom management, grading, and predictive analytics for student outcomes. In healthcare, ML Modeller can assist doctors in diagnosing diseases and predicting patient outcomes based on electronic medical records (EMR) data. Recruiting and human resources professionals can benefit from the platform's ability to identify the best candidates and optimize payroll decisions.

The performance evaluation of ML Modeller reveals its impressive growth in web traffic and user engagement. The application has garnered interest from both existing and new users, indicating its potential to meet the demand for easy-to-use and accurate machine learning predictions.

Looking into the future, the paper discusses the possibility of incorporating a deep learning module into ML Modeller. While deep learning offers remarkable predictive capabilities, it currently requires substantial computational resources, making it impractical for budget-friendly applications.

2.3 'Use case of no code machine learning tools for medical image classification' by Kalshetty A., & Rakshit S

This paper [3] shows how AI tools have brought about significant paradigm shifts in the field of medical imaging, holding immense promise for enhancing diagnostic capabilities and elevating patient care. Despite this potential, the development and validation of AI tools for clinical use have not been uniformly accessible, leading to uneven distribution and adoption by physicians and radiologists, who are the primary end-users of such software solutions. As the medical community continues to push towards the democratization of AI, no-code machine learning (ML) tools have emerged as versatile and user-friendly alternatives, offering a pathway to bridge the gap. However, despite their potential to revolutionize medical imaging tasks, the utilization of no-code ML tools in this domain remains largely under-explored.

In light of this context, the purpose of this proof-of-concept study was to investigate the feasibility and effectiveness of deploying no-code ML tools, specifically Teachable Machine, for a fundamental medical image classification task. By focusing on simplicity and ease of use, the study sought to evaluate whether such tools could serve as viable options to empower medical professionals in leveraging AI for diagnostic purposes. As medical imaging technology continues to evolve, understanding the capabilities and limitations of no-code ML tools in this domain becomes increasingly crucial for unlocking their true potential and ensuring broader access to advanced AI solutions in healthcare settings.

2.4 'A Novel Browser-based No-code Machine Learning Application Development Tool' by Ozan E.

This paper[4] presents a browser-based machine learning (ML) application development tool aimed at researchers with limited programming knowledge. Traditionally, ML development demands significant technical expertise, creating barriers for researchers in various fields. The tool allows users to create customized image classifier models without the need for software installations or manual coding. By offering a no-code workflow, researchers can develop and test ML models directly in their browsers, making ML more accessible and efficient for their specific research needs.

2.5 'Coding as a Playground: Promoting Positive Learning Experiences in Childhood Classrooms' by Bers M., Gonz alez-Gonz alez C., & Armas-Torres M.

This paper [5] presents a study that evaluates a "coding as a playground" experience with the KIBO robotics kit in three Spanish early childhood centers. The aim is to integrate coding and computational thinking into formal curriculums for young children aged 3 to 5 years old. The researchers employed a combination of qualitative and quantitative methods to study the impact of this educational intervention. The results indicate that teaching coding and computational thinking at an early age is feasible, and the strategies used during the experience promote communication, collaboration, and creativity in the classroom. Teachers exhibited autonomy and confidence in integrating coding into various curricular areas, such as art, music, and social studies. Moreover, the study highlights the importance of project-based learning to effectively introduce coding and computational thinking in early childhood classrooms.

The study findings indicate that the "coding as a playground" approach with KIBO robotics can successfully foster computational thinking and coding skills in young children, regardless of their socio-economic backgrounds. Teachers played a crucial role in customizing the curriculum to suit their classrooms' needs and integrating coding concepts with other subjects. However, some challenges were encountered, such as logistical difficulties in implementing certain assessment methods and issues with using the KIBO robot. Despite these limitations, the research offers valuable insights into the positive implications of early coding education and calls for further research to explore individual adaptations of curriculums and cross-cultural comparisons to identify best practices in introducing coding and computational thinking in early childhood education.

In conclusion, this study provides compelling evidence that early integration of coding and computational thinking using robotics can be effective in early childhood education. By leveraging the "coding as a playground" approach, young children can develop essential coding skills and computational thinking abilities. The findings emphasize the crucial role of teachers in designing tailored curriculum and connecting coding concepts with other subjects, promoting a holistic learning experience. However, further research is needed to address logistical challenges and assess the impact of teachers' pedagogical strategies on children's learning outcomes. Overall, this research contributes to the growing body of knowledge on introducing coding and computational thinking in early childhood classrooms, making strides toward a technologically literate generation with enhanced creativity and problem-solving skills.

2.6 'Budding Novel Applications in agriculture Victimization data processing' by Lekhaa T., Aruna A., & Malarmathi M.

This paper [6] introduces WEKA (Waikato Environment for Knowledge Analysis), a comprehensive tool for various machine learning tasks, particularly data processing assignments. WEKA offers a wide range of machine-learning algorithms that can be directly applied to datasets or invoked from Java code. The paper also discusses a proposed approach for analyzing learning models and highlights the support provided by the rail tool for this model. rail, within WEKA, provides a set of tools for tasks such as data pre-processing, clustering, regression, association rules, and visualization. Additionally, it is suitable for developing new machine-learning algorithms.

WEKA serves as a powerful and versatile platform for researchers and developers engaged in machine learning projects. Its extensive collection of algorithms and integration with Java code enable seamless data analysis and model development. The rail tool, in particular, enhances the utility of WEKA by facilitating various essential tasks, including data pre-processing and advanced data mining techniques. Researchers can leverage these functionalities to explore, evaluate, and optimize learning models, promoting innovation and discovery in the field of machine learning.

WEKA, with its wide array of machine learning algorithms and flexible integration with Java code, stands as a valuable resource for data analysis and model development. The inclusion of rail further enhances the capabilities of WEKA, making it a comprehensive platform for researchers and developers to tackle diverse machine-learning tasks effectively. By empowering users to perform data pre-processing, clustering, association rule mining, and more, WEKA facilitates the development of new machine-learning solutions and encourages advancements in the field.

2.7 'Intrusion Detection System: An Automatic Machine Learning Algorithms Using Auto- WEKA' by Samawi V., Yousif S., & Al-Saidi N.

This paper [7] focuses on developing an Intrusion Detection System (IDS) using Automated Machine Learning (AutoML) to enhance Network Intrusion Detection Systems (NIDS) accuracy and reduce false alarms. Two machine learning software tools, Weka and RapidMiner, are utilized to create the proposed model. Four different classifiers, namely Naïve Bayes (NB), Multilayer Perceptron (MLP), Random Forest (RF), and Sequential Minimal Optimization (SMO), are applied to the intrusion detection dataset to study their performance. Auto-WEKA is employed to automatically select the best classifier with optimal hyper-parameters, while Auto-Model in RapidMiner implements the chosen classifier for evaluating accuracy and ease of use for non-expert developers.

Using the NLS-KDD dataset, the experimental results reveal that Random Forest (RF) classifier demonstrates superior accuracy and acceptable time consumption compared to other classifiers. Furthermore, Auto-WEKA proves to be the preferred tool as it automatically selects the best classifier with suitable hyperparameters, minimizing the effort required by non-expert developers. The combination of AutoML tools, Weka, and RapidMiner, showcases the promising potential for efficient and accurate Intrusion Detection Systems, essential for maintaining network security.

2.8 'Evaluating Animations as Student Aids in Learning Computer Algorithms' by Byrne M., Catrambone R., & Stasko J.

This paper [8] utilized two different algorithms, namely depth-first search and binomial heaps, and involved two distinct subject populations: students with little to no computer science background and computer science majors. The primary focus was to determine whether animations aided students in acquiring both procedural and conceptual knowledge about the algorithms. The findings indicated that animations might enhance learning procedural knowledge by encouraging learners to predict the algorithm's behavior. However, it was also discovered that a similar learning improvement occurred when learners made predictions based on static diagrams, suggesting that prediction, rather than the animation itself, played a significant role in facilitating learning in these experiments.

Although the initial experiments shed light on the possible benefits of algorithm animations and prediction in the learning process, they also highlighted certain methodological issues that require further investigation in future studies. An in-depth exploration of the relationship between animation and prediction is necessary to fully comprehend their impact on learning outcomes. Additionally, researchers should explore other potential benefits of animations in the context of algorithm learning. By addressing these methodological challenges and conducting more systematic experiments, a comprehensive understanding of the role of animations in algorithm learning can be achieved, providing valuable insights for educational practices.

The two experiments conducted to assess the impact of algorithm animations on learning highlighted the potential significance of prediction in aiding students' acquisition of procedural knowledge. While animations seemed to encourage prediction and enhance learning, the results also indicated that prediction could be similarly effective when based on static diagrams. Therefore, the connection between animation and prediction needs further examination. Despite these initial findings, additional methodological considerations must be taken into account for future research to explore other possible benefits of animations in algorithm learning. By systematically addressing these issues, educators and researchers can better utilize animations to optimize algorithm learning experiences and enhance educational outcomes.

2.9 'Automating the Creation of Machine Learning Algorithms using basic Math' by Marrapu B., Raju K., Chowdary M., Vempati H., & Praveen S.

This paper [9] discusses the field of machine learning that has seen significant advancements in model structures and learning methodologies. One notable development is Auto ML, a domain that seeks to automate the construction of machine learning algorithms by utilizing basic mathematical operations as building blocks. However, this progress has predominantly focused on the architecture of deep neural networks. By introducing this novel framework, there is an effort to minimize human intervention through the utilization of a generic search space. A case in point is neural architecture search, where complex layers are employed to automatically design neural networks, presenting modest achievements in creating ML algorithms from scratch.

The emergence of Auto ML and its application in neural architecture search mark a promising new avenue for machine learning research. By automating the process of building machine learning algorithms, researchers can explore more extensive search spaces and discover innovative architectures that might not have been feasible with manual design. The focus on deep neural networks signifies the potential for enhanced performance and efficiency in complex learning tasks. Although the achievements in creating algorithms from scratch may be modest, the ongoing developments in Auto ML open up exciting possibilities for advancing the field and pushing the boundaries of machine learning capabilities.

Auto ML's progress in automating the construction of machine learning algorithms, particularly in the context of neural architecture search, holds great promise for the field of machine learning. By reducing human intervention and exploring vast search spaces, researchers can uncover novel architectures that improve performance and efficiency in complex learning tasks. Although the advancements in creating ML algorithms from scratch may be in early stages, they pave the way for further exploration and innovation, offering exciting prospects for the future of machine learning research.

2.10 'An evaluation of GUI and kinesthetic teaching methods for constrained-keyframe skills' by Kurenkov A., Akg un B., & Thomaz A.

This paper [10] discusses a research proposal on a method for teaching robots skills using Keyframe-based Learning from Demonstration. The proposed approach involves using multiple keyframe demonstrations to learn skills as sequences of positional constraints known as c-keyframes. These c-keyframes can then be used for planning skill execution. An interactive GUI (Graphical User Interface) is introduced to display the learned c-keyframes to the teacher, allowing them to modify aspects of the skill after it has been taught or specify a skill directly without providing kinesthetic demonstrations.

The researchers compare three methods of teaching c-keyframe skills:

1. Kinesthetic Teaching: This involves physically demonstrating the skill to the robot using hands-on movements and interactions.

2. GUI Teaching: This method uses the graphical interface to teach the robot skills.

3. K-GUI Teaching: This method combines kinesthetic teaching with GUI editing of the learned skill. First, the skill is demonstrated through kinesthetic teaching, and then the teacher uses the GUI to make adjustments or refinements to the learned skill.

Results from user evaluation indicate that the K-GUI method of teaching is the most preferred among the three methods, while the GUI teaching method is the least preferred. The paper also mentions several use cases of K-GUI teaching, highlighting how the GUI can be utilized to enhance the results of kinesthetic teaching.

2.11 Existing Code Generation Platforms

Google's AutoML

Google's AutoML platform[12] is an innovative and user-friendly tool that empowers developers to build machine learning models effortlessly. AutoML focuses on enabling users to create custom ML models without requiring extensive coding or machine learning expertise.

The platform offers a wide range of pre-built templates, making it easy to get started with various ML tasks, such as image and text classification, object detection, and recommendation systems. Developers can easily upload their datasets, and AutoML handles the training, tuning, and evaluation processes automatically.

The platform's drag-and-drop interface and intuitive workflows streamline the model development process, allowing users to experiment and iterate quickly. It also provides real-time performance feedback and visualizations, enabling users to gain insights into model accuracy and make data-driven decisions.

Apple's Create ML

Apple's Create ML[11] is a cutting-edge machine learning platform designed to simplify the creation of custom ML models for developers and data scientists. Part of Apple's Core ML framework, Create ML empowers users to build powerful models without extensive coding or machine learning expertise.

The platform offers an array of pre-built templates for tasks like image and text recognition, object detection, and natural language processing. Developers can easily upload their data and let Create ML handle the training and optimization processes automatically.

Create ML's user-friendly interface and intuitive workflows streamline the model development process, enabling users to iterate and experiment efficiently. The platform provides real-time feedback on model performance, allowing users to fine-tune their models for optimal results.

Microsoft Azure's Machine Learning Studio

Microsoft Azure Machine Learning Studio[13] is a powerful and user-friendly cloud-based platform that simplifies the process of building and deploying machine learning models. With a no code, drag-and-drop interface, it enables users to create ML models without the need for extensive coding knowledge.

Azure Machine Learning Studio integrates seamlessly with other Azure services, allowing users to leverage the full potential of Microsoft's cloud ecosystem. It also provides capabilities for collaborative model development, version control, and sharing, facilitating teamwork and knowledge sharing within organizations.

By automating various aspects of the machine learning pipeline, Microsoft Azure Machine Learning Studio empowers users to focus on data exploration, experimentation, and decision-making rather than getting caught up in the intricacies of code. This accessibility and ease of use make it a valuable tool for accelerating ML projects and driving data-driven insights across various industries.

DataRobot

DataRobot[14] is a leading automated machine learning platform that empowers organizations to build and deploy sophisticated ML models with ease. It offers a user-friendly interface, making it accessible to both data scientists and non-technical users.

The platform automates the end-to-end ML process, from data preparation to model selection and deployment. DataRobot leverages advanced algorithms to optimize model performance and offers extensive model evaluation tools to ensure accurate predictions. With a vast library of pre-built models and a powerful feature engineering system, DataRobot streamlines the development of ML solutions for various industries, including finance, healthcare, and marketing.

Platform	Target Audience	Integration	User Interface	Customization
Azure Machine Learning Studio	Data scientists, developers in Microsoft Azure	Seamless integration with Azure services	Drag-and-drop interface with pre-built modules	Customization options available for experienced data scientists
AutoML (Google Cloud)	Developers, data scientists using Google Cloud	Seamless integration with Google Cloud	Emphasizes ease of use with pre-built models	Aimed at simplifying model development for users with limited ML expertise
Create ML (Apple)	iOS and macOS developers using Core ML	Integrates with Apple's Core ML framework	User-friendly interface with pre-built templates	Designed for on-device ML processing and user-friendly model development
DataRobot	Organizations across industries	Supports various integrations	Comprehensive user interface for data scientists and business users	Offers extensive customization options for advanced data scientists

Figure 2.5: Platform comparison

While all these platforms aim to simplify machine learning model development, they differ in their target audience, ecosystem integration, user interface, level of customization, and use cases. The choice between these platforms depends on specific project requirements, expertise, and the ecosystem in which the users operate.

2.11.1 Comparison

While traditional online courses on platforms such as Coursera[15] offer valuable learning resources for machine learning, they often require users to learn the syntax of the code first. This coding-centric approach may deter some individuals, especially those without a technical background, from exploring machine learning concepts and applying them to real-world problems.

In contrast, emerging code generation platforms like Apple's Create ML, Google Cloud's AutoML, and DataRobot take a no code or low-code approach to machine learning. These platforms enable users to create ML models without requiring in-depth knowledge of machine learning algorithms or coding expertise. By automating various steps of the machine learning process, such as data preprocessing, feature engineering, and model selection, these platforms streamline the development of ML models.

While these code generation platforms offer convenience and rapid model development, they may not necessarily provide users with a deep understanding of machine learning algorithms, concepts, or coding. As a result, users may lack the ability to customize models or troubleshoot potential issues effectively. The trade-off is that these platforms prioritize ease of use and speed over the educational aspect of learning machine learning from the ground up.

For individuals seeking a quick solution to create ML models without delving into the intricacies of machine learning, code generation platforms can be valuable tools. However, for those interested in building a solid foundation in machine learning and understanding the underlying principles of ML algorithms, conventional code generation platforms alone fall short.

Our application bridges this gap between traditional online courses and code generation platforms by providing a comprehensive learning experience. By combining the educational aspects of online courses with hands-on practical code generation, the platform offers developers a unique opportunity to not only create ML models but also gain a deeper understanding of how different ML algorithms work.

By engaging users in the process of code generation and model development, the platform empowers them to experiment with various ML concepts, fine-tune models, and observe the outcomes in real-time. This experiential learning approach fosters a greater understanding of ML principles and algorithms, allowing developers to grasp the underlying concepts more effectively.

By offering a holistic learning experience, the platform caters to the needs of aspiring data practitioners who seek a thorough understanding of machine learning, making it a valuable tool in democratizing ML education.

Chapter 3

System Analysis

3.1 Hardware Requirements

The following are the hardware requirements to develop Codio platform:

- Processor: Intel Core i5
- Hard Disk: Minimum 100GB
- RAM: Minimum 8GB
- Nvidia GPU

3.2 Software Requirements

The following are the software requirements used in the development of the app:

- Operating System: Windows or Linux
- React.js
- Node.js
- Python
- DynamoDB
- TensorFlow

Chapter 4

System Design & Methodology

4.1 Architecture Diagram

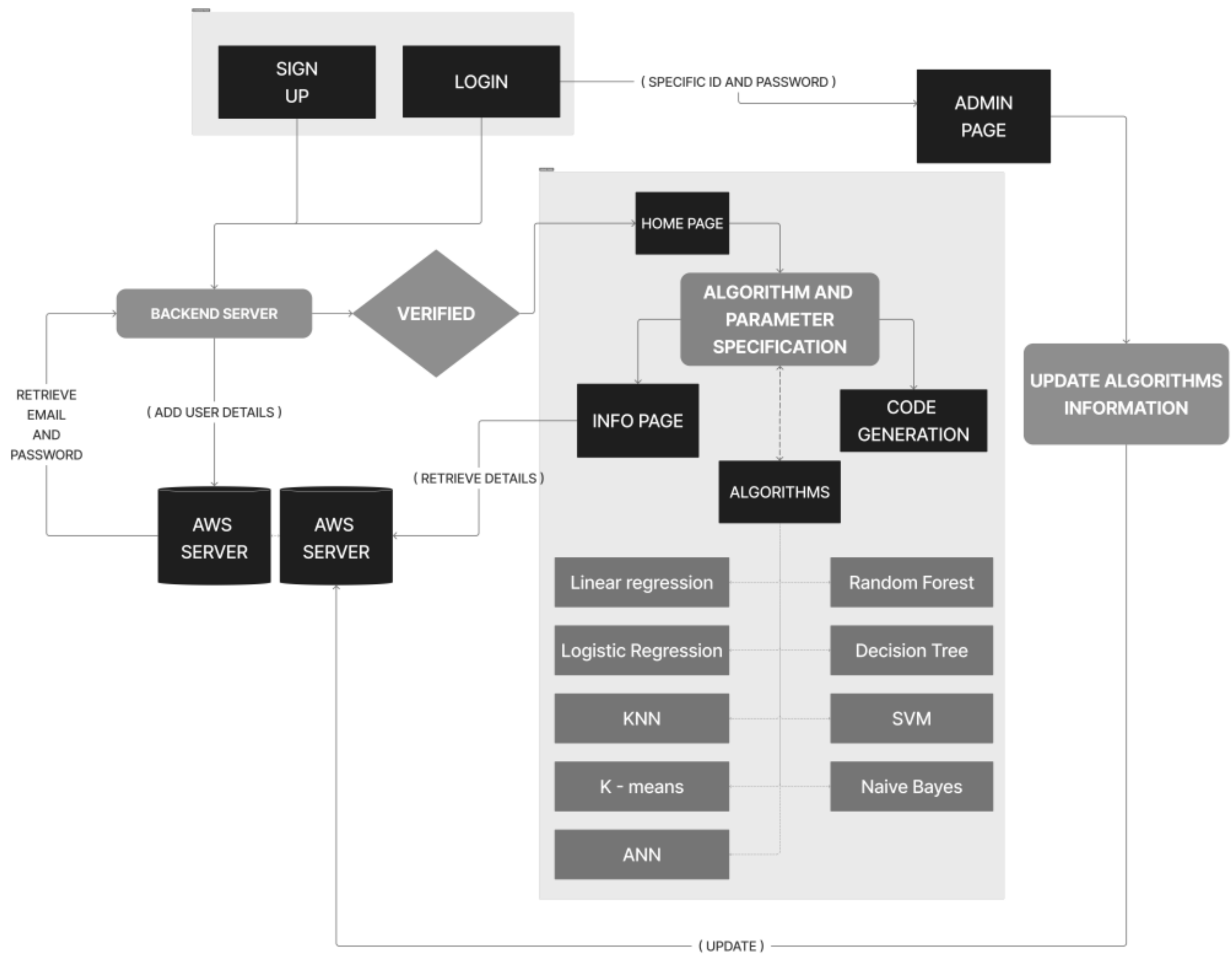


Figure 4.1: Architecture diagram

The platform allows users to register and log in to initiate code generation. Users have the flexibility to input their data path or select from built-in datasets. They can then choose an algorithm from the user-friendly interface and input or select its associated parameters from a menu.

Once the algorithm and parameters are selected, the platform automatically generates the corresponding code along with the machine learning model. Users can conveniently download this code and model by clicking the download button. Additionally, to aid better understanding, users can access information on the algorithms and their parameters.

Furthermore, an admin login grants access to valuable insights such as the number of users and code generation occurrences. The admin can also update algorithm-related information as needed, ensuring the platform stays up-to-date and efficient in serving users' needs.

4.2 Methodology

4.2.1 Login and sign-up

A login and sign-up page has been created where users can either log in if they already have an account or sign up as new users. The user data, including usernames and passwords, are stored in the database. When a user attempts to log in, the system verifies the provided username and password against the records stored in the database. If the credentials match an existing user, the user is authenticated and redirected to the code generation area. However, if the user logging in has admin privileges, the user is redirected to the admin page, granting them access to special features and functions unavailable to regular users.

4.2.2 Input and Preprocessing

Before proceeding with algorithm selection and parameter specification, users have the option to upload their dataset or use a pre-existing dataset from the system. If the data requires preprocessing, the system provides tools to normalize the data for compatibility with the selected algorithm.

4.2.3 ML Algorithms

Users can choose the most suitable algorithm based on their specific task requirements and the nature of their dataset. The system provides brief descriptions and use cases for each algorithm to assist users in making informed decisions.

Linear regression

Assume X as the input feature matrix (independent variables) Assume y as the output vector (dependent variable)

Step 1: Initialize model parameters Initialize random values for coefficients (weights) 'w' and the intercept 'b'

Step 2: Define hyperparameters Choose a learning rate 'alpha' (controls the step size during gradient descent)

Choose the number of iterations 'num _iterations' for training

Step 3: Model Training using Gradient Descent

for i in range(num _iterations):

Step 3.1: Compute the predicted values 'y _pred' using the current model parameters

y _pred = X * w + b

Step 3.2: Compute the loss (error) between predicted and actual values

loss = (1 / (2 * m)) * sum((y _pred - y)$\hat{2}$)

Step 3.3: Compute gradients of the loss for model parameters

dw = (1 / m) * sum(X * (y _pred - y))

db = (1 / m) * sum(y _pred - y)

Step 3.4: Update model parameters using gradient descent

w = w - alpha * dw

b = b - alpha * db

Step 4: Final model parameters The final values of 'w' and 'b' are the learned coefficients and intercept.

Step 5: Model Prediction To make predictions on new data 'X _new': y _pred _new = X _new * w + b

Logistic regression

Assume X as the input feature matrix (independent variables)

Assume y as the binary output vector (0 or 1) (dependent variable)

Step 1: Initialize model parameters Initialize random values for coefficients (weights) 'w' and the intercept 'b'

Step 2: Define hyperparameters Choose a learning rate 'alpha' (controls the step size during gradient descent) Choose the number of iterations 'num _iterations' for training

Step 3: Model Training using Gradient Descent

for i in range(num _iterations):

Step 3.1: Compute the logits (z) using the current model parameters

z = X * w + b

Step 3.2: Apply the sigmoid function to compute the predicted probabilities 'y _pred'

y _pred = sigmoid(z)

Step 3.3: Compute the cross-entropy loss between predicted probabilities and actual values

loss = - (1 / m) * sum(y * log(y _pred) + (1 - y) * log(1 - y _pred))

Step 3.4: Compute gradients of the loss for model parameters

dw = (1 / m) * sum(X * (y _pred - y))

db = (1 / m) * sum(y _pred - y)

Step 3.5: Update model parameters using gradient descent

w = w - alpha * dw

b = b - alpha * db

Step 4: Final model parameters The final values of 'w' and 'b' are the learned coefficients and intercept.

Step 5: Model Prediction

To make predictions on new data 'X _new':

z _new = X _new * w + b

y _pred _new = sigmoid(z _new)

KNN

Assume X_train as the training feature matrix (independent variables)

Assume y_train as the training output vector (dependent variable)

Assume X_test as the test feature matrix (unseen data)

Step 1: Choose the value of 'k'

Choose the number of neighbors 'k' (a positive integer)

Step 2: For each test sample, do the following

for each sample in X_test:

Step 2.1: Calculate the distance to all training samples distances = calculate _distances(X_train, sample)

Step 2.2: Sort the distances in ascending order and get the 'k' nearest neighbors

nearest_neighbors = sort_and_get_nearest(distances, k)

Step 2.3: Determine the class label of the test sample

predicted_label = majority_vote(nearest_neighbors, y_train)

Step 2.4: Store the predicted label for the test sample

Step 3: Model Prediction

Return the predicted labels for all test samples

K Means

Assume X as the input data matrix (data points)

Assume K as the number of clusters to create

Step 1: Initialize cluster centroids

Randomly choose K data points from X as the initial centroids

Step 2: Assign data points to clusters

Repeat until convergence (when no data point changes cluster):

For each data point x in X:

Calculate the distance between x and each centroid

Assign x to the cluster with the nearest centroid

Step 3: Update cluster centroids

For each cluster:

Calculate the mean of the data points in the cluster

Set the cluster centroid to be the calculated mean

Step 4: Repeat Steps 2 and 3 until convergence

Step 5: Final result

The algorithm terminates when the centroids no longer change significantly or after a fixed number of iterations. The final result is a set of K clusters, each with its centroid.

SVM

Assume X as the input feature matrix (independent variables)

Assume y as the binary output vector (dependent variable) with values 1 or -1

Step 1: Data Preparation

Ensure that the input features X and output values y are appropriately preprocessed and standardized if needed.

Step 2: Choose the SVM kernel

Choose a suitable kernel function (e.g., linear, polynomial, radial basis function) for the SVM.

Step 3: Define hyperparameters

Choose the regularization parameter 'C' and other kernel-specific hyperparameters (e.g., degree for polynomial kernel, gamma for RBF kernel).

Step 4: Model Training

Train the SVM model on the training data (X train, y _train) to find the optimal decision boundary.

Step 4.1 (For Kernel SVMs): Precompute kernel matrix (optional)

If using a kernel other than linear, compute the kernel matrix (Gram matrix) K between all data points in X _train.

Step 4.2: Define the optimization problem (dual problem)

Set up the optimization problem to find the Lagrange multipliers (alphas) that maximize the dual objective function subject to constraints.

Step 4.3: Solve the optimization problem

Use numerical optimization techniques (e.g., Sequential Minimal Optimization) to find the optimal values of the Lagrange multipliers.

Step 4.4: Compute the bias term (intercept)

Compute the bias term 'b' using the support vectors and their corresponding alphas.

Step 5: Model Prediction

For each test sample x in X _test:

Step 5.1 (For Kernel SVMs): Compute kernel values (optional) If using a kernel other than linear, compute the kernel values between the test sample x and all support vectors.

Step 5.2: Calculate the decision function (decision score) Calculate the decision function 'f(x)' using the learned alphas, bias 'b,' and the kernel values (if applicable).

Step 5.3: Classify the test sample If f(x) ¿= 0, predict class 1; otherwise, predict class -1.

Step 6: Model Evaluation (optional)

Evaluate the SVM model's performance on the test data (X _test, y _test) using appropriate evaluation metrics.

Decision Tree

Assume X as the input feature matrix (independent variables)
Assume y as the binary output vector (dependent variable) with values 0 or 1
Step 1: Data Preparation
Ensure that the input features X and output values y are preprocessed and standardized if needed.
Step 2: Define hyperparameters
Choose hyperparameters like maximum tree depth, minimum samples per leaf, and others for controlling tree growth.
Step 3: Define the Decision Tree Node Class (optional)
Define a class to represent the nodes of the decision tree, which includes attributes like feature index, threshold value (for continuous features), and child nodes.
Step 4: Tree Construction
Recursively build the decision tree using a recursive function: def build _decision _tree(X, y):

Step 4.1: If the stopping criteria are met, create a leaf node and return
if stopping _criteria _met(X, y): return LeafNode(y)

Step 4.2: Find the best feature and threshold to split the data best _feature, best _threshold = find _best _split(X, y)

Step 4.3: Split the data based on the best feature and threshold X _left, y _left, X _right, y _right = split _data(X, y, best _feature, best _threshold)

Step 4.4: Recursively build left and right subtrees left _subtree = build _decision _tree(X _left, y _left) right _subtree = build _decision _tree(X _right, y _right)

Step 4.5: Create a decision node with the best feature and threshold and link the subtrees return DecisionNode(best _feature, best _threshold, left _subtree, right _subtree)
Step 5: Tree Pruning (optional)
Perform tree pruning to reduce overfitting and improve generalization performance. Step 6: Model Prediction
For each test sample x in X _test:

Step 6.1: Traverse the decision tree to make a prediction

predicted _class = traverse _decision _tree(x, decision _tree _root)

Step 6.2: Assign the predicted class to the test sample

Step 7: Model Evaluation (optional)

Evaluate the decision tree model's performance on the test data (X _test, y _test) using appropriate evaluation metrics.

Random Forest

Assume X as the input feature matrix (independent variables) Assume y as the output vector (dependent variable) with multiple classes or continuous values

Step 1: Data Preparation

Ensure that the input features X and output values y are appropriately preprocessed and standardized if needed.

Step 2: Define hyperparameters

Choose hyperparameters like the number of decision trees 'n _estimators', maximum tree depth, minimum samples per leaf, and others.

Step 3: Random Forest Construction Create 'n _estimators' decision trees using the bootstrapped samples of the original data:

def build _random _forest(X, y, n _estimators):

random _forest = []

for i in range(n _estimators):

Step 3.1: Create a bootstrapped sample (randomly sampled with replacement) X _sample, y _sample = bootstrap _sample(X, y)

Step 3.2: Build a decision tree using the bootstrapped sample decision _tree = build _decision _tree(X _sample, y _sample)

Step 3.3: Add the decision tree to the random forest random _forest.append(decision _tree)

return random _forest

Step 4: Model Prediction For each test sample x in X _test:

Step 4.1: Make predictions using all decision trees in the random forest

predictions = []

for decision_tree in random_forest:

predicted_value = traverse_decision_tree(x, decision _tree)

predictions.append(predicted_value)

Step 4.2: Aggregate predictions to obtain the final prediction final _prediction = aggregation _function(predictions)

Step 4.3: Assign the final prediction to the test sample

Step 5: Model Evaluation (optional) Evaluate the Random Forest model's performance on the test data (X _test, y _test) using appropriate evaluation metrics.

Naive Bayes

Assume X as the input feature matrix (independent variables) representing the document-term matrix Assume y as the output vector (dependent variable) with class labels. Assume V as the vocabulary set containing all unique words in the training data.

Step 1: Data Preparation Ensure that the input feature matrix X and output vector y are appropriately preprocessed and encoded as a document-term matrix.

Step 2: Calculate Class Priors Calculate the prior probabilities of each class: for each class C: prior _probability(C) = count(y == C) / total _samples

Step 3: Calculate Word Probabilities for Each Class Calculate the conditional probabilities of each word given in each class: for each word w in vocabulary V and each class C: word _probability(w — C) = (count(word w occurrences in class C) + 1) / (total _words _in _class _C + —V—)

Step 4: Model Prediction

For each test sample x in X _test:

Step 4.1: Calculate the class likelihoods for each class C For each class C: class _likelihood(C) = prior _probability(C) * (word _probability(w — C) ĉount(w occurrences in x))

Step 4.2: Assign the class with the highest likelihood as the predicted class for the test sample predicted _class = argmax(class _likelihoods)

Step 4.3: Assign the predicted class to the test sample

Step 5: Model Evaluation (optional) Evaluate the Naive Bayes model's performance on the test data (X _test, y _test) using appropriate evaluation metrics.

ANN

Assume X as the input feature matrix (independent variables). Assume y as the output vector (dependent variable) for supervised learning tasks

Step 1: Data Preparation Ensure that the input features X and output values y are appropriately preprocessed and standardized if needed.

Step 2: Define the ANN Architecture Define the number of input units, hidden units, and output units. Randomly initialize the weights and biases for the hidden and output layers.

Step 3: Choose Activation Functions Choose appropriate activation functions for the hidden and output layers (e.g., sigmoid, ReLU, softmax).

Step 4: Define the Learning Rate and Number of Training Epochs Choose a suitable learning rate 'alpha' and the number of training epochs 'num _epochs'.

Step 5: Model Training using Backpropagation

For each epoch in 1 to 'num _epochs':

For each training sample x, y in the dataset:

Step 5.1: Forward Propagation Calculate the weighted sum and apply activation functions to the hidden layer and output layer. Obtain the predicted output y _pred.

Step 5.2: Calculate Loss Calculate the loss between predicted output y _pred and actual output y.

Step 5.3: Backpropagation Update the weights and biases of both the hidden and output layers based on the loss using gradient descent.

Step 6: Model Prediction

For each test sample x _test in X _test:

Step 6.1: Forward Propagation

Calculate the weighted sum and apply activation functions to the hidden layer and output layer. Obtain the predicted output y _pred _test.

Step 6.2: Assign the predicted output to the test sample

Step 7: Model Evaluation (optional) Evaluate the ANN model's performance on the test data (X _test, y _test) using appropriate evaluation metrics.

4.2.4 Parameter Specification

Users choose the most suitable algorithm based on their specific task requirements and the nature of their dataset. Once users have selected an algorithm, they are presented with options to specify parameters that influence the behavior of the chosen algorithm. These parameters vary based on the selected algorithm and often determine its performance and accuracy. For instance, in the case of Linear Regression, users may need to specify parameters like regularization strength or the type of regularization (L1 or L2). In the case of KNN, users will need to input the number of neighbors (K) to consider during classification or regression.

4.2.5 Code Generation

Once the user has chosen the algorithm and specified the relevant parameters, they can generate the code which is a Python script (.py file). The generated code includes all the necessary import statements, data loading, model creation, parameter settings, and training/fitting steps for the chosen algorithm. There is also an option for downloading the code and the processed pickle file as output.

4.2.6 Information Section

The algorithm information page serves as a valuable resource for users in the code generation and parameter specification section. By clicking on the information (i) button, users can access detailed descriptions and explanations of all eight machine learning algorithms retrieved from the database. This page provides users with insights into each algorithm's functionality, and its parameters, aiding them in making informed decisions while choosing the appropriate algorithm for their specific tasks.

4.2.7 Admin

The admin page is a privileged section that provides administrators with access to important statistics and data management functionalities. Administrators can monitor the overall system's performance and make necessary modifications to the information displayed on the algorithm information page. All data relating to the total number of users and the frequency of code generation is stored and managed securely in the web server database.

4.2.8 UML Diagrams

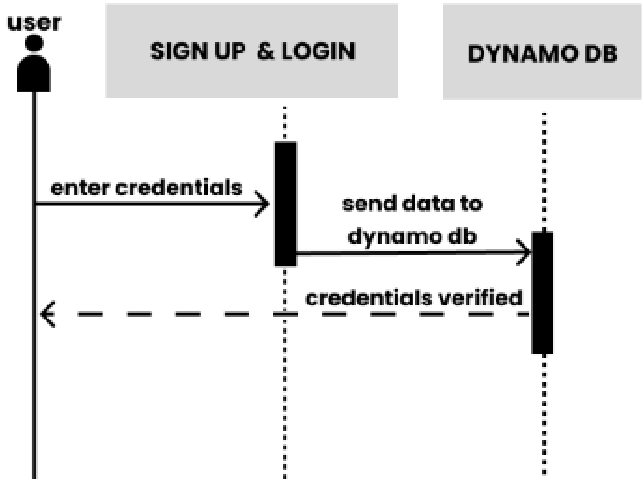


Figure 4.2: Login

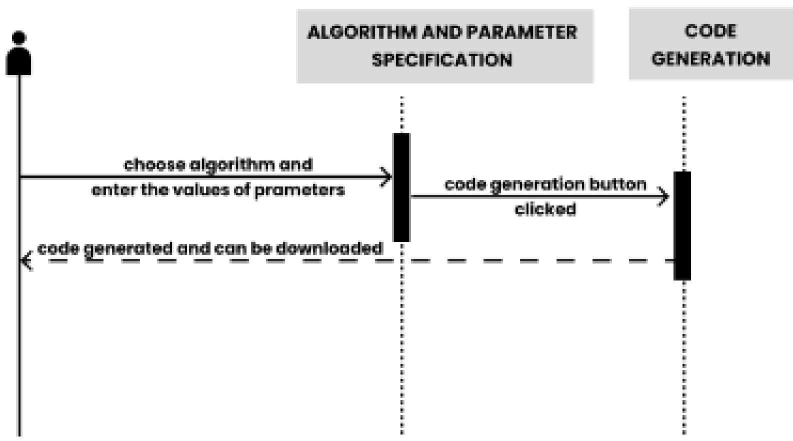


Figure 4.3: Landing page

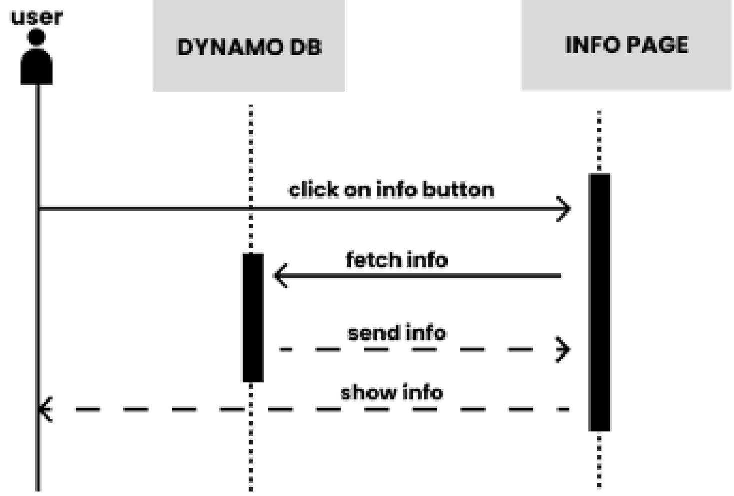


Figure 4.4: Information page

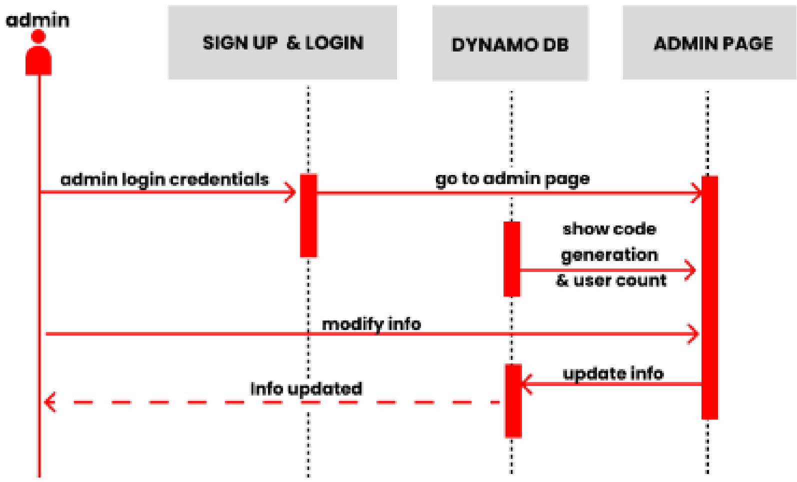


Figure 4.5: Admin

Chapter 5

System Implementation

5.1 Frontend

The frontend development for our webpage is powered by React, a versatile and widely used JavaScript library renowned for its efficiency and reusability in building modern web applications. With React at the core of our frontend architecture, we embark on creating a visually appealing, intuitive, and user-friendly interface that captivates users and fosters seamless interactions. By leveraging the power of React's component-based architecture, we break down the user interface into small, reusable components, such as login and sign-up pages, the algorithm choosing and parameter specification section, the algorithm information page, and the admin page. This modular approach simplifies code organization, promotes reusability, and facilitates efficient maintenance and extensibility as the application evolves.

One of React's key features is its Virtual DOM, a lightweight in-memory representation of the actual browser DOM. This innovative approach allows us to efficiently update only the components that have changed, minimizing rendering overhead and enhancing the webpage's performance. As users interact with our webpage, React diligently handles these interactions, updating the Virtual DOM and performing a diffing process to identify the most efficient way to update the browser DOM, resulting in a smooth and responsive user experience.

In harmony with the frontend, our webpage communicates seamlessly with the backend server through HTTP requests, allowing us to fetch data, perform user authentication, and interact with the DynamoDB database. For making these HTTP requests, we employ Axios, a popular JavaScript library that simplifies the process of sending asynchronous requests from the frontend to the backend. Axios provides a user-friendly and intuitive API, allowing us to make GET, POST, PUT, DELETE, and other types of requests effortlessly, and handle responses efficiently. This integration with Axios ensures a dynamic and data-driven frontend, enabling users to explore various machine learning algorithms and functionalities effortlessly.

5.2 Backend

The backend development for our web application is implemented using Node.js and Express.js, two powerful technologies that enable server-side development. Node.js allows us to run JavaScript code on the server, while Express.js provides a framework that simplifies the setup of the server and the definition of API endpoints. Together, they form a robust platform for handling incoming HTTP requests from the frontend and responding with the necessary data.

The backend communicates with databases such as DynamoDB, a NoSQL database service provided by AWS, to store and retrieve user information, algorithm data, and code generation counts. API endpoints are exposed to the frontend, allowing seamless data exchange through HTTP requests.

Additionally, data processing and code generation functionalities are carried out on the backend, responding to user requests for model training or code execution. When a user requests code generation or model training, the backend forks a child process to run the code separately, ensuring that the main server remains responsive to other requests. The child process performs the necessary computations, and upon completion, the results are sent back to the backend.

Proper error handling and logging mechanisms are implemented to ensure error management and facilitate debugging. Once developed, the backend is deployed on a server or cloud platform to make the application accessible to users.

By leveraging Node.js, Express.js, and DynamoDB, we create a reliable and scalable backend system that seamlessly communicates with the frontend, enabling the delivery of dynamic and feature-rich web applications. The ability to fork child processes ensures efficient code execution, while robust error handling guarantees a smooth user experience. With this well-crafted backend infrastructure, users can confidently explore various machine learning functionalities while enjoying a seamless and responsive web application.

Column	Data type
Algorithm	String
Details	String

Figure 5.1: Information table schema

Column	Data type
email	String
name	String
password	String
Phone number	number
Status	String

Figure 5.2: User data table schema

Chapter 6

Results

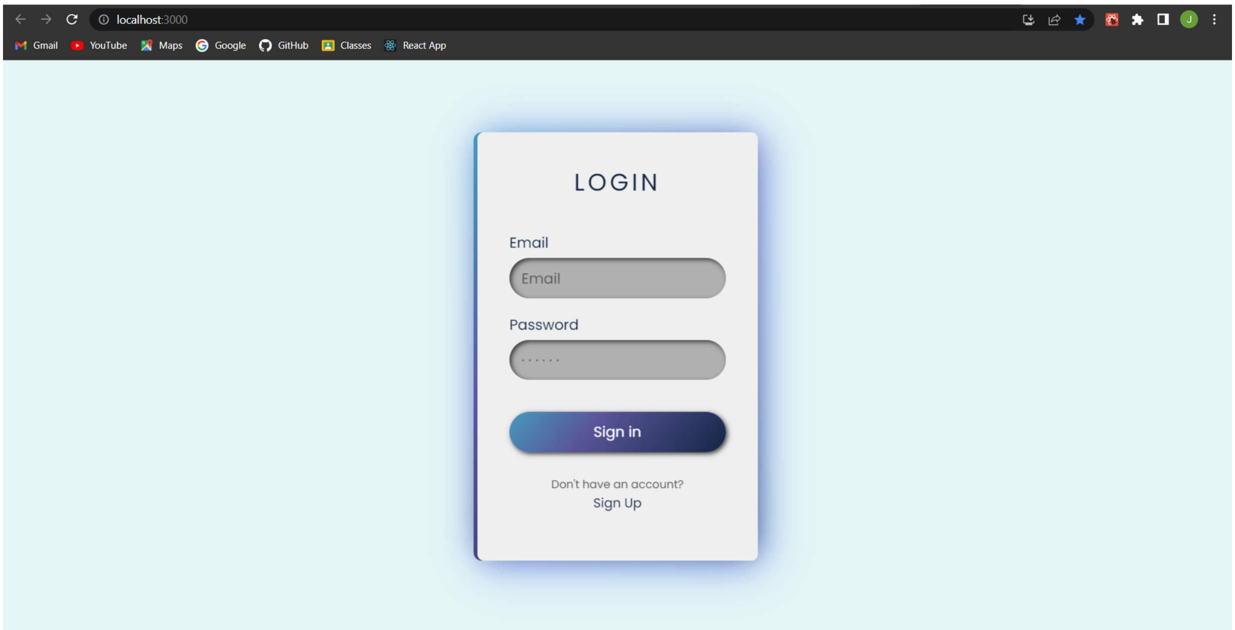


Figure 6.1: Login page

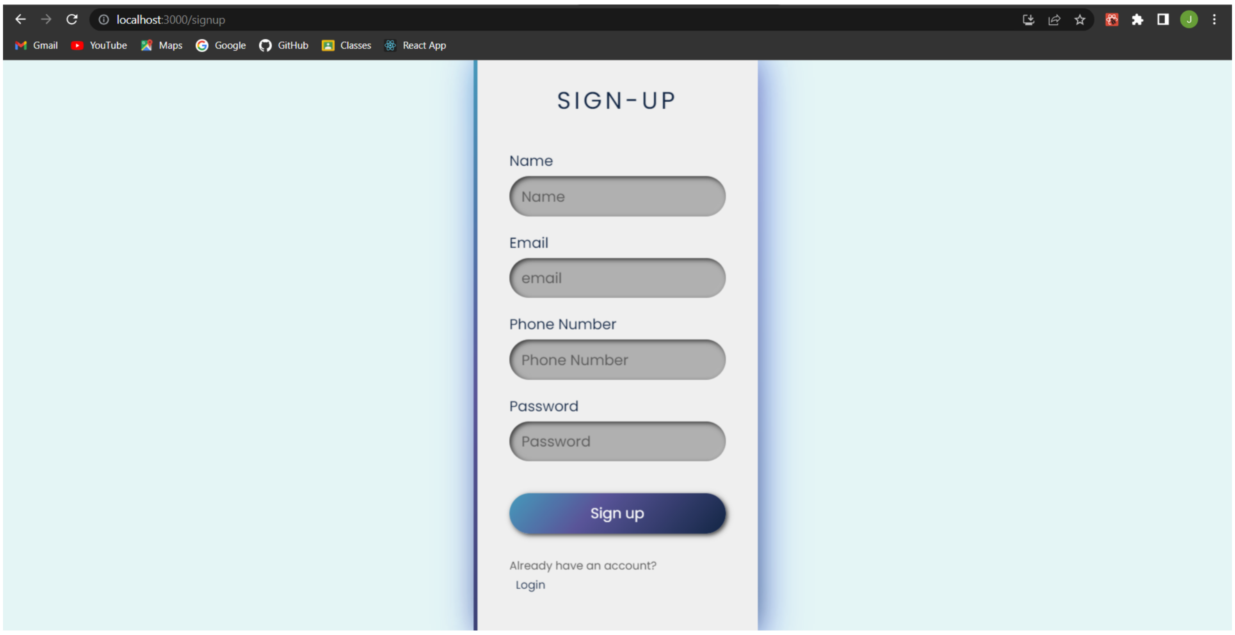


Figure 6.2: Sign-up page

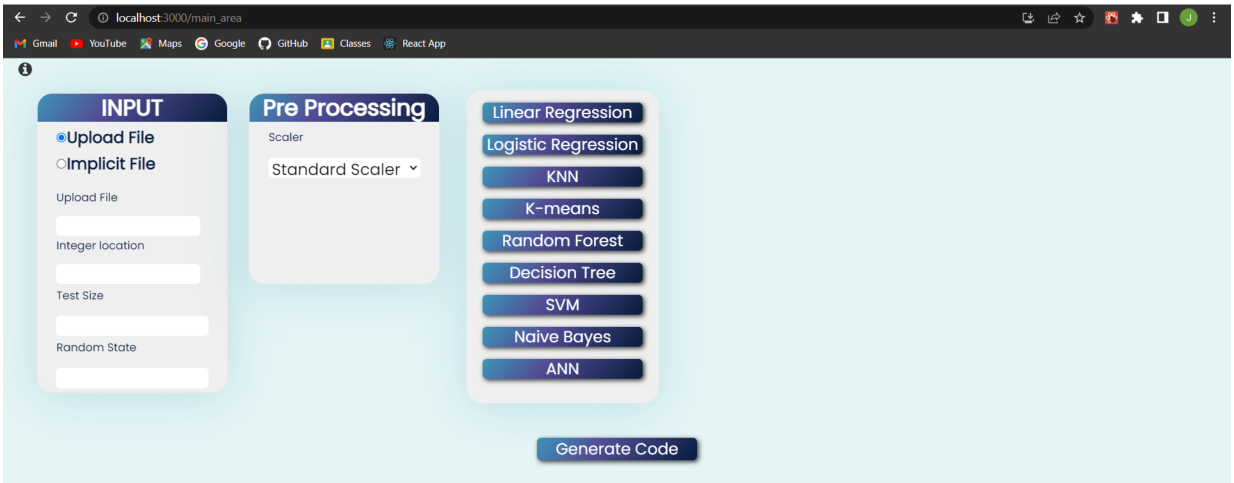


Figure 6.3: Home page

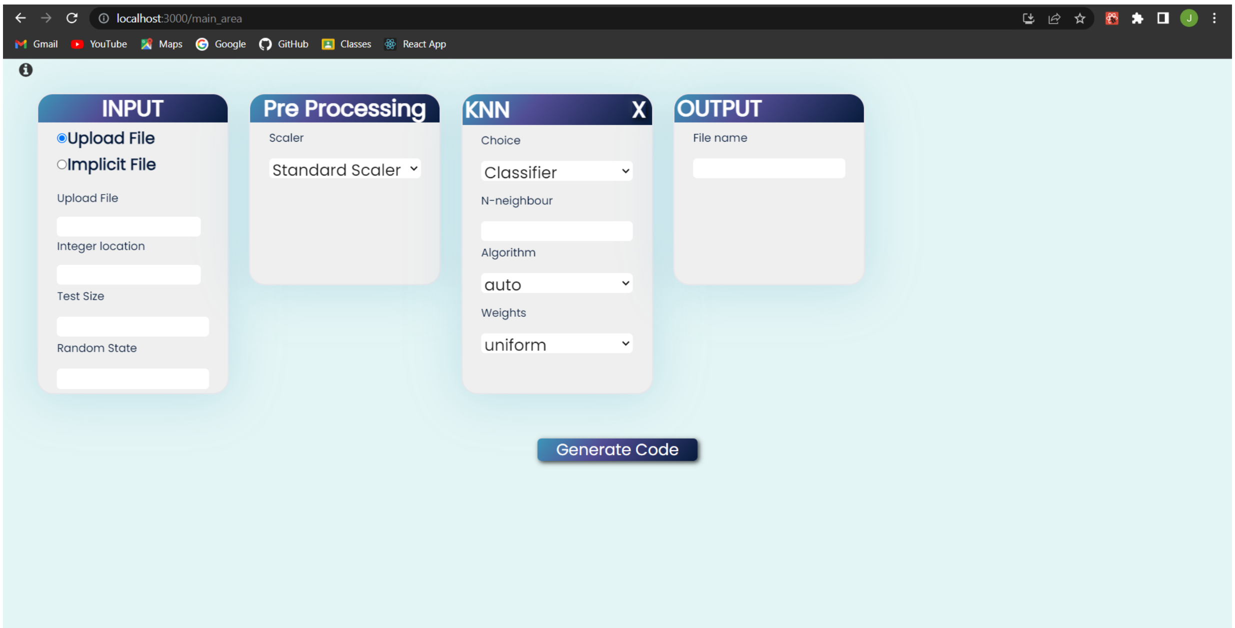


Figure 6.4: Choosing algorithm and its hyperparameters

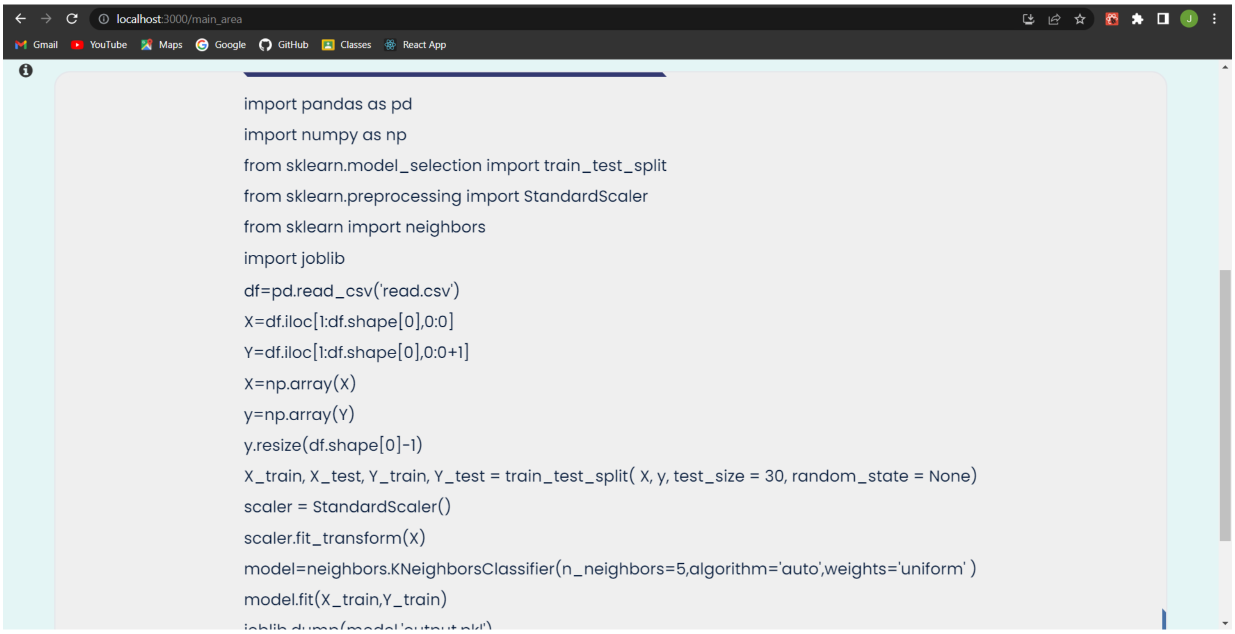


Figure 6.5: Generated code

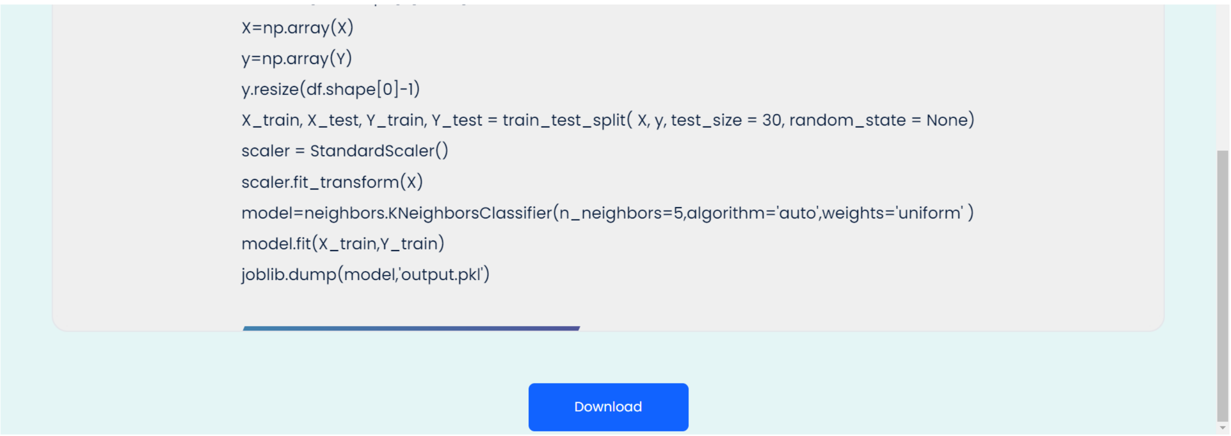


Figure 6.6: Generated code

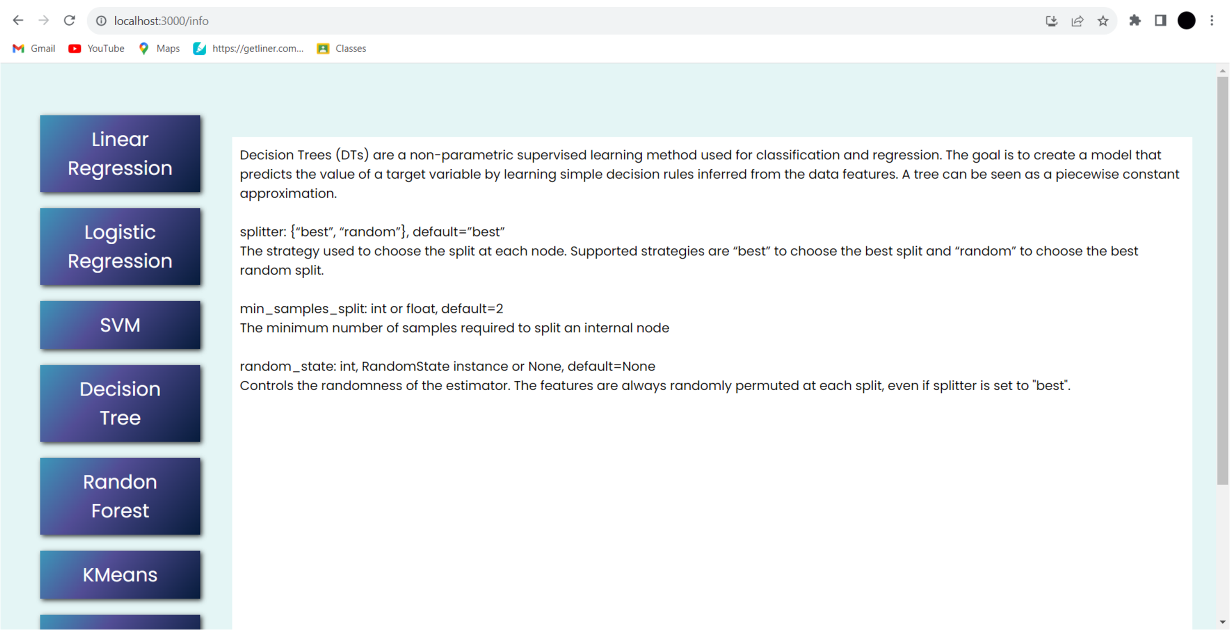


Figure 6.7: Information section

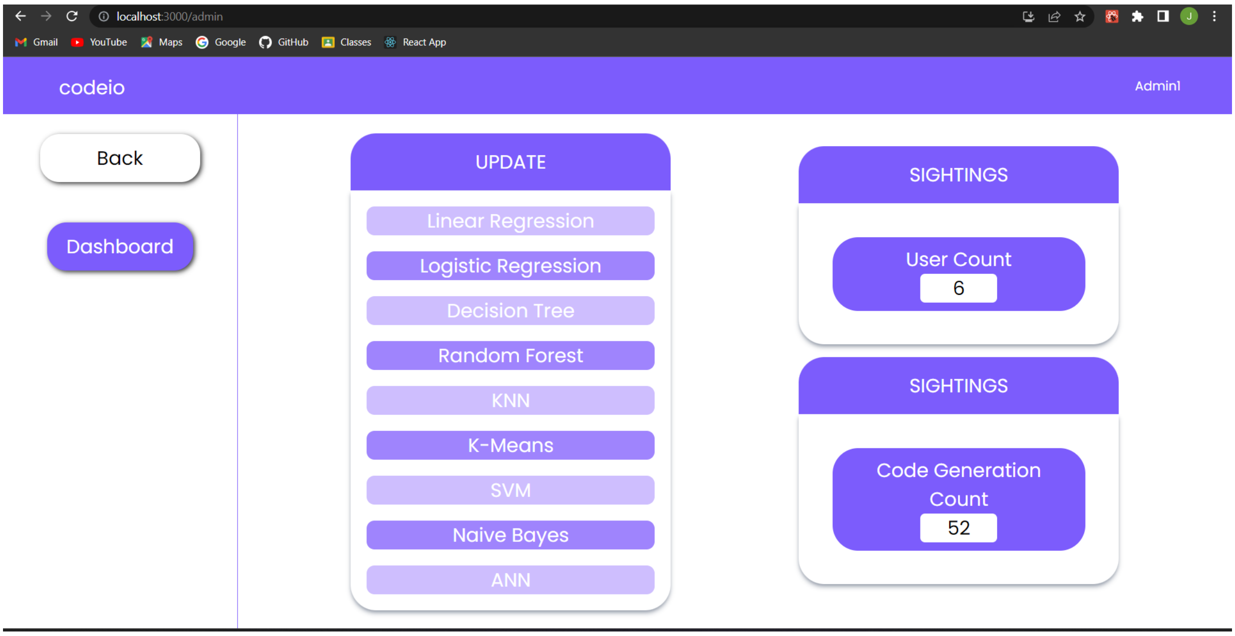


Figure 6.8: Admin page

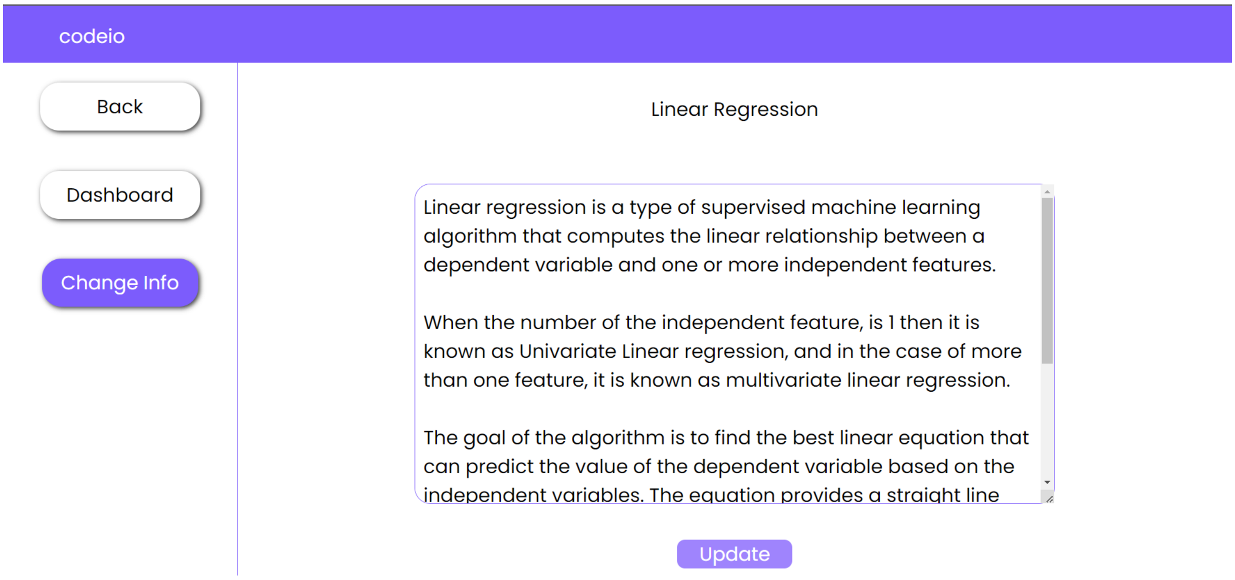


Figure 6.9: Update information section

Chapter 7

Risks and Challenges

1. The user must have a basic knowledge of ML concepts to use the platform.
2. The input dataset cannot be uploaded to the platform.
3. Code is restricted to the parameters on the platform.
4. Admin cannot add new algorithms dynamically.

Chapter 8

Conclusion

We have developed a web based application to give budding developers a solid foundation in understanding ML concepts. The application consists of the following features: algorithm selection, parameter specification, code generation, information section as well as an admin view . All of these features have been verified to be working as intended.

By offering a user-friendly and intuitive platform, users will be able to explore and experiment with various ML techniques without the need to master complicated syntax first. This approach will empower individuals to gain hands-on experience with ML algorithms, fostering a deeper comprehension of the underlying principles and enabling them to apply ML effectively in their projects.

This platform promises exciting opportunities to revolutionize the way individuals approach machine learning experimentation and learning. By expanding the algorithm repertoire, introducing advanced customization options, option to upload the dataset, add more data preprocessing options, and promoting collaboration and educational resources including videos, the application can become a comprehensive ML learning platform for learners and practitioners alike. These enhancements will empower users to gain practical insights, build robust ML models, and contribute to a dynamic community of ML enthusiasts, ultimately democratizing machine learning knowledge and applications.

References

[1] Leif Sundberg, Jonny Holmström, (2024 forth coming). Using No-code AI to Teach Machine Learning in Higher Education. Journal of Information Systems Education (JISE). Vol 35, Issue 1.

[2] Kartikey Ahlawat, 2022. Study of Machine Learning No-Code. International Journal of Multidisciplinary Innovative Research ISSN: 2583-0228 Volume 2, Number 1. Platform pp. 1-14.

[3] Kalshetty, A., & Rakshit, S. (2021). Use case of no code machine learning tools for medical image classification.. . https://doi.org/10.21203/RS.3.RS-498907/V1.

[4] Ozan, E. (2021). A Novel Browser-based No-code Machine Learning Application Development Tool. 2021 IEEE World AI IoT Congress (AIIoT), 0282-0284.

[5] Bers, M., González-González, C., & Armas-Torres, M. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. Comput. Educ., 138, 130-145.

[6] Lekhaa, T., Aruna, A., & Malarmathi, M. (2018). Budding Novel Applications in agriculture Victimization data processing. 2018 International Conference on Soft-computing and Network Security (ICSNS), 1-9.

[7] Samawi, V., Yousif, S., & Al-Saidi, N. (2022). Intrusion Detection System: An Automatic Machine Learning Algorithms Using Auto- WEKA. 2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC), 42-46.

[8] Byrne, M., Catrambone, R., & Stasko, J. (1999). Evaluating animations as student aids in learning computer algorithms. Comput. Educ., 33, 253-278.

[9] Marrapu, B., Raju, K., Chowdary, M., Vempati, H., & Praveen, S. (2022). Automating the Creation of Machine Learning Algorithms using basic Math. 2022 4th

International Conference on Smart Systems and Inventive Technology (ICSSIT), 866-871.

[10] Kurenkov, A., Akgün, B., & Thomaz, A. (2015). An evaluation of GUI and kinesthetic teaching methods for constrained-keyframe skills. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3608-3613.

[11] Create ML Overview - Machine Learning - Apple Developer,https://developer.apple.com/machine-learning/create-ml/, 26th July 2023

[12] Google Cloud AutoML - Train models without ML expertise, https://cloud.google.com/automl, 26th July 2023

[13] Microsoft Machine Learning Studio (classic), https://studio.azureml.net/, 26th July 2023

[14] DataRobot AI Platform - Deliver Value from AI, https://www.datarobot.com/, 26th July 2023

[15] Coursera - Degrees, Certificates,& Free Online Courses, https://www.coursera.org/, 26th July 2023

Appendix A: Base Paper

Using No-code AI to Teach Machine Learning in Higher Education

Preprint · May 2023

2 authors:

Leif Sundberg
Umeå University

Jonny Holmström
Umeå University

Sundberg, L & Holmström, J. (2024, forthcoming). Using No-code AI to Teach Machine Learning in Higher Education. Journal of Information Systems Education (JISE). Vol 35, Issue 1.

Teaching Tip: Using No-code AI to Teach Machine Learning in Higher Education

Leif Sundberg
Jonny Holmström
Department of Informatics
Umeå University
Umeå, 901 87, Sweden
leif.sundberg@umu.se, jonny.holmstrom@umu.se

ABSTRACT

With recent advances in artificial intelligence, machine learning (ML) has been identified as particularly useful for organizations seeking to create value from data. However, as ML is commonly associated with technical professions, such as computer science and engineering, incorporating training in use of ML into non-technical educational programs, such as social sciences courses, is challenging. Here, we present an approach to address this challenge by using no-code AI in a course for students with diverse educational backgrounds. The approach was tested in an empirical, case-based educational setting, in which students engaged in data collection and trained ML models using a no-code AI platform. In addition, a framework consisting of five principles of instruction (problem-centered learning, activation, demonstration, application, and integration) was applied. This paper contributes to the literature on IS education by providing information for instructors on how to incorporate no-code AI in their courses, and insights into the benefits and challenges of using no-code AI tools to support the ML workflow in educational settings.

Keywords: Artificial intelligence, Machine learning, IS education research, Information systems education

1. INTRODUCTION

Machine learning (ML) is a discipline devoted to the construction, application and analysis of computer systems that learn from experience. In a common variant, supervised ML, a system is shown numerous examples of a type of data, e.g., images of, or texts describing certain objects or phenomena, to train it to 'learn' or recognize patterns in them. The system can then use this learning to make predictions about new 'unseen' data, i.e., data that it has not previously encountered (Jordan and Mitchell, 2015; Kühl et al., 2022). Leavitt et al. (2021 pp. 750) define ML as "a broad subset of artificial intelligence, wherein a computer program applies algorithms and statistical models to construct complex patterns of inference within data" (see also, Bishop, 2006).

Massive increases in processing power of digital technology and available data, in combination with better algorithms, e.g., deep learning algorithms (see Lecun et al., 2015) have set the stage for increases in the use of ML in many contexts (Dwivedi et al., 2021). Accordingly, organizations are increasingly deploying intelligent systems that can process large amounts of data, provide knowledge and insights, and operate autonomously (Simsek et al., 2019; Sturm et al., 2021).

As noted by Ma and Siau (2019, p. 1), "Higher education needs to change and evolve quickly and continuously to prepare students for the upheavals in the job market caused by AI, machine learning, and automation." Among other things, these authors argue that AI must be integrated into academic curricula, and not only those of science, technology, engineering, and mathematics (STEM) departments. However, despite abundant research on applications of AI in educational settings (e.g., Luan and Tsai, 2021; Humble and Mozelius, 2022), much less attention has been paid to instruction of students with non-technical backgrounds in ML's practical use and applications (Kayhan, 2022). As ML is commonly associated with technical professions, such as computer science and engineering, incorporating training in its use into non-technical educational programs, such as business- and management-oriented social sciences and Information Systems (IS) programs, is challenging. Similar issues have been raised in previous research on novel intelligent systems (Liebowitz, 1992; 1995) as educators have sought to integrate their use into business and IS programs. Recently, scholars have identified a need to integrate AI curricula in ways that enable students to develop sufficient understanding of technology such as ML to apply it without detailed knowledge of AI algorithms (Chen, 2022). In this paper, we assess 'no-code' AI platforms' potential utility in efforts to meet this need. In contrast to conventional AI systems, which require significant resources for installation and use, these platforms can be readily applied in educational contexts. Thus, they are easy-to-use and affordable forms of AI, and they guide users through the process of developing and deploying AI models, with no need to learn all about the intricacies associated with complex algorithms (Lins et al., 2021; Richardson and Ojeda, 2022). Hence, in this paper, we pose two research questions (RQs):

RQ1*: How can no-code AI be used to teach ML in non-technical educational programs?*

RQ2*: What are the benefits and challenges of using no-code AI in education?*

As already mentioned, 'non-technical' refers here to non-STEM programs, such as business- and management-oriented IS courses. To answer the RQs, we present a teaching tip based on a case study of a master's level AI for business course at Umeå University, Sweden, in which qualitative data were collected through interactions with, and observations of, the students. In the remaining sections of the paper we: summarize previous research on no-code software, describe the educational setting, describe the materials and methods used, present the results, discuss them, and finally offer concluding remarks.

2. BACKGROUND: TOWARDS 'LIGHTWEIGHT' AI

In this section, we present a brief overview of the ML workflow (sub-section 2.1), then summarize literature on the emergence of no-code AI platforms (sub-section 2.2).

2.1 What is Machine Learning?

ML refers to a broad set of AI applications in which computers build models based on patterns they recognize in datasets and use the models to generate hypotheses about the world. Such models have myriads of uses in problem-solving software exploited in industrial and other organizations (Russel and Norvig, 2022). The general ML workflow (see e.g., Chapman et al., 1999; Kelleher and Brendan, 2018; Schröer et al., 2021) begins with creation of a training dataset from which a machine can learn something (Figure 1). Most applications today are based on supervised learning procedures through which a machine learns from labeled data, e.g., text describing an image, such as a photo or drawing of a dog or cat (Fredriksson et al., 2020). Then the training dataset is processed by an algorithm that 'trains' the machine to recognize corresponding patterns. The outcome of this process is a ML model that can be used to make predictions regarding previously unseen data. During the training process, part of a dataset (e.g., 20% of the images in an image classifier case) is reserved for testing the model to avoid problems such as overfitting. Acceptable performance of the model on the test datasets indicates that it may be used to solve problems in real world contexts, such as organizational settings, if the data provide relevant representations of the things or phenomena that must be recognized to solve the problems.

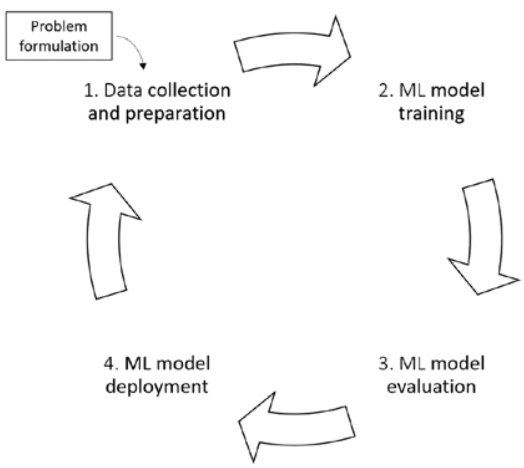


Figure 1. A Simplified ML Workflow

This description is a somewhat simplified version of the ML workflow. In reality, it takes several iterations of data collection loops and knowledge consolidation processes to create a model that provides meaningful results as experts may have diverging perceptions of what data represent (see Lebovitz et al., 2021 for a detailed discussion on experts' disagreements during data annotation).

2.2 No-code AI

No-code solutions for software development have been subject to previous research as they enable non-programmers with little or no coding experience to produce various applications (Bhattacharyya and Kumar, 2021; Luo et al., 2021; Lethbridge, 2021; Sahay et al., 2020; Yan, 2021). By adopting low-code principles, enterprises may not only save time and costs, but also narrow the gaps between business operations and information technologies, thereby enabling more rapid development and improvements in product and service quality (Rokis and Kirikova, 2022).

As noted by Sundberg and Holmström (2022, see also Sundberg and Holmström, 2023), a new generation of 'lightweight' no-code AI platforms—also known as AI as a service (Lins et al., 2021) or simply AI service (Geske et al., 2021) platforms—enables non-data scientists to train ML models to make predictions. Such platforms may match, or even outperform coded solutions (Kling et al., 2022). Hence, no-code AI platforms may be widely applied in diverse settings, including citizen science and as low-cost solutions in emerging markets. In the long run, it has been argued that access to user-friendly, low-code AI could democratize adoption of these systems and stimulate their multidisciplinary use (How et al., 2021). For example, new 'drag-and-drop' interfaces enable anyone to develop, train and test AI algorithms in a few hours. In combination with a range of open-source solutions and plugins, this vastly simplifies algorithm development and deployment (Coffin, 2021). The advances are so rapid that within two years of Woo (2020, pp. 961) stating that "AI might be able to automatically produce code", advances in generative AI, tools such as GitHub Copilot and ChatGPT are enabling code generation based on the input of a user. Computer scientists have always dreamt of writing programs that write themselves, and the dream is becoming a commonplace reality. Recently, authors have also recognized the powerful potential utility of no-code apps in educational settings. For instance, Wang and Wang (2021) argue that no-code (or low-code) app development is transforming traditional software development practices, and present a teaching case involving development of a business app.

3. EDUCATIONAL SETTING

As noted by Holmström et al. (2011), rapid technological developments create challenges for maintaining up-to-date curricula for educating professionals who will work in environments with high levels of technology. They highlight several important issues regarding IS teaching, including the importance of ensuring that the students acquire practically relevant skills, through use of appropriate pedagogical approaches, and generic types of knowledge. As AI is being increasingly adopted in diverse domains (Dwivedi et al., 2021), most, if not all, professionals will engage with or be affected by intelligent systems in their careers. However, as mentioned, AI is associated with needs to understand algorithms and hence skills rooted in computer science and engineering. This poses challenges for professionals rooted in other disciplines, not because they have nothing to contribute to AI or gain from its use, but because of a lack of fundamental knowledge of how, for example, a ML system works. A potential remedy, also already mentioned, is to use 'lightweight' AI (Sundberg and Holmström, 2022) in the form of AI service platforms (Geske et al., 2021; Lins et al., 2021), which are easy to use with little to no installation requirements (as they are cloud-based) and have graphical interfaces that help users to train ML models. Here we present an approach for using such a system, the Peltarion (2022) 'no-code' deep learning AI platform (hereafter 'the no-code AI platform', or just 'the platform'), in a higher education setting at the Department of Informatics, Umeå University, Sweden. The department is part of the university's faculty of social sciences and provides three undergraduate educational programs (on behavioral science with an orientation towards IT-environments, digital media production, and system science) and two master programs (on human-computer interaction and IT management), together with individual courses.

The mentioned AI solution enables non-data scientists to upload data, then train and evaluate a ML model that can be deployed via an application programming interface (API). The platform guides users via a graphical interface together with suggestions regarding problem types, workflows, pre-trained models and iterative improvements. The platform was used in an 'AI for business' course (15 credits) at Umeå University, to give the students hands-on experience in training ML models by engaging in a case-based task. The course is open for students with diverse educational backgrounds, as requirements for enrolment are 90 credits in informatics, computer science, business administration, media and communication studies, pedagogics, psychology, political science, sociology (or equivalent competence). In line with the course curriculum (Umeå University, 2022), the learning objectives of the exercise were to "Account for and explain the role of AI in organizational value creation", by giving the students first-hand experience of training ML models. The educational approach is further described in the following section.

4. MATERIALS AND METHODS

To address the RQs posed in Section 1, we followed a group-based project approach presented by Mathiassen and Purao (2002) in the course, inviting the students to engage in development of ways of working and participating in communicative activities regarding 'real-life' problems. As noted by Leidner and Jarvenpaa (1995), such approaches provide opportunities for students to understand the 'messiness' professionals face in industry, acknowledging the social situatedness of these contexts, and that the problems students will face are "unstructured, ambiguous, and immune to purely technical solutions" (Holmström et al., 2011, pp. 2).

We applied the principles of instructions framework advocated by Merrill (2007, 2013) in the educational setting. This incorporates five principles summarized in Table 1: problem-centered learning, activation, demonstration, application, and integration. The framework provides an integrated, multi-strand strategy for teaching students how to solve real-world problems, or complete complex real-world tasks.

Principle	Description
Problem-centered learning	Humans learn better when they are solving problems, so learning is promoted when learners acquire skills in contexts of real-world problems.
Activation	Learning is promoted when learners activate existing knowledge and skills as foundations for a new skill. An important step here is to start at the learner's level. Activation requires learning activities that stimulate the development of mental models and schemes that can help learners to incorporate new knowledge or skill into their existing knowledge framework.
Demonstration	Learning is promoted when learners observe a demonstration of the skill to be learned, e.g., by exposure to examples of good and bad practice.
Application	Learning is promoted when learners apply new skills they have acquired to solve problems. Applying new knowledge or skills to real-world problems is treated as almost essential for effective learning.
Integration	Learning is promoted when learners reflect on, discuss, and defend knowledge or skill they have acquired. The effectiveness of a course is enhanced when learners are provided opportunities to discuss and reflect on what they have learned in order to revise, synthesize, recombine and modify their new knowledge or skills.

Table 1. Principles of the educational approach

The case presented to the students described a fictive organization, 'WeldCorp', specialized in welding, seeking to expand and acquire customers in additional geographical markets while retaining and automating quality measures. To assist the company, we invited the students to develop ways to use ML as a tool to assess welding points. The course module described in this paper consisted of a workshop, a Q&A session, supervising sessions, and a final seminar. Its content is further outlined in Section 5.1. Nineteen students attended the course (14 male and five female), with educational backgrounds including bachelor's degrees in business and administration, computer science, and behavioral science. The empirical materials used in the study presented here, as summarized in Table 2, stem from interactions with the students, the no-code AI platform, and teachers' reflections.

Materials	Source(s)
Students' feedback and course evaluations	E-mails, notes taken during the course, written evaluations and feedback from students.
Students' written assignments and presentations	Two written group reports, and two presentations during a final seminar.
Datasets, models and deployments created by the students	The Peltarion (2022) no-code AI platform
Observations	Teachers' experiences and reflections during and after the course

Table 2. Materials

These materials allowed us to both provide educators with recommendations for using no-code AI and present interesting findings on the benefits and challenges associated with these platforms' use in educational settings. We identified the benefits and challenges by subjecting the empirical data to thematic analysis (Braun and Clarke 2012; Clarke and Braun 2014) through inductively coding the students' activities during the module. More specifically, we coded the activities undertaken by the students in our empirical setting mentioned and observed in the materials, and then aggregated them into themes, informed by the steps in the ML workflow presented in Section 2.1.

5. RESULTS: USING NO-CODE AI IN AN EDUCATIONAL CONTEXT

This section is divided into three parts. In line with Lending and Vician (2012), in Section 5.1 we provide a description of our educational procedures to enable instructors to adopt our approach. Then, the benefits of using no-code AI in education are presented in Section 5.2, followed by challenges we experienced in Section 5.3.

5.1 Detailed Educational Approach

The course module was initiated on December 2, 2021, and the final seminar was held on January 10, 2022. Thus, the duration of the module was a little over a month, including Christmas holiday breaks. The module was initiated with a 3 h workshop session that included an introduction to ML, followed by a demonstration of the no-code AI platform's functionalities, and description of the group assignment. The information presented, and considerations applied, in this workshop are summarized in the following text.

As the students came from different backgrounds, it was clearly stated that the workshop would not include deep examination of phenomena such as neural networks,and focus instead on providing students with sufficient information to get hands-on experience of training ML or deep learning (DL) models. An overview of the current status of ML was presented as increases in the scale of datasets, together with improvements in algorithms and processing speed have increased capabilities for machines to 'learn'. This included presentation of:

- A short video showing how neural networks 'see' things in image data: https://www.youtube.com/watch?v=xS2G0oolHpo&ab_channel=NOVAPBSOfficial

- Figures from an overview by Hilbert and López (2011) of how the capacities of storing data rapidly shifted from analogue to digital formats.
- A comparison of the world's fastest supercomputer in 1997 (ASCI Red), which reached a speed of 1.8 teraflop, and the SONY Playstation 3 video game console that reached the same speed nine years later.

Then, the differences between supervised, unsupervised, and reinforced ML were briefly presented. We emphasized that the module would focus largely on supervised learning, the basis of most commercial and industrial applications of ML today, so the students would need to engage with data labeling. This is important for two reasons. First, collecting and annotating data are crucial but time-consuming activities that take most of the time spent during ML development (Fredriksson et al., 2020). Second, if this element is neglected or poorly done, the resulting ML models will perform poorly and generate inaccurate, irrelevant or even harmful results (Sambasivan et al., 2021).

Next, the lecture outlined the kinds of problems that can be solved by using ML. As noted by Kayhan (2022, p. 123), "many students lack the preparation for the workforce because they cannot conceptualize valid input-output relationships for the problems they propose to solve using ML". Thus, despite the widespread hype surrounding intelligent systems, there is a lack of specificity of the kinds of problems algorithms can actually solve. As noted in Section 2.1, ML is a set of technologies that involve training of algorithms to create models that can provide predictions concerning previously unseen datasets. Hence, ML cannot solve 'general' problems such as 'increasing efficiency' or 'improving quality': they need specific problem formulations accompanied by relevant datasets. Thus, in this part of the lecture we presented a checklist for determining whether ML would be suitable to apply:

1. Do you have a use case?
2. Can the use case be solved by AI / ML (or simpler means)?
3. Do you have data?
4. Do you have annotated data?

We also presented examples of various problems/use cases that ML can solve, such as anomaly detection, classification problems (identifying features in texts and images), building chatbots based on text similarity functions, and various regression problems, such as predictions of sales and housing costs. Before demonstrating the functionality of the no-code platform, we described the ML workflow, both generally as shown in Figure 1 and more specifically for the Peltarion platform, as displayed in Figure 2. Although the platform is now discontinued, this workflow (data collection + preparation, training, evaluation and deployment of an ML model) is at the core of most ML development efforts and protocols applied in other no-code AI platforms (such as BigML, Amazon SageMaker, Google AutoML and Teachable Machine etc.).

Figure 2. The ML Workflow in the No-code AI platform

After presenting the above activities in a traditional lecture, supplemented by visual aids and other materials, we turned attention to the no-code platform.

An important step during the use of no-code AI is to check requirements of the platform of choice in terms of data types (e.g., tabular, images, or text). Familiarity with the selected platform's tools for processing and labeling data is also important. Thus, to provide participating students with an understanding of how the no-code AI platform handled different data types, we used free datasets from Kaggle (2023):

- To explore tabular data, we used the popular "IRIS" dataset, which can be used to predict the species of a flower based on the size of petals and sepals.
- For image data, images of cats and dogs can be used to train a binary classifier. Images of craters on the Moon and / or Mars can be used to train object detectors (if this feature is available in the platform. See Figure 3 for an example).
- To train a model that can make predictions based on NLP (natural language processing), data from the Internet Movie Database (IMDB) can be used to predict whether a text is 'positive' or 'negative'.

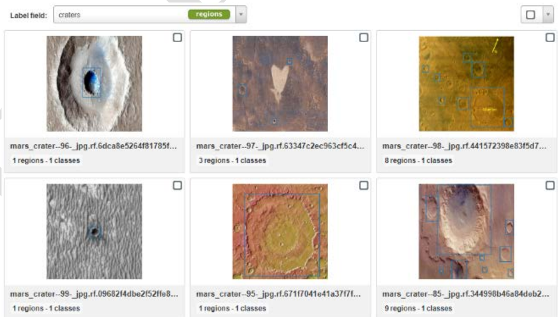


Figure 3. Image Annotation for Object Detection in the BigML Platform

During the demonstration of how to upload data, we briefly described and outlined procedures for various possible formats for tabular and text data (e.g., csv and npy), but not procedures for connecting to 'data warehouses', such as BigQuery or Azure Synapse, as it was irrelevant for the planned task. Instead, we focused more on how to upload image data to the platform, as this was the type of data the students would handle in the following case. An advantage of using no-code AI in such cases is that images can be annotated by placing them in folders that acts as labels, compressing them into zip-files, and then uploading them to the platform. The platform then takes care of processing and cropping the images to standardized formats. A negative effect, which we informed students about, is that important features near edges of the images may be cropped.

Then, we demonstrated various examples of ML problems, and their possible solutions using the no-code AI platform. Depending on the type of data involved, the platform suggests certain problems, as the user chooses the input (data), and one or more targets (labels). As mentioned, examples of such problems include image classifications and image/text similarity searches. Thus, in this phase we also displayed examples of ways to use pre-trained ImageNeT-based and NLP (e.g., BERT)-based models for classifying and predicting patterns in images and texts, respectively. Use of pre-trained models relaxes the requirements to use big datasets, as users can fine-tune these models with their own data. Links to online tutorials and datasets (e.g., Kaggle) were uploaded to the course teaching platform, for students who wanted to proceed by experimenting with different types of data and problems.

In another important part of this demonstration, we showed how ML models can be evaluated. This is done by splitting the dataset(s) into a training set and test (and/or validation) set. The algorithm is not exposed to the test set during training, so it can be used to evaluate how a model performs on previously unseen data. Common pitfalls, such as data bias and overfitting, were also introduced during this session. The platform enabled generation of two indicators that are commonly used for evaluating models: receiver operating characteristic (ROC) curves and confusion matrices, which are especially useful for enhancing students' understanding of the output of ML models, and why their deployment requires careful consideration. Essentially, an image model outputs a probability of what it thinks is present in an image, e.g., '0.76 cat'. Depending on the problem at hand, and associated requirements, a threshold can be set to determine how 'certain' a model must be before it can classify something. Important measures here include accuracy, recall and precision. While accuracy is a measure of a model's overall performance, there is always a trade-off between recall and precision. Students can be taught the relevance of this tradeoff using two types of examples: ML-based spam-filters, and medical diagnostics. When constructing a spam filter it is often more important to minimize numbers of 'false positives' (potentially important emails that end up in the spam filter) than numbers of 'false negatives' (spam emails that end up in the inbox). Thus, precision is a good measure for such a model, as it assesses whether what is being classified as 'spam' really is spam. In contrast, during medical diagnosis avoiding false negatives is often much more crucial than avoiding false positives (as assessed by a recall measure), because wrongly classifying ill people as healthy can have severe consequences for them. For understanding such issues, knowledge of ROC-curves is important, because they illustrate three key aspects of ML models. First, they output probabilities (in contrast to 'exact knowledge'). Second, configuring these outputs involves active choices of thresholds. Third, these choices entail trade-offs between different evaluation measures.

At the end of the demonstration session, the students were divided into two groups and assigned the problem-centered task of helping 'WeldCorp' to use ML as an instrument to assess the quality of their welding joints. A rubric for the task provided a backstory, stating that WeldCorp was launched in 1994 in Gothenburg, and subsequent expansion to other Swedish cities led to the CEO experiencing problems with maintaining quality control. So, s/he is now turning to ML for this purpose. The rubric then told the students:

Your assignment is to help WeldCorp to sustain their growth by leveraging machine learning. Specifically, your task is to analyze welding images (images of good and bad welding points) to develop a model – using the no-code AI platform – that can be useful for WeldCorp in a quality assurance context.

1. *Describe and justify your choices regarding the data processing, problem selection, and model training in the no-code AI platform.*
2. *Describe how you evaluated your model's predictions. Are they accurate enough to use live for WeldCorp? Why/why not?*
3. *Discuss: What could be done by WeldCorp to improve the model's results? How would they implement this type of solution in their business?*

An important aim during this assignment was to prompt students to think about and justify their choices during training, and the output of their model(s), rather than simply striving to optimize the performance of the model(s). As the module is a part of an AI business course, we also wanted the students to discuss how WeldCorp could integrated AI in their organization.

The students were divided into two groups. The start of the course included a presentation exercise, in which the students were asked to state their name and educational background. As two of the students had experience in computer science, we intentionally placed these students in separate groups. To get the students started, they were given a small dataset of 157 images of good and bad welds. The groups were then given enterprise accounts providing access to the no-code AI platform. Before engaging in a similar project, we advise instructors to carefully assess the kinds of user configurations that candidate platforms offer, as their user management options vary, and potential issues must be addressed before the students attempt to use them.

Five days after the initial workshop, a Q&A session was held with the student groups. No instructions were given before this session and the content was largely based on the students' queries. Most questions concerned data. This was consistent with expectations, as models trained using the intentionally limited dataset handed out during the previous session would perform badly, regardless of the platform settings that the students chose. As already mentioned, data collection and processing play a key role in ML and "there is no AI without data" (Gröger, 2021). Illustrative queries from the students concerned the quality of the supplied dataset, tentative workarounds, and image formats. However, the main conclusion the students drew was that more data was needed to train a model that would produce relevant results.

Between the Q&A and final seminar, the students were supposed to email or book appointments with the responsible teachers if they needed supervision. The teachers could observe and aid the students as they uploaded data then trained and evaluated ML models. After the Q&A session we observed how the students engaged in data collection and uploaded larger datasets with various images to the platform. As the students aimed to train models based on a binary classification of good and bad welds, they needed two labels ('good' and 'bad'). The students applied the procedures previously demonstrated to them, trained several models, and iteratively fine-tuned the platform settings, using several sources of data, including social media, Google image search, and Kaggle.

While the workshop and Q&A session were held on campus, the final seminar was held via Zoom (January 10) as this was during a time when staff and students at higher education institutions were gradually returning to campus after the COVID-19 pandemic. The written assignment included the following instructions:

You will be presenting your results both in the form of a short paper, max ten pages, and orally in the final seminar. During the seminar each group will get 30 minutes to present their results. You must also participate actively by answering questions and comments regarding the presentation. Your short paper should begin with a cover page on which you state the names of the group participants, the name of the course and the semester. It is to be handed in at the start of the seminar.

During discussions in a final seminar the students were encouraged to reflect upon the ML process, to enable them to integrate their acquired skills. In addition to discussing the ML workflow, the students also proposed ideas for operationalizing their work in a live setting, such as using automated cameras to feed data on welding points for evaluation by the DL model. In this seminar the teachers mainly played a facilitating role, as the students posed questions and reflected on their results. The students received pass or fail grades for the task. To pass they needed to:

- Present a logically coherent suggestion for WeldCorp, both in writing and orally during the seminar.
- Formulate results and associated discussion in a grammatically correct way and with consistent use of concepts and terms.

The teaching activities outlined above are linked to the five instruction principles and summarized in Table 3. Depending on the course, and available data and case(s), these activities can be varied. For example, the workshop can be divided into two separate events, with an initial lecture focusing on theoretical aspects of ML, followed by a more hands-on workshop. Moreover, the group case can be presented as an individual or pair-wise task, although this might neglect the collective character of data work.

Principle	Description
Problem-centered learning	The students were presented with a case of a welding company, WeldCorp, seeking to expand and scale up its business while improving quality control. To help these efforts they were encouraged to apply ML to differentiate between good and bad weld points.
Activation	Since the students had diverse educational backgrounds (business and administration, computer science, and behavioral science), we chose to use a no-code AI platform. This enabled them to incorporate previous skills and work during the course, even if they lacked previous experience of data science.
Demonstration	We showed the students several examples of ways to train ML models via the no-code AI platform. Students were encouraged to take tutorials and experiment with different types of open datasets (e.g., table-, text-, and image-based), and problems that can be accessed through the platform.
Application	The students were divided into two groups and each student was given access to an enterprise account enabling them to use the no-code AI platform to address a new type of problem by applying the previously demonstrated procedures.
Integration	Students were encouraged to reflect on their learning during the final seminar in both a survey and the course evaluation. During the final seminar they were also expected to learn from each other by preparing questions for the other group.

Table 3. Activities that we and the students engaged in, linked to the five principles of instruction

5.2 The Benefits of Using No-code AI in Education

This subsection presents observed benefits of using no-code AI to teach ML, which are described below and summarized in Table 4.

Benefit 1: *Visualization of data and provision of a graphical interface for uploading data.*

As already mentioned, a crucial and time-consuming part of working with ML is collecting and processing data. As the no-code AI platform automated many parts of the ML workflow, students had time to spend during the exercise on consideration and labeling of the data. This was an anticipated and important part of the task, especially as previous studies have highlighted tensions among people involved in labeling data for supervised learning (Lebovitz et al., 2021).

In their course evaluations and written feedback the students heavily emphasized an increase in their awareness of the importance of data, and how the no-code approach enabled them to focus on important features of the datasets used, potential flaws in them, and problem-solving rather than model-optimization, as illustrated by the following three quotations:

"I've obtained practical knowledge and experience of the impact of data. And I've seen the impact of flaws in the dataset first-hand. Thus, I think this was an optimal learning method considering our (and my) educational background." – student evaluation.

"[I've learnt] that data matters! The choice, generating and cleansing of data are crucial." – student evaluation.

"For me, the barrier to understanding the practical use of AI (or to ever try it myself) has been my lack of programming and coding skills. With the no-code approach, I got the opportunity to try experiments and thus got a 'black-boxed' grasp of how it works. With that, I could focus on the problem that I wanted to solve, the learning dataset and its effect on the results, and also on the result itself. So, I think I learned more about AI in this course than I have in all the other courses combined, and that is without any code." – student evaluation.

Both groups chose to label their images in a binary fashion as 'good' or 'bad'. To establish the consensus required for creating 'ground truths', one of the groups formalized the data labeling process in their report with a ‘weld quality framework’. The other group strongly engaged in data augmentation as they extended their dataset 4-to 5-fold by manipulating the images through zooming, cutting and rotating them. These slightly different approaches were displayed in the results and reflected upon in the student reports. While the group that applied data augmentation focused more on the performance of the models they created, and thus achieved better measures (lower rates of false positives or negatives), the other group focused more on trying to explain the output of the models they created, i.e., why the models made certain predictions.

Benefit 2: *Access to a portfolio of pre-trained models, tutorials and datasets, as well as automatic selection and fine-tuning of algorithm(s) for training.*

Both groups ended up using a pre-trained model (EfficientNetB0) to solve an image classification problem (single label) in the platform. Each group formed training, validation and test sets respectively containing 80, 10 and 10% of their full datasets (images), which is common practice and a default option in the platform. The students refined their models’ outputs in two ways. First, they iteratively adjusted settings in the platform, such as increasing the training rate (with careful monitoring of the variances of performance measures of the predictions generated by splitting the dataset to avoid overtraining the model). The platform assists such adjustment by suggesting settings to enhance the models’ performance, e.g., switching to a different pretrained model, and modifying the learning rate (Figure 4).

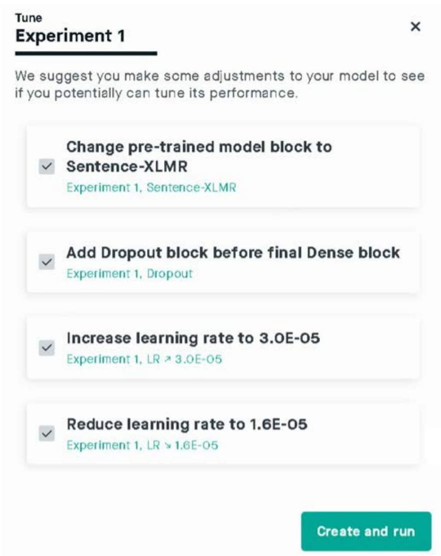


Figure 4. Suggestions to Improve Model Performance

Benefit 3: *Visual interface for evaluating and comparing the performance of models (e.g., through ROC curve- and confusion matrix-based analyses).*

Second, as particularly strongly emphasized by one of the groups, the students strove to ensure that included data were contextually relevant, and suitable for WeldCorp’s purposes. This was done after they received output from the ML model in the form of confusion matrices and ROC-curves (Figures 5 and 6) and could assess whether certain types of images were incorrectly classified, identify potential biases in the data, and signs of model overtraining. Examples mentioned during the final seminar were images of painted welds, which would not be relevant in the industrial context they imagined.

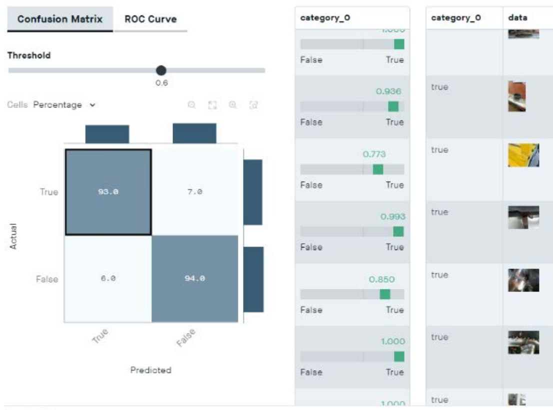


Figure 5. Illustrative Model Evaluation Output

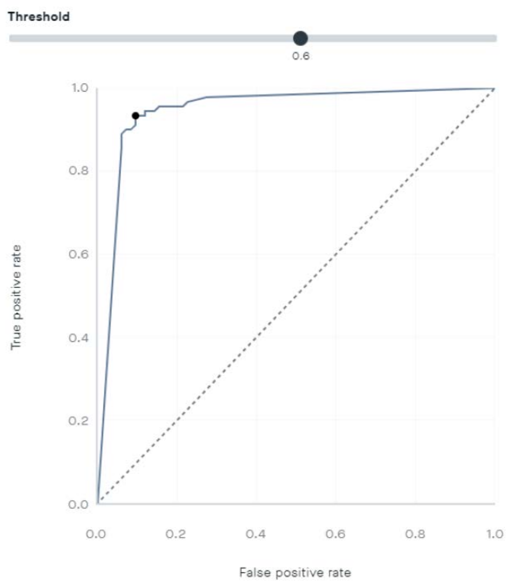


Figure 6. ROC-curve from one of the student reports

Available features briefly mentioned in the course included tools to deploy the models created in the platform. This was not relevant to the assigned task, as the students were not expected to integrate their solution in a live environment, we presented a few paths to do so. Examples included plugins for common software (such as Excel, Google Sheets, Bubble) as well as the ability to call APIs for easy integration of a model in an operating environment. The platform also includes a graphical interface for making predictions regarding new images, as shown in Figure 7. We used this function during the final seminar, to show the students how their models performed on selected images of good and bad welds.

Figure 7. Results of a test of a model's performance on unseen data during the final seminar

Thus, by simplifying parts of the ML workflow related to training, evaluating and deploying models, learners can focus on data collection, and interpreting outputs of the models, to gain a sense of whether the chosen approach is suitable and feasible rather than engaging in model optimization. Based on our materials, we generated themes in the form of distinct ways that no-code AI facilitates learning about ML. These themes are described in Table 4.

ML workflow	Role of no-code AI
Data collection / preparation	Provision of a graphical interface for visualization, uploading and processing data.
Model training	Access to a portfolio of pre-trained models, tutorials and datasets, as well as automatic selection and fine-tuning of one or more algorithm(s) for training.
Evaluation	Visual interface for evaluating and comparing the performance of models (e.g., through ROC curve- and confusion matrix-based analyses).
Deployment	API-interfaces with complementary plugins to aid integration in organizational settings.

Table 4. Ways that no-code AI can facilitate learning about ML

5.3 Challenges with using no-code AI in education

Our approach was not free of challenges, including three that are summarized here. First, it is important to formulate a live case in terms of ML and make a preliminary judgement of the feasibility of the students collecting the necessary data during the task. Finding an appropriate case may be time consuming, but data repositories, such as Kaggle, may aid this process. Second, as mentioned, the teachers also encountered challenges related to user management routines before the module started and needed help from the platform owners to set up separate organizations for the students. These challenges highlight the importance of considering and addressing potential user management issues in advance and choosing an appropriate platform for the intended purposes. The market for these platforms is rapidly evolving. While the Peltarion platform is now discontinued, several alternatives are available, such as BigML, HuggingFace and solutions from large tech companies (e.g., Microsoft Azure, Amazon SageMaker, Google AutoML, and Teachable Machine). These often come in both free and paid versions. For individual use, the free versions may be suitable for smaller tasks and datasets. A common advantage of paid versions is incorporation of more collaborative features, which enables re-use and comparisons of student projects over the years. Whichever platform and version is chosen it is also important to ensure that students do not upload sensitive data, depending on the regulatory context of the educational setting. Third, the student feedback included proposals that groups should be smaller in future versions of the course, as they experienced difficulties in engaging everyone simultaneously when using the platform.

6. CONCLUDING DISCUSSION

As the no-code approach enabled students to engage in collective data work the selected empirical setting provided an ideal opportunity to address our two questions:

RQ1: *How can no-code AI be used to teach ML in non-technical educational programs?*

RQ2: *What are the benefits and challenges of using no-code AI in education?*

We answer RQ1 by proposing a problem-centered approach to using no-code AI in higher education, with instruction to teachers. Regarding RQ2, we show how no-code AI can help to guide students through the ML workflow (data processing, model training, evaluation and deployment), and present important challenges (ML case construction, platform selection and user management, and student group composition) that we encountered during the course.

Our contribution to the IS education literature is two-fold. First, we provide information for instructors on how to incorporate no-code AI in their courses. Second, we provide insights into the benefits and challenges of using no-code AI tools to support the ML workflow in educational settings.

Through this study we have set the stage for incorporating a new generation of AI tools in IS curricula by showing how they can be used to support students in analyzing live cases, particularly in conjunction with an approach based on principles of instruction. By doing so, in this paper we have proposed an innovative solution to an IS teaching need, grounded in theory, and tested in an educational setting (Lending and Vician, 2012). The novelty of our approach is the application of tools that are usually only accessible to computer scientists to problems related to business practices and phenomena addressed in social sciences. As the no-code AI tools available are rapidly increasing and evolving (a few, of many, examples of contemporary no-code or low-code solutions that support the ML workflow include BigML, Huggingface and Teachable Machine) we urge educators to keep track of this development, and find approaches to implement such tools in their curricula, in combination with lessons on how to use AI in effective and responsible ways.

7. REFERENCES

Bhattacharyya, S. S., & Kumar, S. (2021). Study of deployment of "low code no code" applications toward improving digitization of supply chain management. *Journal of Science and Technology Policy Management,* 14(2).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY: Springer.

Braun, V., & Clarke, V. 2012. Thematic analysis, In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology, Vol. 2. Research designs: Quantitative, qualitative, neuropsychological, and biological* (pp. 57–71). American Psychological Association.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999, March). The CRISP-DM user guide. In *4th CRISP-DM SIG Workshop in Brussels in March* (Vol. 1999).

Chen, L. (2022). Current and Future Artificial Intelligence (AI) Curriculum in Business School: A Text Mining Analysis. *Journal of Information Systems Education*, 33(4), 416-426.

Coffin, E. (2021). I Think I Need AI! What is AI? *BNP Media.*

Clarke, V., & Braun, V. (2014). Thematic analysis. In *Encyclopedia of Critical Psychology* (pp. 1947-1952). Springer, New York, NY.

Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., ... & Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy, *International Journal of Information Management*, 57.

Fredriksson, T., Mattos, D. I., Bosch, J., & Olsson, H. H. (2020). Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In *International Conference on Product-Focused Software Process Improvement* (pp. 202-216). Springer, Cham.

Geske, F., Hofmann, P., Lämmermann, L., Schlatt, V., & Urbach, N. (2021). Gateways to Artificial Intelligence: Developing a taxonomy for AI Service platforms, *European Conference on Information Systems (ECIS)*.

Gröger, C. (2021). There is no AI without data. *Communications of the ACM*, 64(11), 98-108.

Hilbert, M., & López, P. (2011). The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025), 60-65.

Holmström, J., Sandberg, J., & Mathiassen, L. (2011). Educating reflective practitioners: The design of an IT Management Masters Program, *Americas Conference on Information Systems (AMCIS)*.

How, M. L., Chan, Y. J., Cheah, S. M., Khor, A. C., & Say, E. M. P. (2021). Artificial Intelligence for Social Good in Responsible Global Citizenship Education: An Inclusive Democratized Low-Code Approach. In *Proceedings of the 3rd World Conference on Teaching and Education,* 81-89.

Humble, N., & Mozelius, P. (2022). The threat, hype, and promise of artificial intelligence in education. *Discover Artificial Intelligence*, 2(1), 1-13.

Jordan, M. I., & Mitchell, T. M. 2015. Machine learning: Trends, perspectives, and prospects, *Science,* 349(6245), 255-260.

Kaggle (2023). www.kaggle.com, last accessed, March 2023.

Kayhan, V. (2022). When to Use Machine Learning: A Course Assignment, *Communications of the Association for Information Systems*, vol 50, 122-142.

Kelleher, J.D., Brendan, T. (2018). Machine Learning 101. In *Data Science*, MIT Press, 97-150.

Kling, N., Runte, C., Kabiraj, S., & Schumann, C. A. (2022). Harnessing Sustainable Development in Image Recognition Through No-Code AI Applications: A Comparative Analysis. In International Conference on Recent Trends in Image Processing and Pattern Recognition, 146-155. Springer, Cham.

Kühl, N., Schemmer, M., Goutier, M., & Satzger, G. (2022). Artificial intelligence and machine learning. *Electronic Markets*, 32, 2235-2244.

Leavitt, K., Schabram, K., Hariharan, P., & Barnes, C. M. (2021). Ghost in the machine: On organizational theory in the age of machine learning. *Academy of Management Review*, 46(4), 750-777.

Lebovitz, S., Levina, N., & Lifshitz-Assaf, H. (2021). Is AI Ground Truth Really 'True'? The Dangers of Training and Evaluating AI Tools Based on Experts' Know-What. *MIS Quarterly*, 1501-1525.

LeCun, Y., Bengio, Y., & Hinton, G. (2015) Deep learning, *Nature*, 521(7553), 436-444.

Lending, D., & Vician, C. (2012). Writing IS Teaching Tips: Guidelines for JISE Submission. *Journal of Information Systems Education*, 23(1), 11-18.

Lethbridge, T. C. (2021, October). Low-code is often high-code, so we must design low-code platforms to enable proper software engineering. In *International Symposium on Leveraging Applications of Formal Methods*, 202-212. Springer, Cham.

Liebowitz, J. (1992). Invited Paper: Teaching an Applied Expert Systems Course: A Content Outline. *Journal of Information Systems Education*, 4(3), 5-10.

Liebowitz, J. (1995). Integrating Expert Systems Throughout the Undergraduate Curriculum. *Journal of Information Systems Education*, 7(1), 34-36.

Lins, S., Pandl, K. D., Teigeler, H., Thiebes, S., Bayer, C., & Sunyaev, A. (2021). Artificial Intelligence as a service, *Business & Information Systems Engineering*, 63(4), 441-456.

Luan, H., & Tsai, C. C. (2021). A review of using machine learning approaches for precision education, *Educational Technology & Society*, 24(1), 250-266.

Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021, October). Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1-11.

Ma, Y & Siau, K. (2019). Higher Education in the AI Age. *Americas Conference on Information Systems (AMCIS)* Proceedings. 4.

Mathiassen, L., & Purao, S. (2002). Educating reflective systems developers, *Information Systems Journal*, 12(2), 81-102.

Merrill, M.D. (2007). A task-centered instructional strategy, *Journal of Research on Technology in Education*, 40(1), 5-22.

Merrill, M.D. (2013). *First principles of instruction: Identifying and designing effective, efficient and engaging instruction,* Hoboken, NJ: Pfeiffer/John Wiley & Sons.

Peltarion (2022). *The Peltarion deep learning platform. www.peltarion.com,* accessed September 2022 (discontinued).

Richardson, M. L., & Ojeda, P. I. (2022). A "Bumper-Car" Curriculum for Teaching Deep Learning to Radiology Residents. *Academic Radiology*, 29(5), 763-770.

Rokis, K., & Kirikova, M. (2022). Challenges of Low-Code/No-Code Software Development: A Literature Review. In *International Conference on Business Informatics Research*, 3-17. Springer, Cham.

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020, August). Supporting the understanding and comparison of low-code development platforms. In 2020 *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 171-178. IEEE.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021, May). "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1-15.

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181, 526-534.

Simsek, Z., Vaara, E., Paruchuri, S., Nadkarni, S., & Shaw, J. D. (2019). New ways of seeing big data, *Academy of Management Journal*, 62(4), 971-978.

Sturm, T., Gerlach, J. P., Pumplun, L., Mesbah, N., Peters, F., Tauchert, C., ... & Buxmann, P. (2021). Coordinating human and machine learning for effective organizational learning. *MIS Quarterly*, 45(3).

Sundberg, L. & Holmström, J. (2022). Towards 'Lightweight' Artificial Intelligence: A Typology of AI Service Platforms. *Americas Conference on Information Systems (AMCIS).*

Sundberg, L., & Holmström, J. (2023). Democratizing artificial intelligence: How no-code AI can leverage machine learning operations. *Business Horizons.*

Umeå University (2022). *AI for business course curriculum, https://www.umu.se/en/education/syllabus/2in408/* accessed March 2023.

Yan, Z. (2021). The Impacts of Low/No-Code Development on Digital Transformation and Software Development. *arXiv preprint arXiv*:2112.14073.

Wang, S., & Wang, H. (2021). A Teaching Module of No-Code Business App Development. *Journal of Information Systems Education*, 32(1), 1-8.

Woo, M. (2020). The rise of no/low code software development—No experience needed? *Engineering*, 6(9), 960-961.

AUTHOR BIOGRAPHIES

Leif Sundberg is an associate professor at the Department of Informatics, Umeå University. Sundberg's research interests involve digital government, the use of no-code artificial intelligence, philosophy of technology and risk society studies. Sundberg has a broad teaching experience from engineering management and information systems. He has published his work in journals such as Safety Science and Information Polity and presented it at international conferences like IFIP EGOV-CeDEM-EPART and AMCIS.

Jonny Holmström is a professor of Information Systems at Umeå University and director and co-founder of Swedish Center for Digital Innovation. His research interests are digital innovation, digital transformation and digital entrepreneurship. He is serving at the editorial boards of CAIS, EJIS, Information and Organization, and JAIS. His work has appeared in journals such as Communications of the AIS, Design Issues, European Journal of Information Systems, Information and Organization, Information Systems Journal, Information Technology and People, Journal of the AIS, Journal of Information Technology, Journal of Strategic Information Systems, MIS Quarterly, Research Policy, and The Information Society.

International Journal of Multidisciplinary Innovative Research
ISSN: 2583-0228 Volume 2, Number 1 (Jan' 2022) pp. 1-14
© CIIR, Noida, INDIA
https://www.ciir.in

Study of Machine Learning No-Code Platform

Kartikey Ahlawat

University Institute of Engineering, Chandigarh University, Mohali, Punjab, India.

Email: kartikeyprit@gmail.com

ABSTRACT: Study on a web application based on no-code platform ML Modeller, which targets to provide predictions using various machine learning algorithms under classification and regression. Users have to upload data on which predictions are needed, the no-code platform will automatically train on the data and even show you the accuracy is achieved, now users can get required predictions. No-Code platforms will reduce Manual Data Analyzing efforts and overcome the problem of false insights and guessing. Website's performance is also evaluated if people are using it or not.

KEYWORDS: Machine learning, Regression, Classification, No-code platform, Automation, Website, Prediction, and Analysis.

INTRODUCTION

Machine learning no-code platform has a huge use case in today's scenario. Machine learning-based no-code platform, to help general people train machine learning models on their datasets, just by uploading them and making predictions on the input dataset. Less than 1% of working people are software developers and the main objective of no-code platforms is to disperse the power of software to the rest 99%. Moreover, this industry is expected to cross $21 billion by the end of 2022 [1]. It will be useful for people doing business or caring for some research work or any task requiring data analysis and making predictions. As most of these people do not know how to analyze or make predictions so that they can take decisions accordingly. Currently, they might be going through data manually, which is time-consuming and might even give false insights in most cases, which could eventually result in making bad decisions. Almost anyone with the relevant data can use it and get the required predictions with just a few clicks. Gartner predicted that by 2024 at least 65% of all new applications will be using or created with low-code or no-code platforms. No-code platforms will decrease the time and cost to the company during performing tasks, eventually benefiting the company and shareholders.

Moreover, it will help in developing and deploying applications fast, resulting in a decrease in development to production time [2]. During the pandemic, in online mode of working it was observed that its workload increased substantially as compared to offline mode, majority organizations stated that IT was not able to complete all tasks in that year and the picture could have been different if some automated tool was there to ease the load [3]. ML Modeller is based on a no-code platform, which means even a person with no coding skills can avail of benefits. Users do not require any prior machine learning knowledge. ML Modeller works on images and comma-separated files (.csv) as well, containing numeric and character values. These datasets are more common, easily available, and in more use than audio datasets. (According to the current scenario).

It will help in decision-making for businessmen, recruiters, analysts, students, teachers, and the list continues. Less time-consuming and more accurate predictions than manual analysis. It will be useful for people doing business or caring for some research work or any task which requires data analysis and making predictions. Users have to upload two excel sheets, one would be on which model will get trained (Training set) and the other which contains input values. And then click upload to get the corresponding predictions to input values and accuracy and metrics for evaluation (MSE, RMSE, and MAE). And the best part is no need to log in to do so, moreover, websites would not store any user data. Moreover, a Blog platform will also be integrated where users can sign-up and start creating content for the community [4].

OBJECTIVE

The main objective is to make the power of machine learning easily available for normal people having no prior knowledge of coding or machine learning. Under no-code platform technology, users have to just upload a dataset and choose the appropriate machine learning model/algorithm and then can have insights as well as predictions. Moreover, users can check the accuracy of the model as a classifier or metrics for evaluation like root mean square error (RMSE), mean square error (MSE), and mean absolute error (MAE). Less time-consuming and more accurate predictions than manual analysis [5].

REVIEW OF LITERATURE

At the time of the creation of ML Modeller, there was no such application in the market. And the existing state-of-the-art is writing machine learning code manually and then getting predictions or making observations according to one's insightfulness, this requires the knowledge of machine learning and coding. But now there has been more development made in the field of ml-based no-code platforms such as Google had launched a similar app- 'Teachable Machine', which only works on image and audio datasets. AWS launched 'SageMaker' accepts all kinds of data like text, numeric, image, etc, moreover automating the ml pipeline i.e. create, train, and deploy. Similarly, in 2019 Microsoft released the AutoML feature in ml.net, for the purpose of automating the ml pipeline.

ML Modeller works on images and moreover on comma-separated files (.csv) as well, containing numeric and character values. These datasets are more common, easily available, and in more use than audio datasets (according to the current scenario) [1]. ML Modeller is based on a no-code platform, which means even a person with no coding skills can avail of benefits. Users do not require any prior machine learning knowledge. It will help in decision-making for businessmen, recruiters, analysts, students, teachers, and the list continues. Less time-consuming and more accurate predictions than manual analysis. The only benefit of writing ML code manually is that one can build data specific ml model which might perform better than the website (Higher accuracy) but only if the person knows ML enough to do so [5].

PROBLEM STATEMENT AND SOLUTION

In this section, we discuss the data prediction problems that people around the globe face irrespective of their knowledge in technical terms.

Problem Statement

Machine learning is considered a branch of artificial intelligence, the main idea is that computers not only use pre-written algorithms but also learn how to solve problems. Starting from the analysis of traffic jams and ending with self-driving cars, more and more tasks are transferred to self-learning machines. But the problem arises when people with no prior machine learning knowledge or coding skills require machine learning models as decision-making tools in their daily life. Apart from this, Data Analyzing and prediction making models are costly and due to which even most of the technology companies are unable to afford this technique but here an introduction to self-learning machine models and their better accuracy plays a crucial role to avail them the same benefit as a Machine Learning knowledgeable person or tech giant companies.

Solution

As stated in the above problem statement self-learning machine models play a crucial role to overcome this situation and also it makes the machine models more accurate, reliable, effective, less effort, and time-consuming model at the cheapest affordability. ML Modeller is designed to take care of research-related work and business alliances [5].

Table 1: Drawbacks of existing state of the art and how ML Modeller overcome them.

Sl.	The existing state-of-the-art	Drawbacks in existing state of art	Overcome
1.	Coding the ml model from scratch.	→ Need to have programming knowledge. → Need to know how to code like python, R, etc. → Need to have knowledge about different ml models and when to apply which. → Time-consuming	→ No need to have programming knowledge as the process is automated, you only have to check predictions and metrics for evaluation (MSE, RMSE and MAE) and accuracy. → No need to know how to code as it is done at the backend for you. → No need to have prior knowledge about ML models →Saves time, as users do not have to write code.
2.	Going through data manually	→ Time-consuming → False insights → This results in making bad decisions.	* (Above points apply)

ML Modeller is invented to solve:

(1) Saving time: Saves time, as users do not have to write code.
(2) Data: The website is not storing any user data.
(3) No need to have programming knowledge as the process is automated, you only have to check predictions and metrics for evaluation (MSE, RMSE, and MAE) and accuracy.
(4) No need to know how to code as it is done at the backend for you.
(5) No need to have prior knowledge about ML models.

(6) No need to go through data manually, as it might give false insight and eventually result in making bad decisions [5].

PERFORMANCE EVALUATION

The application has been deployed on the web in August 2021. And ever since application is given consistent growth. Moreover, let's analyze the service-share per country with per month growth rate from the stats below.

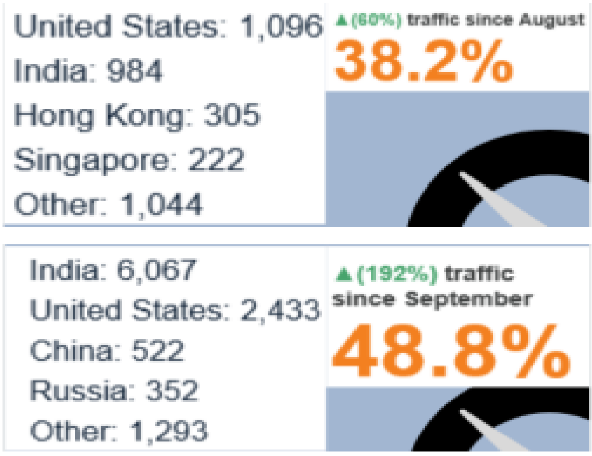


Figure 1: Web traffic country wise and increase from the previous month

[Source: newsletter@cloudflare.com] [3]

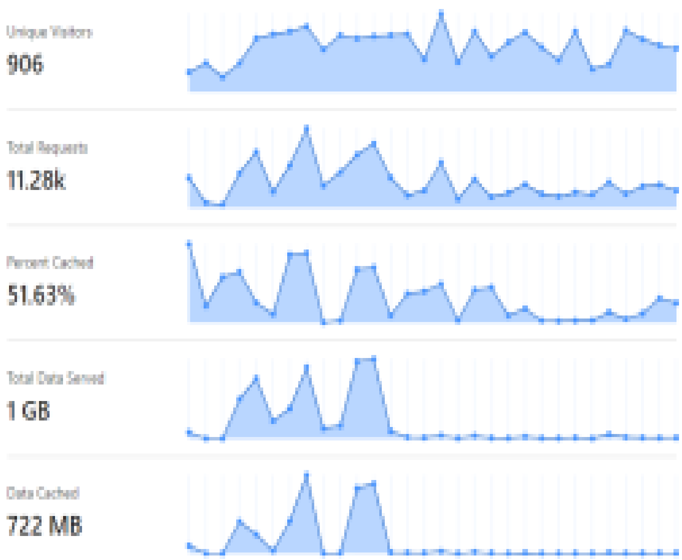


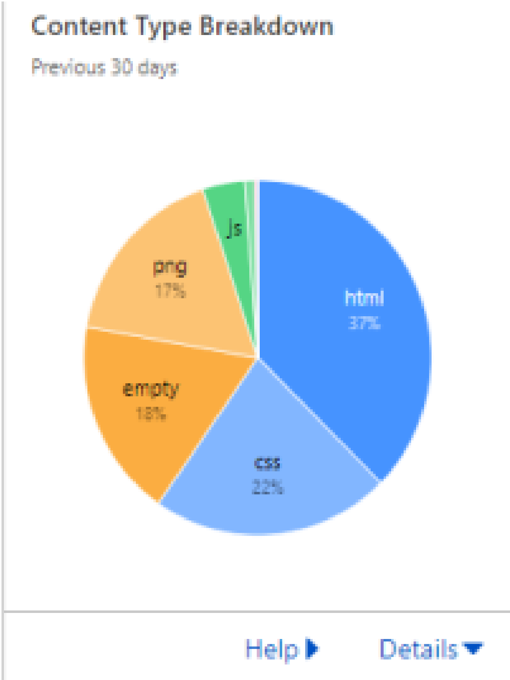
Figure 2: Basic website statistics [3]

So by observing the above data (Figure 1.) there is significant growth in traffic. Moreover by observing figure 2 (data is from 9th October to 8th November); unique visitors/users are also considerably balanced with existing visitors/users. It is as important as existing visitors/users to revisit as it is to gain or attract new visitors/users. Total request is 11.28K, which is considered decent for a new service-based web application. Percent cached is 51.63%, which is poor. The reason being is that lately more users are from India and the server on which the app is running is in the USA, this causes CDN misses (CDN→ Content Delivery Network), as the local servers do not always cache the user data. But if you are data is cached then the next time you visit the app URL it will take less time to load and increase the throughput and decrease the overhead processing [3].

Version	Requests
HTTP/1.1	4.46k
HTTP/3	4.32k
HTTP/2	2.49k
HTTP/1.0	12

Figure 3: HTTP versions used by visitors [3]

On observing figure 3, we observed that most of the users access the website via HTTP/1.1 closely following HTTP/3. HTTP/3 is the latest and faster as it follows UDP and its previous versions follow TCP. HTTP/1.0 is the oldest among these and least used to access ‘ML Modeller’ [3].



Type	Requests
html	4.22k
css	2.51k
empty	2.04k
png	1.93k
js	444
svg	106
woff	24
txt	2

Figure 4: Content-Type Breakdown [3]

Data in figure 4 represents the breakdown by content type of all traffic flowing through Cloudflare to the website (including both cached and uncached responses). The classifications include: “jpeg”, “HTML”, “png”, “gif”, “CSS”, “javascript”, “JSON”, “octet-stream”, “plain”, “ocsp-response”, “x-shockwave-flash”, “XML”, “mixed”, “jpg”, “SVG”, “webp” Note: For “empty” there was either no content-type header or the content header was empty [3].

WORKING OF APPLICATION

Users have to upload two excel sheets, one would be on which model will get trained (Training set/dataset file) and the other which contains input values (Input file). And then click upload to get the corresponding predictions to input values and accuracy and metrics for evaluation (MSE, RMSE, and MAE). And the best part is no need to log in to do so, moreover, websites would not store any user data [5].

Note: ‘Input file’ will only contain independent feature values. ‘Dataset file’ will contain training data, containing both dependents as well as independent features.

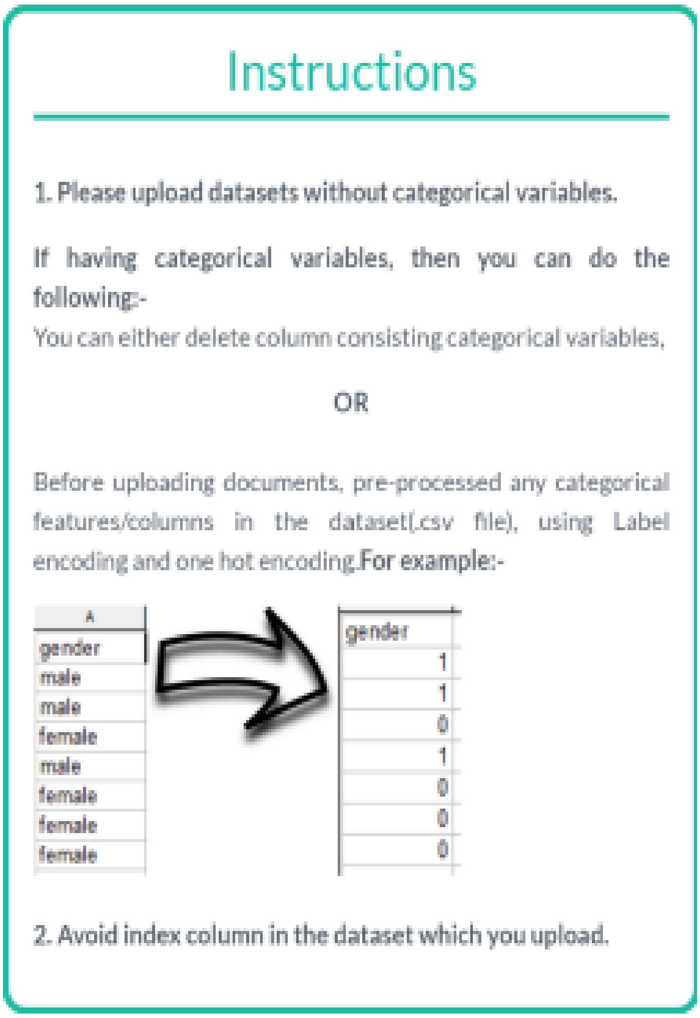
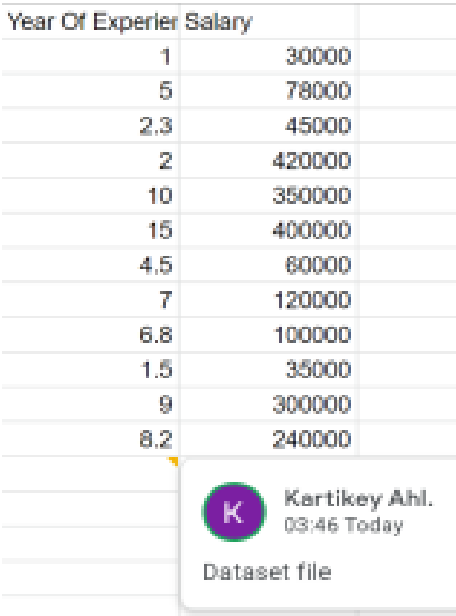


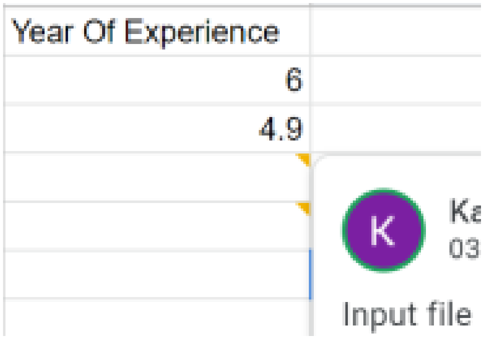
Figure 5: Instruction to be followed while uploading dataset on the website [5]

Below are the snapshots (Figure 6 and Figure 7), which might help in understanding file structure:



Year Of Experier	Salary
1	30000
5	78000
2.3	45000
2	420000
10	350000
15	400000
4.5	60000
7	120000
6.8	100000
1.5	35000
9	300000
8.2	240000

Figure 6: Dataset file on which model is trained [5]



Year Of Experience
6
4.9

Figure 7: Input file on which user need predictions [5]

METHODOLOGY

Collection of Datasets

Have to gather data according to requirement/specifications. In our case Data training and input data are given by the user [6].

Installation of Libraries

Before getting our hands dirty in code, we have to install some required libraries. These libraries have a diverse set of ml functions and also increase the execution speed of programs [6].

Data Pre-processing

In this step, data is pre-processed and converted into a well-structured table of information. There are five different steps involved under it, as stated under:

Importing the Libraries

In this, we import desired libraries needed in developing models. Like some of the usual libraries are NumPy → for solving mathematical expressions, matplotlib.pyplot → for visualization of the result, and Pandas → for importing dataset [6].

Importing the Dataset

At this step with the help of the pandas library, we import the dataset, on which our model will be built. After this, we have to split the data set into independent (X) and dependent (y) variables [6].

Taking Care of Missing Data

Sometimes data contains missing values in columns. These missing values can reduce a model's accuracy. We replace these missing values (Nan) with the mean of the whole column [6].

Splitting the Dataset into the Training Set and Test Set

Via using the train_test_split method we split the dataset into training (on which model is trained) and test set (on which model is tested). Splitting results in 4 variables- X_train, X_test, y_train and y_test [6].

Feature Scaling

Sometimes values in our dataset differ/deviate very much. Like one column containing values from 100 to 1000 and another column containing values from 0 to 30. So in this case, by default first column will have more effect on the dependent variable (output) than the other. So to scale all features in one scale so there is no partiality on basis of value, on the dependent variable [6].

Choosing a Model

'ML Modeller' provides the service of classification and regression models.

Classification model users can choose from:

- Decision Tree
- Kernel SVM
- K-nearest neighbour
- Linear SVM
- Naive Bayes
- Random Forest

Regression model users can choose from:

- Simple Linear Reg.
- Multiple Linear Reg.
- Polynomial Reg.
- Support Vector Reg.
- Decision Tree Reg.
- Random Forest Reg. [6]

Training Model

Now website will train the chosen model on the user dataset. The model will then be able to find trends present in the dataset [6].

Evaluating

In this step, via scikit-learn will import the accuracy_score, mean_squared_error, and mean_absolute_error, from which in the case of the classifier model website will be able to get the model's accuracy by calculating the count of false negative and false positive. And in the case of the regressor model website will get Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as metrics for evaluation [6].

Hyperparameter Tuning

It is the problem of choosing a set of optimal hyperparameters for a learning algorithm (the algorithm used for the training model, e.g. SVM, KNN, etc). A hyperparameter is a parameter whose value is used to control the learning process. After performing it, we get optimal parameters that will give us the best accuracy. We put these parameters in the training step, i.e step 7.5. [2].

Note: In ML Modeller we have kept ideal hyperparameters and we are not implementing it explicitly [6].

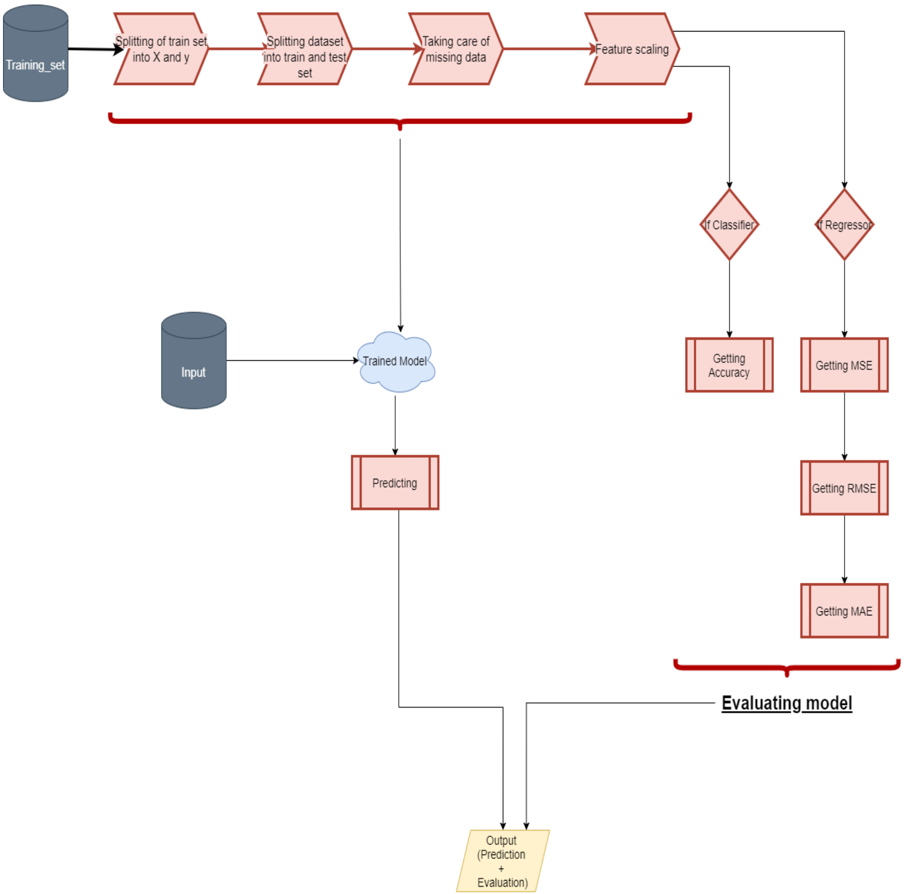


Figure 8: Data Flow Diagram

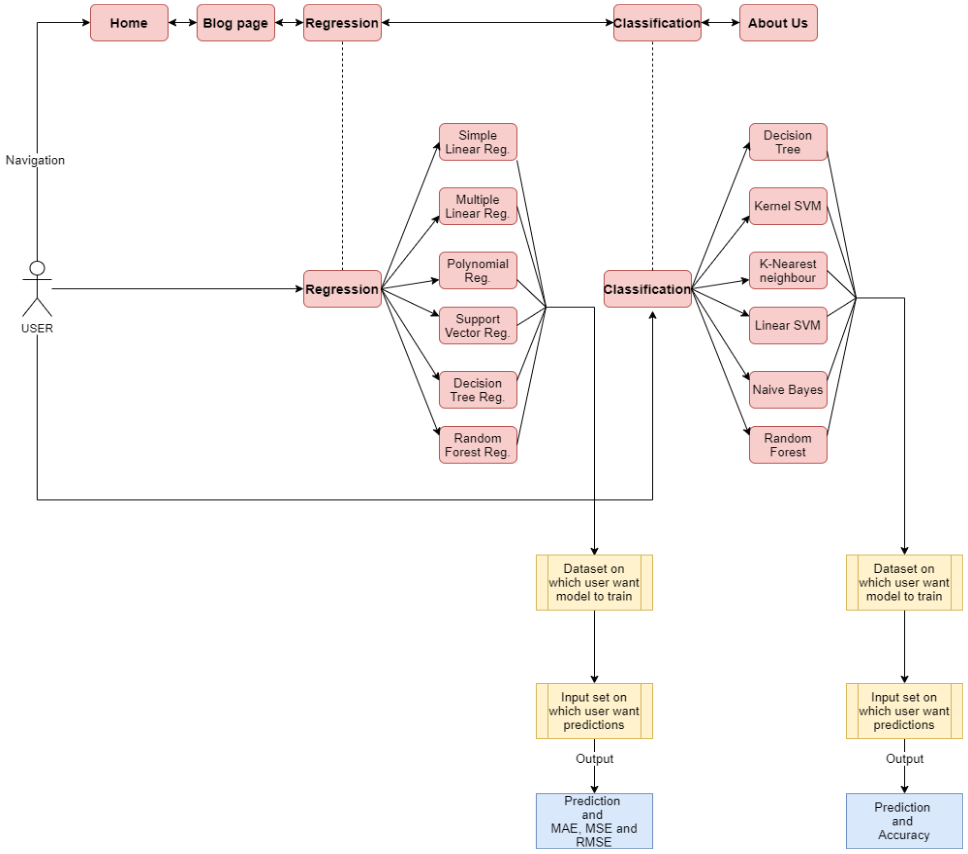


Figure 9: ER diagram of no-code platform

PROGRAMMING LANGUAGES USED

ML Modeller consists of 4 programming languages, namely: HTML, CSS, javascript, and python.

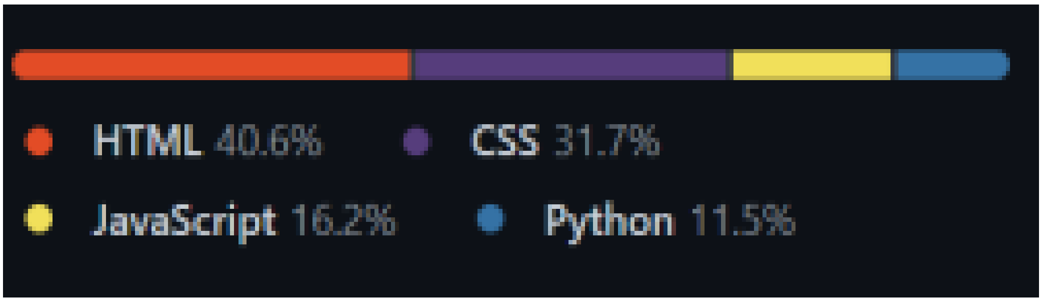


Figure 10: Line graph representing % of languages used in website

HTML used for creating the skeleton for the website, to which user can interact, an UI. CSS is used for styling of the HTML page, also used bootstrapping as well. JavaScript is used to create dynamic alerts and add DOM functionalities such as on-click or mouse hover effect. Finally, python is used for backend like, in creating machine learning codes for classification and regression models, and developing API which is the backbone of the application [6].

FUTURE PROSPECTS

In the near future we are planning to add one more service which will tell users which machine learning model suits best for their dataset by checking evaluation metrics. This service would be useful for both type of visitors, one being who does not have prior machine learning knowledge and are dependent on the website, and secondly for those who know machine learning but wants to save time on deciding a model for training [7]. Users with no prior machine learning knowledge would then not have to go to each model for raining and evaluating the accuracy and then deciding the best one. This would also attract visitors with Machine learning knowledge and ML Modeller’s market hold will expand with all types of different users [8].

THE EXISTING STATE OF THE ART

Currently, there is no such application in the market. And the existing state-of-the-art is writing machine learning code manually and then getting predictions or making observations according to one's insightfulness, this requires the knowledge of machine learning and coding. Google had launched a similar app- 'Teachable Machine', which only works on image and audio datasets [9]. Our app works on images and moreover on comma-separated files (.csv) as well, containing numeric and character values. These datasets are more common, easily available, and in more use than audio datasets (according to the current scenario) [1]. ML Modeller will be based on a no-code platform, which means even a person with no coding skills can avail of benefits. Users do not require any prior machine learning knowledge. Help in decision-making for businessmen, recruiters, analysts, students, teachers, and the list continues. Less time-consuming and more accurate predictions than manual analysis. The only benefit of writing ML code manually is that one can build data specific ml model which might perform better than the website (Higher accuracy) but only if the person knows ML enough to do so [5].

PRACTICALITY IN DIFFERENT PROFESSIONS

Teachers and People in the Education System

ML Modeller has the potential to make educators more efficient by completing tasks such as classroom management, scheduling, etc. In turn, educators are free to focus on tasks that cannot be achieved by AI, and that require a human touch. ML Modeller in the form of learning analytics can help teachers gain insight into data that cannot be gleaned by using the human brain and give conclusions that positively impact the teaching and learning process. With it, educators can make conclusions about things that may happen in the future. For instance, using a data set of middle school students’ cumulative records, predictive analytics can tell us which ones are more likely to drop out because of academic failure or even their predicted score on a standardized exam, such as the ACT or SAT. It can grade student assignments and exams more accurately than a human can. It may require some input from a human being, but the results will have higher validity and reliability [5].

Doctors and in Healthcare

Doctors can use ML Modellers to reduce the cost of healthcare, for example uploading EMR datasets on our website and getting results for specific inputs. Similarly, doctors can upload a dataset for diabetes, heart disease, pneumonia, and different types of cancer and diagnose if a person is having any of these ailments or not. The ultimate goal is to improve and provide healthcare facilities at a lower cost. Moreover, doctors can analyze oncology data, providing insights that allow oncologists, pharmaceutical companies, payers, and providers to practice precision medicine and health. Similarly, providing a dataset can predict illness and treatment to help physicians and payers intervene earlier, predict population health risk by identifying patterns and surfacing high-risk markers and model disease progression, and more [5].

Recruiters and Hiring Team

Recruiting a perfect candidate or fixing a payroll is often very tireless work. But if you have relevant past data, then an ML Modeller will help you in finding the best suit for a job and with how much you should pay him or her according to your company policies. It would become easier to identify top candidates from large candidate pools [5].

Real Estate

Since ML Modeller has the ability to analyze patterns in vast amounts of data, it can be used to make reasonable predictions of the future value of a property. For example, it can combine current market data from the marketplace and CRM as well as consider public information such as transportation network characteristics, crime rates, schools, and buying trends. The number of property attributes or market data points can exceed tens of thousands, which is definitely a kind of analysis no human analyst or market research is capable of conducting. Similarly, ML Modeller can help students of school and colleges with their projects, debate, home tasks, research, and much more [5].

So the list of people who can benefit from it is countless.

CHALLENGES FACED

Initially we also tried to add a module for deep learning in ML Modeller and was successful in doing so but soon we had to discontinue it as deep learning requires high processing speed and if you are making a budget-friendly project then it is not recommended to use deep learning in it as at the time of deploying on the cloud you have to purchase service with higher CPUs. And if you try to run it on a low CPU, as we did initially, then the time required to train the model will increase exponentially, which is again not considerable. That is why deep learning still is not much used in industries and is only limited to research work due to its cost inefficiency. The other challenge we faced was applying hyper-parameter tuning. It can be almost described as a brute force technique, as it checks all possible parameters and tells the best possible match in which models give the best performance with high accuracy. But hyper-parameter tuning is time-consuming and again requires more CPUs to process fast. Thus impractical to apply. To encounter this issue we chose to keep parameters ideal while training.

CONCLUSION

As per the present scenario, we have researched and observed there is a high demand for no-code platforms as per the performance evaluation done on ML Modeller, which is not currently available in the market. Still, stepping out as the first community to introduce this technology offering people high standard prediction experience with our automated self-learning models with the least error rate, maximum accuracy, and easy-to-function capabilities. Most people with no prior knowledge of programming and machine learning can take benefit of such applications train their dataset and get predictions. Otherwise, the existing approach was very naive, time-consuming, and less accurate as the dataset has to be analyzed manually and then taken the decision as per the individual's perspective and insight. Even after taking a decision person would have no clue that what is going to be its probability of getting correct. But in the case of no-code platforms such as ML Modeller, we not only get predictions with their corresponding accuracy rate which makes it easier to rely on. On a performance basis, the website is giving consistent growth with new visitors/users every day and engaging existing visitors/users as well as the re-visit also. Most of the users are from USA or India.

REFERENCES

[1] Teachablemachine.withgoogle.com, 2021. [Online]. Available: https://teachablemachine.withgoogle.com/.

[2] K. Eremenko, 2021. [Online]. Available: https://www.udemy.com/course/machinelearning/learn/lecture/20203260?start=0#overview.

[3] [Online]. Available: https://dash.cloudflare.com/.

[4] [Online]. Available: https://www.w3schools.com/.

[5] "ML MODELLER ! No-code platform", Mlmodeller.in, 2021. [Online]. Available: https://www.mlmodeller.in/.

[6] "Ml Modeller", 2021. [Online]. Available: https://github.com/kartikeyahl/mlmodeller_blog.

[7] "At +26.1% CAGR, Low-code Development Platform Market size is Expected to reach 65.15 Billion by 2027, says Brandessence Market Research", Brandessence Market Research And Consulting Private Limited, 2021.

[8] M. Tommasi, "Democratizing software with no-code", Makerpad Ltd., 2021.

[9] V. Afshar, "92% of IT leaders comfortable with business users using low-code tools", ZDNet, 2021.

Appendix B: Sample Code

New component

```
import React from 'react'
import { useState, useEffect } from 'react'
import Linear_regression from './Linear_regression.jsx'
import Logistic_resgression from './Logistic_resgression.jsx'
import KNN from './KNN.jsx'
import KMeans from './KMeans.jsx'
import RandomForest from './RandomForest.jsx'
import Decision_tree from './Decision_tree.jsx'
import Svm from './Svm.jsx'
import Naive_bayes from './Naive_bayes.jsx'
import Preprocess from './Preprocess.jsx'
import Input from './Input.jsx'
import CodeGeneration from './CodeGeneration.jsx'
import Output from './Output.jsx'
import ANNInput from './ANNInput.jsx'
import ANNHidden from './ANNHidden.jsx'
import ANNOutput from './ANNOutput.jsx'
const NewComp = () => {
  const
[layers,setLayers]=useState([{key:crypto.randomUUID(),type:'input',filename:'read.csv',iloc:'0',inbuilt
:'iris_plant',testsize:'30',inputrandomstate:'None'},{key:crypto.randomUUID(),type:'preprocess',scale
r:'StandardScaler'}])
  const [output,setOutput]=useState(true)
  const [ann,setAnn]=useState(false)
  useEffect(() => {
 if(layers.length===3)
    {
      setLayers((currentLayers)=>{return currentLayers.filter(layer => layer.type !== 'output')})
      setOutput(true)
    }
  }, [layers])
  console.log(layers)
  const [selectedOption, setSelectedOption] = useState('option1');
  function handleInputChange(event) {
    setSelectedOption(event.target.value);
  }
function addDecisionTree()
  {
setLayers((currentLayers)=>{return[...currentLayers,{key:crypto.randomUUID(),type:'decision_tree',s
plitter:"'best'",min_samples_split:'2',random_state:'None'}]})
    setOutput(false)
    addOutput()
  }
function changeDecisionSplitter(key,splitter)
  {
    setLayers(currentLayers => {
      return currentLayers.map(layer=>{
      if(layer.key===key){
          return {...layer,splitter}
        }
```

```
      return layer
    })
    })
  }
  function changeDecisionMinSamplesSplit(key,min_samples_split)
  {
    setLayers(currentLayers => {
      return currentLayers.map(layer=>{
      if(layer.key===key){
        if(min_samples_split==="")
            min_samples_split='2'
          return {...layer,min_samples_split}
        }
      return layer
    })
    })
  }
  function changeDecisionRandomState(key,random_state)
  {
    setLayers(currentLayers => {
      return currentLayers.map(layer=>{
      if(layer.key===key){
        if(random_state==="")
            random_state='None'
          return {...layer,random_state}
        }
      return layer
    })
    })
  }
return (
    <div className='flex flex-col'>
      <div className='m-12 flex flex-row-reverse place-content-start gap-4'>
        {
          output?
            (<div className='card1 w-96 py-2 border-dashed  gap-1 text-4xl rounded-lg background-
color1'>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addLinearRegression()}>Linear Regression</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addLogisticRegression()}>Logistic Regression</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addKNN()}>KNN</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addKMeans()}>K-means</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addRandomForest()}>Random Forest</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addDecisionTree()}>Decision Tree</button>
              <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addSvm()}>SVM</button>
```

```
            <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={()=>addNaiveBayes()}>Naive Bayes</button>
            <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={(e)=>addANN(e)}>ANN</button>
          </div>):ann?(<div className='flex flex-row'>
          <div className='card1 w-96 py-2 border-dashed  gap-1 text-4xl h-24 rounded-lg background-
color1'>
            <button className='w-80  m-1 background-color-blue rounded-lg'
onClick={(e)=>addANNLayer(e)}>Add layer</button>
          </div>
          </div>):(<div></div>)
         }
         <ul className='flex flex-row text-4xl gap-2 p-1'>
          {
            layers.map(layer=>
            {
             if(layer.type==='input')
            {
             return <Input setKey={layer.key} selectedOption={selectedOption}
handleInputChange={handleInputChange} changeInputFileName={changeInputFileName}
changeInputFileInteger={changeInputFileInteger} changeInputInbuilt={changeInputInbuilt}
changeInputTestSize={changeInputTestSize} changeInputRandomState={changeInputRandomState}/>
            }
             if(layer.type==='preprocess')
             {
               return <Preprocess setKey={layer.key} changePreprocess={changePreprocess}/>
             }
             if(layer.type==='linear_regression')
            {
             return <Linear_regression setKey={layer.key} removeLayer={removeLayer} />
            }
            if(layer.type==='logistic_regression')
            {
             return <Logistic_resgression setKey={layer.key} removeLayer={removeLayer}
changeLogisticPenalty={changeLogisticPenalty}
changeLogisticClassWeight={changeLogisticClassWeight}
changeLogisticRandomState={changeLogisticRandomState}
changeLogisticMaxIter={changeLogisticMaxIter}/>
            }
            if(layer.type==='knn')
            {
             return <KNN setKey={layer.key} removeLayer={removeLayer}
changeKNNChoice={changeKNNChoice} changeKNNAlgorithm={changeKNNAlgorithm}
changeKNNNumber={changeKNNNumber} changeKNNWeights={changeKNNWeights} />
            }
            if(layer.type==='kmeans')
            {
             return <KMeans setKey={layer.key} removeLayer={removeLayer}
changeKMeansClusterNo={changeKMeansClusterNo} changeKMeansInit={changeKMeansInit}
changeKMeansInitNo={changeKMeansInitNo} changeKMeansMaxIter={changeKMeansMaxIter}
changeKMeansRandom={changeKMeansRandom} />
```

```
        }
        if(layer.type==='randomforest')
        {
          return <RandomForest setKey={layer.key} choice={layer.choice} removeLayer={removeLayer}
changeRandomChoice={changeRandomChoice} changeRandomCriterion={changeRandomCriterion}
changeRandomEstimators={changeRandomEstimators}
changeRandomMaxFeatures={changeRandomMaxFeatures}
changeRandomMinSample={changeRandomMinSample} />
        }
        if(layer.type==='decision_tree')
        {
          return <Decision_tree setKey={layer.key} removeLayer={removeLayer}
changeDecisionSplitter={changeDecisionSplitter}
changeDecisionMinSamplesSplit={changeDecisionMinSamplesSplit}
changeDecisionRandomState={changeDecisionRandomState}/>
        }
        if(layer.type==='svm')
        {
          return <Svm setKey={layer.key} removeLayer={removeLayer} changeSvmC={changeSvmC}
changeSvmKernel={changeSvmKernel} changeSvmDegree={changeSvmDegree}
changeSvmGamma={changeSvmGamma} changeSvmRandomState={changeSvmRandomState}/>
        }
        if(layer.type==='naive_bayes')
        {
          return <Naive_bayes setKey={layer.key} removeLayer={removeLayer}
changeNaiveBayesEstimator={changeNaiveBayesEstimator} />
        }
        if(layer.type==='output'&& ann===false)
         {
          // return <Output1 setKey={layer.key}/>
           return <Output setKey={layer.key} changeOutputFileName={changeOutputFileName}/>
         }
        if(layer.type==='annInput')
        {
          return <ANNInput setKey={layer.key} removeLayer={removeLayer}
changeANNInputDim={changeANNInputDim}/>
        }
        if(layer.type==='annHidden')
        {
          return <ANNHidden setKey={layer.key} removeLayer={removeLayer}
changeANNHiddenActivation={changeANNHiddenActivation}
changeANNHiddenUnits={changeANNHiddenUnits} />
        }
        if(layer.type==='annOutput')
        {
          return <ANNOutput setKey={layer.key} removeLayer={removeLayer}
changeANNOutputActivation={changeANNOutputActivation}
changeANNOutputUnits={changeANNOutputUnits} changeANNOutputLoss={changeANNOutputLoss}
changeANNOutputOptimizer={changeANNOutputOptimizer}
changeANNOutputMetrics={changeANNOutputMetrics}
changeANNOutputBatchSize={changeANNOutputBatchSize}
```

```
changeANNOutputEpochs={changeANNOutputEpochs}
changeANNOutputFilename={changeANNOutputFilename}/>
        }
       })}
      </ul>
    </div>
    <CodeGeneration data={layers} selectedOption={selectedOption}/>
   </div>
 )
}
export default NewComp
```

Decision Tree

```
import React from 'react'
const Decision_Tree =
({setKey,changeDecisionSplitter,changeDecisionMinSamplesSplit,changeDecisionRandomState,remov
eLayer}) => {
   function changeSplitter(key,value)
     {
       changeDecisionSplitter(key,value)
     }
  return (
   <div className='card1 flex flex-col w-96 border-2 rounded-lg background-color1' key={setKey}>
     <div className='heading1 w-96 flex flex-row justify-between background-color-blue p-2'>
     Decision Tree
     <button className='text-2xl delete' onClick={e => removeLayer(setKey)}>X</button>
     </div>
     <div className='flex flex-col w-80 place-self-center p-2 rounded-lg gap-1 interior'>
       <p className='self-start'>Splitter</p>

       <select name="splitter" id="splitter" className='border-1 h-10'
onChange={e=>changeSplitter(setKey,e.target.value)}>
         <option value="'best'">best</option>
         <option value="'random'">random</option>
       </select>
       <p className='self-start'>Min Samples Split</p>
       <input type='number' className='rounded-lg'
onChange={e=>changeDecisionMinSamplesSplit(setKey,e.target.value)}></input>
       <p className='self-start'>Random State</p>
       <input type='number' className='rounded-lg'
onChange={e=>changeDecisionRandomState(setKey,e.target.value)}></input>
     </div>
   </div>
 )
}
export default Decision_Tree
```

Code Generation

```
import React from 'react'
import linearRegression from './codeGeneration/CGLinearRegression'
```

```
import SVM from './codeGeneration/CGSVM'
import logisticRegression from './codeGeneration/CGLogisticRegression'
import decisionTree from './codeGeneration/CGDecisionTree'
import kMeans from './codeGeneration/CGKMeans'
import kNN from './codeGeneration/CGKnn'
import randomForest from './codeGeneration/CGRandomForest'
import naiveBayes from './codeGeneration/CGNaiveBayes'
import { useState } from 'react'
import CodeSection from './CodeSection'
import inputs from './codeGeneration/CGInput'
import preprocess from './codeGeneration/CGPreprocess'
import outPut from './codeGeneration/CGOutput'
import axios from 'axios'
import annInput from './codeGeneration/CGANNInput.jsx'
import annHidden from './codeGeneration/CGANNHidden.jsx'
import annOutput from './codeGeneration/CGANNOutput.jsx'
const CodeGeneration = ({data,selectedOption}) => {
  const [completed,setCompleted]=useState(false)
  const [imports,setImports]=useState('')
  const [code,setCode]=useState('')
  // var imports
  // var code
  function handleCompleted()
  {
     setCompleted(!completed)
  }
function createCode(data){
  var temp1=''
  var temp2=''
  data.map(layer=>
    {  if(layer.type==='decision_tree')
       {
        var temp=decisionTree(layer)
        temp1=temp1+temp.imports
        temp2=temp2+temp.code
        setImports(temp1)
        setCode(temp2)
       }
  }
  )
       handleCompleted()
       var temp3=temp1+'\n'+temp2 ;
       console.log("hello")
       axios.defaults.withCredentials = true //since no ssl..
       const response= axios.post('http://localhost:8080/codegeneration', {
          data : {
             code:temp3,
          }})
          console.log("sent?");
}
export default CodeGeneration
```

CG Decision Tree

```
function decisionTree(layer) {
   return {imports:["from sklearn import
tree"],code:["model=tree.DecisionTreeClassifier(splitter="+layer.splitter+",min_samples_split="+layer
.min_samples_split+",random_state="+layer.random_state+" )\n"]}
}
export default decisionTree;
```

Appendix C: CO-PO And CO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	**DESCRIPTION**	**Blooms' Taxonomy Level**
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
C O1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
C O2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
C O3	3	3	3	3	3	2	2	3	2	2	2	3			2
C O4	2	3	2	2	2			3	3	3	2	3	2	2	2
C O5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.