MLAPI: A MACHINE LEARNING API TOOL FOR DATA ANALYTICS

by

Antonios Konomos

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

August, 2026

MLAPI: A MACHINE LEARNING API TOOL FOR DATA ANALYTICS

by

Antonios Konomos

APPROVED BY Iva Buljubasic

Dissertation chair

RECEIVED/APPROVED BY:

Renee Goldstein Osmic

Admissions Director

Dedication

To curious minds who believe that every answer is just the beginning of a better question.

Acknowledgements

I would like to express my sincere gratitude to my research supervisor, Dr. Hemant Palivela, for his continuous support and encouragement throughout the development of this dissertation. I am also deeply grateful to the management and staff of SSBM for granting me the opportunity to study at such a high-caliber business school. Finally, I extend my heartfelt thanks to all the participants in this research for their time and patience. Without their genuine support, the completion of my research would not have been possible. Thank you all once again.

ABSTRACT

MLAPI: A MACHINE LEARNING API TOOL FOR DATA ANALYTICS

Antonios Konomos
2026

Dissertation Chair:
Co-Chair:

This study presents the design, development, and usability evaluation of MLapi, a novel machine learning API tool aimed at facilitating access to Python-based data analysis and machine learning techniques. MLapi was developed to address the growing need for user-friendly analytical tools that bridge the gap between technical complexity and accessibility, particularly for users with limited programming experience. The system architecture integrates Microsoft Excel as a front-end interface with a PHP-based API and a Python Anaconda back-end, forming a modular and scalable three-tier structure. MLapi offers pre-configured templates for statistical and machine learning methods, automatically generating results in Jupyter Notebook format to enhance transparency and educational value.

The empirical component of the study employed the System Usability Scale (SUS) to assess perceived usability among 150 data-analytics professionals in the Greek banking sector. MLapi achieved a mean SUS score of 90.0, exceeding the benchmark of 85 typically associated with excellent usability. Principal Component Analysis (PCA) of the SUS responses revealed three latent dimensions -Complexity, Agility, and Learnability- each demonstrating high internal consistency. Statistical analysis showed no significant differences in usability perceptions across gender, age, education level, or professional experience, indicating that MLapi provides a universally accessible user experience.

These findings suggest that MLapi is both technically robust and inclusive, offering intuitive workflows and minimizing cognitive load. The tool’s integration with Excel enhances accessibility, while its educational features support gradual skill development in Python and machine learning. The study contributes to the fields of educational technology and usability engineering by validating MLapi as a scalable and effective solution for data science education. Recommendations for future research include expanding the sample population, incorporating qualitative methods, and exploring integration with additional platforms to further enhance usability and applicability.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I:

INTRODUCTION

1.1 Introduction

The development of Machine Learning (ML) and Artificial Intelligence (AI) has a long history that is connected to philosophical studies about human thinking. In ancient times, classical philosophers like Aristotle discussed ideas related to logic and reasoning (Zhang, 2022). While these philosophical foundations shaped the theoretical groundwork, significant technological progress in AI and ML emerged around World War II, when the creation of digital computers enabled researchers to test abstract ideas in practical settings (Muggleton, 2014). Among the pioneers who shaped this developing field were Alan Turing and John McCarthy. Turing's groundbreaking work in 1950, "Computing Machinery and Intelligence", set the foundational inquiry "Can machines think?" creating the Turing Test, which was used to measure machine intelligence (Turing, 1950). Several years later, McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon, organized the "Dartmouth Summer Research Project" (Moor, 2006), an event which is generally regarded as the milestone that transformed artificial intelligence from a theoretical idea into a distinct area of academic research (Moor, 2006).

Early AI systems were built on explicit rules and logical programming to perform specialized tasks such as solving mathematical problems or playing board games (Reddy and Fields, 2022). Although effective within constrained environments, these systems lacked adaptability and the capacity to learn from prior experiences. To address this limitation, researchers began exploring machine learning that allow computers to derive insights based on data than exclusively depend on predefined rules (Panch, Szolovits and Atun, 2018). The 1980s and 1990s marked a key transition with the development of neural networks and statistical learning methods, signaling a move from logic-based models to data-driven algorithms. With the rapid expansion of data availability and computing capabilities in the early 2000s, this transformation accelerated. Deep learning (an advanced subset of ML based on layered neural architectures), has since driven major progress in areas like image or speech recognition, and automated content creation (Liu et al., 2018; Zou, Han and So, 2009).

Today, ML plays an integral role in both research and daily technology use. It has become indispensable in data analysis, where algorithms help mining important insights from large and complex datasets (Nasteski, 2017). Generally, ML methods can be grouped into supervised and unsupervised learning (Harrison, 2019). Supervised techniques, including regression and classification, rely on labeled datasets for predictive modeling, whereas unsupervised approaches like clustering and dimensionality reduction extract structures and relationships from unlabeled data (Nasteski, 2017; Dike et al., 2018). The advantages of employing ML include process automation, reduced human error, the discovery of nonlinear relationships, and improved predictive accuracy (Nita, 2016). Because these algorithms can continually refine their performance with new data, they remain adaptive and scalable, making them highly effective for addressing dynamic and evolving analytical challenges.

1.2 Research Problem

In today's research landscape, data volumes are expanding rapidly across numerous scientific domains, and getting information from datasets is not only an essential component of modern research but also a major commercial opportunity (Brunton, Noack and Koumoutsakos, 2019; Eckart, Eckart and Enke, 2021; Ono and Goto, 2022).

Machine learning provides the necessary tools to find patterns in data, and support evidence-based decision-making (Injadat et al., 2021). At the same time, Python has gained recognition as a versatile and user-friendly programming language for data-related tasks (Nagpal and Gabrani, 2019; Srinath, 2017). Its straightforward syntax and extensive library ecosystem established it a preferred pick among both academic and industry users (Hodeghatta and Nayak, 2023; Chandel et al., 2022).

However, effectively using such tools often demands a solid background in statistics, machine learning principles, and advanced coding skills. Conducting data analysis remains a complex and time-intensive process, which can slow research productivity (Weisz et al., 2023).

Moreover, many novice researchers express limited confidence in their ability to apply ML approaches or Python programming, which can hinder their engagement with data-driven methodologies (Sundberg and Holmström, 2023; Cochran and Sheehan, 2023). Consequently, this skills-gap creates challenges for individuals, especially those with limited experience in programming or ML frameworks, seeking to adopt contemporary analytical tools (Wu, 2018).

1.3 Purpose of Research

In the era of data-driven research and evidence-based decision-making, data analysis has become an essential but complex task (Nita, 2016). It involves systematic exploration, transformation, and modeling of datasets to reveal patterns and derive actionable insights. As datasets continue to expand in size and complexity (Zhai, 2021), the need for sophisticated analytical systems and advanced visualization techniques has intensified (Zhai, 2021). Such tools are crucial not only for interpreting complex data but also for communicating results to diverse audiences effectively. This study explores various software applications designed for educational purposes in ML area. It offers an overview of widely adopted ML platforms frequently utilized within teaching environments. The study begins by presenting the existing educational software solutions, identifying limitations in their capacity to effectively demonstrate ML techniques and Python programming practices.

Among the various programming languages used in data science, Python has established itself as a leading choice for analytical work. Its importance stems from an extensive suite of open-source libraries that facilitate machine learning, statistical computation, and high-quality data visualization (Hodeghatta and Nayak, 2023). Libraries like Scikit-learn, Matplotlib, Pandas, and Seaborn have established Python's reputation as a preferred platform in both academic and industrial contexts (Hao and Ho, 2019; Molin, 2021). Nevertheless, despite its capabilities, Python presents a significant learning challenge for individuals with limited programming or statistical backgrounds (Ye et al., 2024). Mastering these libraries often demands substantial technical proficiency, creating barriers for students, novice researchers, and professionals from non-technical fields (Alzahrani et al., 2018; Rivers, Harpstead and Koedinger, 2016). This knowledge gap has notable consequences. The complexity of analytical workflows can delay research efficiency and restrict the adoption of advanced computational methods. As a result, valuable findings may remain undiscovered, and the full potential of data-centric research may not be realized. To mitigate these issues, the present thesis proposes the creation of a user-friendly machine learning API tool, named MLapi. The main goal of MLapi is to empower users, regardless of programming expertise, to interact seamlessly with Python and execute ML algorithms with minimal technical difficulty. Its usability will be assessed using the "System Usability Scale" (Brooke, 2013), which is a standardized and reliable framework for evaluating software accessibility and design effectiveness. Through this comprehensive evaluation, the study aims to deliver a practical educational resource that supports both conceptual understanding and applied learning in ML and Python programming.

MLapi is designed to simplify the entry into Python-based data analysis by enabling early-stage analysts to experiment with machine learning algorithms and statistical procedures without extensive coding knowledge. Functioning as both an analytical and instructional tool, MLapi include pre-configured templates for common statistical and ML methods, automatically produce visualizations such as plots and charts, and display computed outputs in a Jupyter Notebook format (Siddik, Li and Bezemer, 2025). This design not only streamlines the analytical workflow but also enhances transparency and reproducibility. Furthermore, by showing the underlying source code alongside results, MLapi serves as an educational platform that helps users progressively build their coding proficiency. With this system, researchers can efficiently perform a wide set of analyses, from simple statistical evaluations to advanced ML modeling, thereby accelerating research productivity and promoting broader participation in data science. Overall, the study outlines the conceptual framework, implementation strategy, and anticipated contributions of MLapi, emphasizing its potential to lower technical barriers and make advanced data analysis more inclusive, accessible, and impactful.

1.4 Significance of the Study

The importance of this study is based on its timely response to the increasing demand for inclusive and efficient learning tools within data science education, particularly in the areas of ML and Python programming. As data analysis becomes a core component of modern research and decision-making across various disciplines, the ability to comprehend and apply ML concepts is evolving from a specialized capability into a fundamental academic and professional skill. Nevertheless, the technical difficulty associated with Python and ML frameworks continues to discourage participation, especially among students and practitioners without prior programming experience. By assessing existing educational platforms and introducing MLapi, this study seeks to identify and subsequently bridge a critical gap in current landscape. MLapi reduces technical challenges by providing ready-to-use templates and automated data visualizations, while simultaneously promoting active learning by displaying the underlying source code in an interactive environment. Serving both as an analytical instrument and a learning tool, MLapi supports transparency, replicability, and skill development. Ultimately, the research contributes to making data science education more inclusive, enabling a wider community to engage confidently in analytical tasks and adopt data-informed approaches across diverse fields.

1.5 Market Sizing and Growth Projections

The global ML market is growing rapidly, driven by the rising demand for intelligent automation, data driven decision-making, and user-friendly AI tools. According to Grand View Research (2025a), the value of the market in 2024 was $55.8B and is anticipated to excess the $282.13B until 2030, representing a CAGR (compound annual growth rate) of 30.4%. This fast expansion is fueled by ML applications in areas like finance, healthcare, retail, and education, where organizations aim to upgrade their operations and gain a competitive advantage through predictive analytics and automation (Statista, 2025).

Similarly, the data analytics market (a key enabler of ML applications) is also experiencing strong growth. Fortune Business Insights (2025) reports that the global data analytics market will grow from $82.23Bin 2025 to $402.7B until 2032, with a CAGR of 25.5%.

The convergence of ML and data analytics is particularly impactful in banking, where compliance, fraud detection, and customer segmentation are critical (Fortune Business Insights, 2025). Education is another sector undergoing significant transformation. The AI in education market is estimated to increase from $7.05B in 2025 to $112.3B until 2034, at a remarkable CAGR of 36.02% (Precedence Research, 2024).

1.6 Cost-Benefit Estimates for Organizations

The adoption of a tool like MLapi offers organizations notable opportunities to reduce costs and enhance productivity, particularly within data-driven industries such as finance, healthcare, and education. Conventional ML processes often demand the expertise of specialized data scientists and software engineers, whose annual compensation frequently exceeds $120,000 (Glassdoor, 2025). By enabling employees without technical backgrounds to conduct complex data analysis through familiar platforms like Microsoft Excel, MLapi decreases reliance on expensive technical staff. This augmentation of ML accessibility enables considerable labor cost savings (Glassdoor, 2025), especially for SMEs (small and medium-sized enterprises) that may lack the financial capacity to sustain dedicated data science departments (Glassdoor, 2025).

Beyond lowering labor expenses, MLapi increases analytical efficiency by automating routine and resource-intensive tasks such as data cleaning, model development, and visualization. For instance, consider a mid-sized company with ten analysts, if each saves five hours weekly by utilizing MLapi, the total annual time saved would reach approximately 2,600 hours. Assuming an average hourly rate of $50 per analyst (Gartner, 2024a), these efficiency gains could translate to roughly $130,000 in annual productivity improvements. Organizations could reinvest these resources into strategic initiatives, workforce development, or infrastructure upgrades, thereby improving overall operational performance.

In addition to financial advantages, MLapi facilitates faster and more precise decision-making by providing real-time ML insights. This capability enhances outcomes across domains such as fraud detection, customer segmentation, and predictive maintenance. Within the banking industry, for example, rapid analysis of transaction data to identify anomalies can reduce fraud-related losses and strengthen compliance with regulatory mandates (McKinsey & Company, 2025a). Moreover, MLapi's foundation in open-source technologies (including Python and PHP) and its seamless integration with familiar tools like Excel, help keep implementation expenses low, offering organizations an affordable path to expand their analytics capacity without significant upfront investment.

MLapi also promotes sustainable organizational growth by nurturing internal talent development. Qualifying users with accessible, hands-on exposure to Python and ML processes, it fosters continuous learning and technical proficiency. This empowerment lessens dependence on external consultants while cultivating a culture of data literacy within the enterprise. Over time, building internal analytical expertise enhances agility, innovation, and adaptability, qualities that prepare organizations to respond effectively to shifting market conditions and technological progress (McKinsey & Company, 2025b).

1.7 Competitive Differentiation from BI Platforms

A significant advantage of MLapi is its integration with Microsoft Excel, a platform deeply embedded in both corporate and academic environments. By embedding ML functionalities directly into Excel, MLapi allows users to perform data analysis without switching between applications or mastering advanced programming, thus substantially lowering the entry threshold for analytics (Rao, 2025). This strategy reflects a wider industry shift toward embedding AI within familiar productivity environments to drive higher adoption rates and operational efficiency (Datarails, 2025). The accessibility that MLapi provides aligns with the expanding adoption of no-code and low-code development solutions, which are forecasted to account for more than 65% of application creation by 2028 (Nguyen, 2024). Designed with usability in mind, MLapi enables professionals, especially those in non-technical roles, to execute sophisticated analytical procedures through prebuilt templates and automated Jupyter Notebook outputs. These tools not only simplify analytical workflows but also act as learning frameworks that help users incrementally develop skills in Python programming and machine learning concepts. Moreover, MLapi's modular, three-layered architecture delivers both scalability and adaptability, allowing deployment across multiple environments, from standalone computers to enterprise systems, without disrupting existing technological frameworks. The capability to invoke ML services within Excel using VBA scripting demonstrates a lightweight yet robust integration approach that boosts productivity while minimizing technical complexity (Flynn, 2024).

MLapi differentiates itself by embedding advanced algorithms directly into Excel, enabling users to conduct classification, regression, and clustering analysis within their existing spreadsheet. This integration simplifies the overall analytical process and empowers users with minimal coding experience to undertake data science projects that exceed the descriptive limitations of conventional BI systems (Rao, 2025; Datarails, 2025). By making computational mechanics transparent, MLapi functions as both a powerful analytical engine and a practical learning platform, encouraging skill advancement in Python and applied machine learning. This dual-purpose design distinguishes MLapi as a hybrid system that promotes both operational efficiency and professional development (Siddik, Li and Bezemer, 2025; Flynn, 2024).

Finally, MLapi's dependence on open-source frameworks such as Python and PHP, combined with its Excel-based interface, lowers implementation costs and technical demands. Through its blend of accessibility, openness, and analytical depth, MLapi emerges as a transformative instrument for organizations aiming to democratize data science and enhance strategic decision-making capabilities (McKinsey & Company, 2025a; Gartner, 2024a).

1.8 Research Design and Questions

This research seeks to evaluate both the usability and overall effectiveness of MLapi. Usability will be assessed using the SUS (System Usability Scale), a widely accepted instrument for assessing system usage through user feedback (Brooke, 2013). The SUS contains ten items presented as questions, graded on a Likert Scale (5-point), and includes a balance of positively and negatively worded questions to minimize response bias (Bangor et al., 2008). A total of up to 150 data analytics professionals from the Greek banking sector will participate in this evaluation. Participants will view a demonstration video showcasing the MLapi system and subsequently complete the SUS questionnaire. In addition to the primary usability metrics, demographics information like gender, age, educational background, and professional experience, will be gathered to determine how these attributes may shape user perceptions. Statistical tests including t-tests, ANOVA, and correlation analyses (Ranganathan, 2021) will then be employed to detect significant deviations in SUS results among different demographic segments (Ranganathan, 2021).

To inspect the potential effect of demographic characteristics on the perceived usability of MLapi, four hypotheses are proposed.

The first hypothesis tests whether SUS scores significantly differ according to participants' gender identities, applying an independent samples t-test or a non-parametric Mann-Whitney U test (Smalheiser, 2017), depending on data distribution (Holmes and Rinaman, 2014).

The second hypothesis explores the relationship between age and SUS ratings, using Pearson or Spearman correlation coefficients as appropriate.

The third hypothesis investigates differences in SUS outcomes across distinct educational levels, employing a one-way ANOVA or Kruskal-Wallis test (Chatzi and Doody, 2024).

Finally, the fourth hypothesis assesses whether years of professional experience are associated with SUS scores, again analyzed using Pearson or Spearman correlation coefficients. Collectively, these hypotheses are designed to yield a holistic view of how demographic factors shape perceptions of MLapi's usability.

Additionally, a Principal Component Analysis (PCA) will be used to examine the underlying factor structure (Jolliffe and Cadima, 2016) within participants' responses to the SUS instrument for MLapi. This procedure will aid in reducing data dimensionality while maintaining the interpretability of the key usability dimensions. The results are anticipated to generate valuable insights into the inclusivity, design, and user-centered features of MLapi, supporting further enhancements and contributing to the creation of more accessible digital learning and analytics tools within the field of data science education.

CHAPTER II:
REVIEW OF LITERATURE

2.1 Introduction

Machine Learning, a key branch within artificial intelligence, has undergone remarkable transformations since its inception. Its progression has alternated between rapid breakthroughs and slower developmental phases, ultimately giving rise to sophisticated techniques and widespread real-world implementations. The term "machine learning" was first invented by Arthur Samuel (1959), who defined it as the ability of computers to "think" without having explicitly tailored algorithms (Mohammadi and Farsijani, 2023). Early approaches relied heavily on symbolic reasoning and rule-based mechanisms, where domain expertise was encoded into algorithms to perform specific tasks. While effective in narrow domains, such systems were inflexible and incapable of adapting to unfamiliar scenarios (Reddy and Fields, 2022). These limitations highlighted the necessity for systems able to transform row data to useful information, leading to the rise of statistical machine learning (Panch, Szolovits and Atun, 2018).

During the late 20th century, statistical machine learning introduced core algorithms such as Linear and Logistic Regression (Aityan, 2022; Maalouf, 2011), Decision Trees (McClarren, 2021), and Support Vector Machines (Mammone, Turchi and Cristianini, 2009). These models leveraged statistical principles to identify relationships in data and make predictive inferences (Panch, Szolovits and Atun, 2018). The motivation to develop such algorithms stemmed from the need to tackle complex problems that could not be solved through explicit programming (Sarker, 2021). Their success depends significantly on both the availability of data and the strength of computational capacity (Simeone, 2018; Rincon-Patino, Ramírez-González and Corrales, 2018). The explosion of data across scientific disciplines has positioned data-driven discovery as both a new research paradigm and a commercial frontier (Brunton, Noack and Koumoutsakos, 2019; Eckart, Eckart and Enke, 2021; Ono and Goto, 2022). Through advanced algorithms, ML enables the processing of massive datasets, identification of hidden patterns, and informed decision-making based on analytical outcomes (Injadat et al., 2021). ML techniques employ statistical and computational methods to learn directly from experience rather than through manual programming (Orji and Vassileva, 2022).

In the transition from the late 20th to early 21st centuries, neural networks gained renewed attention, propelled by greater hardware improvements and the existence of huge datasets (Zhou et al., 2017). Deep learning models, characterized by their multiple network layers, demonstrated exceptional proficiency in autonomously extracting complex features and representations from raw data (Linardatos, Papastefanopoulos and Kotsiantis, 2020; Khan et al., 2023).

The current research aims to examine educational software tools that facilitate the teaching of ML concepts and Python programming. This investigation seeks propose an innovative framework to bridge identified educational gaps. Consequently, numerous tools and platforms have been designed to support ML and DL education, each offering distinctive contributions to the learning process.

Recent research emphasizes the positive role of AI-based tools such as Google Colab in introductory programming courses (Llerena-Izquierdo et al., 2024). The cloud-based environment allows students to write and execute Python code effortlessly while engaging with advanced model training processes (Llerena-Izquierdo et al., 2024). This interactivity has significantly boosted student engagement and satisfaction. However, the same study notes challenges in accessibility for learners with limited computing power or internet availability, underscoring the importance of addressing equity in access to such technologies.

Google Colab has become a leading choice for teaching and experimenting with machine learning because of its ease of access and flexibility (Nelson and Hoover, 2020). It offers a free, cloud-hosted Jupyter Notebook interface that supports Python and other programming languages without requiring local installations (Ferreira et al., 2024; Sukhdeve and Sukhdeve, 2023; Bisong, 2019a). The platform provides adequate computational capacity to conduct modern AI experiments (Nelson and Hoover, 2020) and simplifies the workflow of training, modeling, testing, and deployment (Lee and Kwon, 2024). Several studies illustrate the process of establishing ML environments on Colab, thereby supporting users who lack expensive computing infrastructure (Chang and Zhang, 2022; Widiantoro et al., 2021; Moctezuma et al., 2023; Meyer et al., 2022; Han and Kwak, 2023). The tool also enables dynamic data input and retrieval, allowing researchers to focus more on analysis rather than data management (Halyal, 2019). Furthermore, Colab can be utilized to compare computational performance and evaluate algorithmic efficiency (Huang et al., 2023), while its collaborative nature facilitates real-time code sharing and interactive Python development (Ferraris et al., 2023).

TensorFlow is an open-source library (developed by Google) that supports the creation and deployment of ML and AI models (Pang et al., 2020; Singh et al., 2020). It has gained traction across industries (e-commerce, finance, etc.) for several tasks related to predictive analytics (Lakshmi, 2018; Doleck et al., 2020; Salloum et al., 2022; Grattarola and Alippi, 2021; Kalkha et al., 2023; Upasani et al., 2023; Ike et al., 2023). Its scalability and strong community support make it particularly appealing to practitioners. Abadi et al. (2016) detail TensorFlow's architecture, emphasizing its adaptability for large-scale ML applications. Novac et al. (2022) compare TensorFlow and PyTorch, revealing TensorFlow's user-friendly design and PyTorch's research-oriented flexibility (Chirodea et al., 2021). Nonetheless, a notable gap persists in the integration of such frameworks into formal curricula, as many educational institutions continue to depend on outdated programming platforms (Rovshenov and Sarsar, 2023; Demir, 2022).

Keras is an API created with Python and based on neural networks, which facilitates rapid experimentation and model prototyping (Heaton, J., 2020). It can operate over frameworks such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). Routhier et al. (2021) illustrate Keras's effectiveness in developing genomic deep learning models, emphasizing its user-friendliness and efficiency, though they also note the scarcity of specialized educational programs in this domain. Keras is recognized for its intuitive interface, which enables both beginners and professionals to build models efficiently (Chicho and Sallow, 2021; Bhalerao and Ingle, 2021; Heaton, 2020). It also provides pre-trained models for applications (ex. image recognition and NLP), which can be optimized for various projects, saving precious time and digital equipment (Zhou et al., 2024; Ma et al., 2024; Mathew and Bindu, 2020).

Scikit-learn, another open-source library for Python, is optimized for efficient data mining and analysis. It is built upon NumPy, SciPy, and Matplotlib, providing a robust foundation for statistical computing (Hao and Ho, 2019; Bisong, 2019b). The library includes diverse algorithms for Classification and Regression (ex. SVMs, random forests), as well as Clustering (k-means, DBSCAN, etc.) and cases of Dimensionality Reduction (Pedregosa et al., 2011). Its unified and user-friendly API simplifies experimentation for both novice and expert users (Zhou et al., 2023). In addition to modeling, Scikit-learn includes preprocessing functions like scaling, encoding, and missing data handling (Hao and Ho, 2019). It also provides extensive evaluation utilities (Cross-validation, GridSearch, etc.), and the corresponding metrics (Accuracy, F1-score, etc.) of model's performance (Bisong, 2019a). Raschka et al. (2020) examine the library's contribution to Python-based data science, while Saarela and Jauhiainen (2021) assess its feature importance measures for model interpretability. Despite its strengths, a persistent gap remains between theoretical instruction and practical application, with many programs offering insufficient hands-on experience.

2.2 Applications for ML education

The integration of ML and DL approaches into education introduces a range of challenges (Li et al., 2023). These include the necessity for diverse expertise, the ability to accommodate individual learning variations (Li et al., 2023), ethical implications arising from the application of AI-based systems (Madububambachu et al., 2024), and the importance of grounding instruction in suitable learning theories (Zhai et al., 2021). A significant obstacle in ML education is that many curricula assume prior programming knowledge, which is not always true for all learners. To address this gap, several educational technologies employ creative strategies to teach ML principles without requiring coding proficiency (Lee and Kwon, 2024). Such tools simplify the learning process by visually illustrating stages such as data training, model construction, evaluation, and implementation (Lee and Kwon, 2024). A variety of user-oriented tools (such as Teachable Machine, Create ML, WEKA, KNIME, and Open Mind) have been developed to make building and experimenting with ML models more accessible. Each platform provides distinct advantages, from no-code or low-code environments to advanced analytical capabilities.

Teachable Machine, created by Google Creative Lab, is an online resource that lets users develop ML models quickly without needing programming expertise (Carney et al., 2020). Its simplicity and interactive design allow students, educators, and designers to explore ML fundamentals with ease (Prasad and Manjunath, 2024). The platform supports access to AI learning by providing a visual interface for creating custom classification models (Agassi et al., 2019). In academic settings, Teachable Machine has been successfully used in AI ethics instruction and other curriculum resources (Huang et al., 2022; Hagendorff, 2020). Through its structured approach, users can better understand how ML systems operate, making complex topics more approachable (Carney et al., 2020). Despite these advantages, Teachable Machine is not designed for learning programming languages such as Python. The platform abstracts coding to emphasize conceptual comprehension, which benefits beginners but restricts opportunities for hands-on coding experience. Since developing Python proficiency involves writing, debugging, and applying ML libraries, Teachable Machine serves best for rapid experimentation rather than programming instruction.

Create ML, developed by Apple for macOS, is a tool that simplifies model training while maintaining professional-grade performance. Its interface supports users in training models without writing code, making it accessible to both novices and experts (Apple, 2025). The platform enables the training of multiple models within one project and offers functionality such as pausing, saving, and resuming sessions. Users can preview datasets to identify errors like mislabeling or inaccurate annotations (Healy, 2024). On-device computation using the CPU and GPU improves training efficiency, and built-in visualization tools allow performance evaluation (Fergus and Chalmers, 2022). Thus, Create ML offers a versatile and efficient environment for model development. Nevertheless, because the system conceals programming complexity, it does not foster Python skill acquisition. Consequently, while Create ML is excellent for model prototyping, it lacks the open coding framework required for those seeking to obtain programming skills.

WEKA (Waikato Environment for Knowledge Analysis), implemented by the University of Waikato (Jain et al., 2022), is a comprehensive open-source suite supporting various ML and data mining tasks. It offers modules for preprocessing, classification, clustering, regression, visualization, and feature selection (Jain et al., 2022; Ratra, Gulia, and Gill, 2021; Nidhya and Shah, 2023; Clarin, 2022). With its Java-based interface, WEKA is platform-independent and user-friendly. It also supports SQL database integration and can process query results directly (Chowdhury et al., 2022). Additionally, WEKA connects with Deeplearning4j, enabling deep learning applications (Lang et al., 2019). Its availability under the GNU General Public License enhances its role in education and research. However, WEKA users cannot directly access the underlying source code used in analysis, and the platform does not assist those aiming to learn Python. While it effectively supports ML exploration and analytical workflows, it offers limited value for developing programming expertise.

KNIME (Konstanz Information Miner) serves as a robust educational platform supporting data science and analytics teaching (KNIME, 2025). Its visual, low-code interface allows instructors to demonstrate data manipulation and ML processes without requiring extensive coding. Students can process the data through a drag-and-drop workflow (Berthold et al., 2013). KNIME’s modular design accommodates activities from basic cleaning to complex predictive modeling (Acito, 2023; Celik and Cinar, 2021; Ihrmark and Tyrkko, 2023). Moreover, its integration with Python and R, enables learners to combine theoretical concepts with real coding practices. A large and supportive community provides shared resources, workflows, and educational kits, promoting collaborative and continuous learning (O’Hagan and Kell, 2015). While KNIME does not automatically generate Python code from its workflows, it offers several integration methods to apply them within Python environments.

Open Mind is an online platform designed to enhance accessibility to ML algorithms by providing a simple interface for model training (Open-Mind, 2020). Built with TensorFlow, it allows users to create binary classification models by uploading datasets and training them directly in a web browser. The system also supports exporting trained datasets for continued experimentation. Its minimalistic design allows learners without technical expertise to explore ML concepts interactively. By facilitating dataset uploads and providing real-time feedback, Open Mind offers a practical introduction to ML principles for students and educators alike (Ashtari and Eydgahi, 2017; Salas-Rueda et al., 2022). Nevertheless, since the tool conceals the coding process, it is not optimal for learning Python programming. Proficiency in Python remains essential for advanced ML tasks due to its extensive library support and active community (Raschka, Patterson and Nolet, 2020; Chandel et al., 2022). Libraries like TensorFlow, PyTorch, and scikit-learn streamline model development and implementation, while tutorials and community resources facilitate learning (Xu and Frydenberg, 2021). Thus, Open Mind is highly useful for conceptual understanding but does not offer full Python-based programming experience.

Each of these ML platforms serves distinct educational and practical needs. Teachable Machine is ideal for beginners and fast experimentation; Create ML caters to Apple developers; WEKA supports research and analytical tasks; KNIME excels in comprehensive data analytics; and Open Mind encourages conceptual engagement. Choosing the most appropriate tool depends on users' goals, prior experience, and project requirements.

Table 1
Applications for ML education - Features

Tool	Features
Teachable Machine	User-Friendly Interface: Intuitive and accessible for all skill levels Real-Time Training: Provides real-time feedback during model training Export Options: Models can be exported to TensorFlow.js Versatile Inputs: Supports images, sounds, and poses for training models
Create ML	On-Device Training: Fast training on Mac using CPU and GPU Multi-model Training using different datasets in a single project Data Previews: Visualize and inspect data for issues Integration with Core ML: Seamless deployment in iOS apps
WEKA	Extensive Algorithm Library: Wide range of ML algorithms Data Visualization: Strong tools for visualizing data User Interfaces: Both graphical and command-line interfaces Java-Based: Runs on multiple platforms (Windows, macOS, Linux)
KNIME	Visual Workflow Creation: Intuitive drag-and-drop interface Python Integration: Supports Python scripts within workflows Extensive Node Library: Large collection of nodes for various data tasks
Open Mind	CAD/CAM Integration Innovative Features for optimized programming and machining Global Reach: Widely used in various industries worldwide Focus on Cognitive Development: Encourages critical thinking

2.3 Pricing and Total Cost of Ownership

When assessing ML tools for use in educational settings, it is important to evaluate both their technical features and the financial commitments they involve. The concept of TCO (Total Cost of Ownership) provides a holistic understanding of all direct and indirect expenses involved in acquiring, implementing, and maintaining a technology throughout its lifespan (Lee, 2025). This encompasses factors such as initial licensing or subscription charges, infrastructure demands, training and technical support, as well as any costs linked to system integration (Lee, 2025).

Teachable Machine by Google stands out for being entirely free, offering a highly accessible option for schools and universities operating on constrained budgets. It requires no user registration and supports local, on-device training, which removes the need for external cloud infrastructure and the costs associated with it (AIHungry, 2025). Likewise, open-source tools such as WEKA and KNIME provide extensive ML functionalities without charging licensing fees. Nonetheless, while their software is free, organizations must still consider indirect expenditures such as staff upskilling, integration with existing IT systems, and potential hardware improvements necessary to manage local data processing (Querio, 2025).

In contrast, platforms like Google Colab operate under a freemium model, offering both no-cost and subscription-based options. The free version includes limited GPU access and shorter computational sessions, while Colab Pro and Pro+ subscriptions, priced between $10 and $49 monthly, offer more powerful hardware and longer runtime durations (Google Colab, 2025). For schools or universities relying on scalable, cloud-based environments, these recurring fees can become substantial, particularly when multiple users access the service simultaneously (Google Colab, 2025).

Furthermore, low-code or no-code ML solutions, including Create ML by Apple, provide advanced capabilities but may result in higher overall ownership costs due to proprietary licensing, storage, and computing-related expenses (Cake.ai, 2025). Create ML is freely available to macOS users and allows on-device model training through CPU and GPU utilization, removing the need for cloud computing resources. However, its use is restrained to the Apple ecosystem, meaning access to Macs and sometimes additional Apple hardware such as iPhones or iPads is required for testing and deployment. Although the software itself comes with no direct charge, the TCO must include the price of Apple equipment, which can be substantial when outfitting multiple users. Moreover, deploying models developed in Create ML may also require expenses linked to Apple Developer Program memberships or App Store distribution fees (OnlyNoCode, 2025). Institutions using such tools may further face costs associated with cloud governance, security compliance, and user management (Cake.ai, 2025).

In conclusion, while free and open-source tools like Teachable Machine, KNIME, and WEKA present minimal upfront expenses, their total ownership costs may rise when training, integration, and technical support are factored in. On the contrary, commercial platforms offer enhanced scalability and functionality but require careful financial planning to manage recurring charges. Therefore, educational institutions should evaluate these trade-offs according to their technical resources, number of users, and long-term institutional strategies (Lee, 2025).

2.4 Enterprise Adoption Challenges

Although educational ML platforms like KNIME, Teachable Machine, Create ML, and WEKA provide accessible means for learning and experimentation, extending their use into enterprise contexts introduces a variety of challenges. These tools, (originally developed for educational or individual purposes) face constraints when adapted to the more complex demands of organizational data management, workflow integration, and compliance frameworks.

The primary difficulty lies in scalability and system integration. Most educational ML tools are not engineered to interface effectively with enterprise infrastructures such as ERP systems, data warehouses, or cloud-based analytics pipelines. For instance, Teachable Machine and Create ML are excellent for quick prototyping and local model training but lack essential enterprise components such as robust APIs, version management, and automated deployment capabilities (Aire, 2025). As a result, they offer limited practicality in environments requiring Continuous Integration/Continuous Deployment (CI/CD) processes and extensive team collaboration.

Another major limitation involves customization and support for advanced business logic. Low-code and no-code ML platforms often depend on preset modules and drag-and-drop functions that, while intuitive, restrict the flexibility required to tailor models for enterprise-specific scenarios. Tools such as KNIME and WEKA, for example, may have difficulty managing highly specialized workflows that demand conditional logic, real-time analytics, or sophisticated authentication mechanisms (BayTech Consulting, 2025).

Issues of data governance and security further complicate enterprise adoption. Educational ML applications generally lack enterprise-grade protection for managing sensitive data, including encryption, fine-grained access control, and adherence to legal standards like GDPR. Consequently, they are unsuitable for sectors where data integrity and regulatory compliance are non-negotiable, such as finance, healthcare, and public administration (Skuza, 2025). The absence of centralized user administration, activity monitoring, and audit capabilities also increases IT management complexity and security risk.

Organizational culture and skill deficiencies present additional obstacles. Even though these platforms aim to simplify ML processes, enterprise teams may lack the necessary expertise to use them effectively. Without targeted training, support, and alignment with strategic objectives, adoption becomes a heavy task (McGehee, 2024b; Reina-Parrado et al., 2025). Moreover, the rapid evolution of AI technologies can lead decision-makers to delay adoption within uncertainty (Center on Reinventing Public Education, 2025).

In conclusion, while educational ML tools deliver valuable pedagogical benefits, their application in large-scale enterprise environments remains delayed by constraints in scalability, integration, data protection, and organizational readiness. These limitations underline the growing demand for solutions such as MLapi, which are explicitly designed to merge the accessibility of educational platforms with the robustness required for enterprise operations.

2.5 Market Share and Industry Forecasts

The global market for educational machine learning and artificial intelligence tools is experiencing rapid growth. According to Grand View Research, the value in 2024 of AI in education industry was $5.88B and is projected to hit $32.27B until 2030, achieving a CAGR (Compound Annual Growth Rate) of 31.2% (Grand View Research, 2025b). Another report by Research and Markets, forecasts even more aggressive growth, projecting the market to reach $92.5 billion by 2033, with a CAGR of 32.6% (Research and Markets, 2025).

This expansion is fueled by the integration of AI-driven tools like smart tutoring systems, auto-grading, and virtual facilitators (Precedence Research, 2025). These technologies are being adopted across higher or academic education, and corporate training environments. In 2024, North America had the biggest market-share (38%), Europe's market-share was 30% and Asia-Pacific's share was 25%, the latter being the fastest-growing region due to increasing digital transformation initiatives in education (Precedence Research, 2025).

In terms of platform types, low-code and no-code ML platforms (which include tools like KNIME, Teachable Machine, and Create ML) are also gaining traction. The global low-code and no-code platform market was valued at $32B in 2024 and is expected to reach $207.25B by 2032, growing at a CAGR of 26.1% (Business Research Insights, 2025). These platforms are particularly attractive to educational institutions and enterprises seeking to empower non-technical users to construct ML models with minimum programming (Business Research Insights, 2025).

Despite their growing popularity, traditional educational ML tools like KNIME, WEKA, and Teachable Machine face stiff competition from enterprise-grade platforms (such as Alteryx, Dataiku, and Microsoft Azure ML), which offer more robust integration, scalability, and support for production environments. However, the simplicity and accessibility of educational tools continue to make them appealing for training, prototyping, and small-scale deployments (Gartner Peer Insights, 2025). In summary, the educational ML tools market faces exponential growth, supported by strong demand for personalized and data-driven learning through low-code platforms. These trends underscore the significance of solutions like MLapi, which aim to bridge the gap between educational accessibility and enterprise-grade functionality.

2.6 Statistical Methods

Classic statistical techniques remain essential in data analysis because of their effectiveness in identifying patterns and relationships within datasets. Consequently, these methods will be incorporated into the MLapi tool. The following sections describe each statistical approach and outline practical examples of their use:

1. **Descriptive Statistics** involves summarizing and presenting data through key measures such as mean, standard deviation, and variance. It is a foundational step in Exploratory Data Analysis (EDA), helping researchers understand the fundamental structure of a dataset. For example, descriptive statistics may be used to analyze sales figures by calculating the average sales, variability, and distribution of values (Ranganathan, 2021).
2. **T-Test** is used to evaluate whether the means of two groups differ significantly (Smalheiser, 2017). This inferential test supports hypothesis testing when comparing sample means, either against a known benchmark or between independent or paired groups. The t-test assumes approximately normal distribution and homogeneity of variances. Variants include the independent, paired, and one-sample t-tests. The output, a p-value, indicates the likelihood that observed differences are due to chance (Smalheiser, 2017). The method is frequently used in clinical studies comparing

treatment outcomes or in market research assessing customer satisfaction across two segments (Mishra et al., 2019).

3. **One-Way ANOVA** (Analysis of Variance) extends the t-test to compare the means of three or more independent groups (Chatzi and Doody, 2024). It evaluates overall variance by contrasting variation among group means with variation within groups (Chatzi and Doody, 2024). A significant ANOVA result indicates that at least one group mean differs, prompting additional post-hoc testing to identify which groups vary. This method is often applied in marketing research to examine the effectiveness of several advertising strategies on sales outcomes (Das et al., 2022).
4. **Two-Way ANOVA** is designed to analyze how two independent categorical variables influence one continuous dependent variable. The technique examines both the main effects of each factor and their interaction effect, which reveals whether one variable's impact depends on the level of another (Holmes and Rinaman, 2014). Assumptions include normality, equal variances, and independent observations. Two-way ANOVA is widely employed in experimental contexts such as agricultural studies evaluating how fertilizer type and irrigation level jointly affect crop yields (Das et al., 2022).
5. **Chi-Square Test** determines whether a statistically significant relationship exists between two categorical variables (Musa, 2017). It compares observed frequencies in each cell of a contingency table against expected frequencies under the assumption of independence. Large discrepancies between observed and expected values suggest a probable association between variables (Musa, 2017). This test is commonly used in

disciplines like social science, marketing, and biology to analyze survey data, examine distribution hypotheses, and assess model fit. Adequate sample size and random data collection are key assumptions. For instance, it can be used to evaluate associations between demographic attributes and consumer preferences (Franke et al., 2012).

6. **Linear Regression** models the relationship between a dependent variable and one or more independent predictors (Aityan, 2022). In its simplest form, Simple Linear Regression, it fits a line (expressed as $y = a + bx$) that best describes the linear association between variables, where a is the intercept and b the slope (Aityan, 2022). This technique is widely utilized for forecasting, trend analysis, and identifying key determinants under the assumptions of linearity, constant variance, and normally distributed residuals. A practical example is predicting housing prices based on property features such as floor area, location, and room count (Ghosalkar and Dhage, 2018).
7. **Cronbach's Alpha** measures the internal consistency or reliability of a collection of items within a scale or questionnaire (Kilic, 2016). It quantifies the degree to which the items collectively represent a single underlying construct. The coefficient varies between 0 and 1, where values greater than 0.7 declare satisfactory reliability, while excessively high values (above 0.9) may imply redundancy among items. This statistic is particularly relevant in psychology, education, and social science research to validate measurement instruments (Kilic, 2016). For example, it can be applied in psychometric studies to evaluate questionnaire reliability (Cho and Kim, 2015).

2.7 Machine Learning Methods

The MLapi tool will incorporate a broad range of ML techniques designed to address classification, clustering, regression, and dimensionality reduction tasks. The following provides an expanded overview of each method along with examples of how they are practically applied:

1. **Artificial Neural Networks** (ANN) are computational architectures inspired by the human brain, capable of detecting complex structures and relationships within data. They comprise interconnected nodes (neurons), each with an adjustable weight that evolves during learning. ANNs are particularly effective for problems involving high-dimensional and non-linear relationships such as image recognition, natural language processing, and predictive analytics. Training occurs iteratively as the system refines weights to minimize prediction error, improving accuracy over time (Wang and Wang, 2023; Goldberg, 2017, Dastres and Soori, 2021).
2. **AdaBoost** (Adaptive Boosting) is an ensemble algorithm that sequentially combines multiple weak learners to construct a strong predictive model (Chen et al., 2024). It operates by training several base classifiers (often decision trees) where each subsequent model concentrates on the samples misclassified by its predecessors. Weights assigned to data points are adjusted iteratively to emphasize difficult instances. The final output aggregates predictions through a weighted voting mechanism (Chen et al., 2024). AdaBoost enhances model precision and minimizes overfitting, proving highly effective in tasks like spam filtering, facial detection, and medical diagnostics (Mahkamov et al., 2024).

3. **Bagging** (Bootstrap Aggregating) improves model stability and accuracy by reducing variance through sampling. It generates multiple bootstrap datasets (samples drawn with replacement) and trains an independent model on each subset. Predictions are then combined, typically via voting for classification or averaging for regression (Lee, Ullah and Wang, 2019). Bagging works particularly well with high-variance models like decision trees; its most recognized variant is the Random Forest algorithm. It is frequently applied in credit scoring and other financial risk modeling tasks (Abdoli et al., 2023).
4. **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) identifies clusters of arbitrary shapes based on point density (Khan et al., 2014). Unlike k-means, DBSCAN does not require a predefined number of clusters. Instead, it groups points that are closely packed and labels isolated points as outliers (Khan et al., 2014). The algorithm depends on two parameters: epsilon (the neighborhood radius) and minPts (minimum points required for a cluster). Its robustness to noise makes it valuable for anomaly detection, geospatial analysis, and image segmentation (Thang and Kim, 2011).
5. **Decision Trees** are intuitive supervised learning algorithms used for both classification and regression. They split data hierarchically based on feature values, forming a tree-like structure where internal nodes represent conditions, branches represent outcomes, and leaf nodes provide final predictions (McClarren, 2021). Although easy to interpret, Decision Trees can overfit complex data; techniques such

as pruning or ensemble methods mitigate this. They are applied extensively in areas such as medical diagnostics and loan approval systems (Podgorelec et al., 2002).

6. **Extra Trees** (Extremely Randomized Trees) function similarly to Random Forests but introduce additional randomness when determining feature splits. Instead of searching for optimal thresholds, split points are selected randomly, which can enhance generalization and training efficiency (Geurts, Ernst and Wehenkel, 2006). This algorithm is particularly useful for large, high-dimensional datasets such as those encountered in genomic studies (Wolf et al., 2002).
7. **Factor Analysis of Mixed Data** (FAMD) integrates principles of Principal Component Analysis (PCA) and Multiple Correspondence Analysis (MCA) to handle datasets containing both quantitative and qualitative variables. By reducing dimensionality while maintaining variance, FAMD facilitates exploration of patterns, correlations, and clusters (Albuquerque et al., 2019). It is commonly used in social science and marketing analytics, such as identifying distinct customer profiles (Quinn, 2004).
8. **Gaussian Mixture Models** (GMM) represent data as a blend of multiple Gaussian distributions, each capturing a distinct cluster (Crouse, 2011). This probabilistic model provides flexibility by considering both covariance structure and cluster orientation, unlike k-means (Crouse, 2011). Parameters are optimized using the Expectation-Maximization (EM) algorithm. GMMs are widely used for speaker identification and image segmentation (Zeng, 2014).

9. **Hierarchical Clustering** organizes data into nested groups without requiring a predefined cluster count. It produces a dendrogram illustrating how clusters merge or divide at various similarity levels. The two primary approaches are agglomerative (bottom-up) and divisive (top-down) (Murtagh and Contreras, 2012). Hierarchical clustering is often used for social network analysis and customer segmentation (Jia et al., 2011).
10. **K-Means** partitions a dataset into k clusters by assigning each point to the nearest centroid and iteratively updating centroid positions based on cluster means (Sinaga and Yang, 2020). The process repeats until minimal centroid movement occurs. Although efficient, it assumes roughly spherical clusters and is sensitive to initial centroid placement (Sinaga and Yang, 2020). K-Means is common in retail analytics for customer segmentation and purchase behavior analysis (Kusrini, 2015).
11. **K-Medoids** resembles K-Means but selects actual data points as cluster centers, known as medoids, making the model more resistant to outliers (Kaur, Kaur and Singh, 2014). By iteratively refining medoid positions, the algorithm minimizes within-cluster dissimilarity. It is useful for clustering with non-Euclidean distance metrics, particularly in supply chain optimization (Park and Jun, 2009).
12. **K-Nearest Neighbors** (KNN) is a non-parametric algorithm for classification and regression that bases predictions on the nearest k samples, using metrics such as Euclidean distance (Halder, 2024). Class predictions rely on majority voting among neighbors, while regression outputs are averaged (Halder, 2024). Although straightforward and effective, KNN becomes computationally intensive with large

datasets. It is frequently applied in recommendation systems and fraud detection (Ganji and Mannem, 2012).

13. **K-Prototypes** extends K-Means to handle datasets containing both numeric and categorical variables by merging Euclidean and matching dissimilarity measures. The method minimizes a cost function, balancing both variable types and is particularly suited to customer segmentation and healthcare analytics (Kumar, Rani and Rao, 2017; Ji et al., 2013).
14. **LightGBM** (Light Gradient Boosting Machine), developed by Microsoft, is a high-performance gradient boosting framework optimized for speed and memory efficiency. It employs techniques such as histogram-based learning and leaf-wise tree growth to accelerate training and manage large feature sets (Ke et al., 2017). LightGBM is widely used in real-time applications, including click-through rate prediction (Zhu, 2022).
15. **Logistic Regression** is a foundational supervised learning model for binary classification that predicts the probability of a result using the logistic (sigmoid) function transforming continuous inputs into probabilities between 0 and 1 (Maalouf, 2011). Due to its interpretability and simplicity, it is a standard model in credit scoring, spam filtering, and marketing analytics (Akinci, 2007).
16. **MiniBatch K-Means** modifies the K-Means algorithm to handle large-scale datasets by processing small random subsets, or mini-batches, during centroid updates. This improves computational speed and reduces memory use while maintaining comparable accuracy (Peng, Leung and Huang, 2018). It is suitable for real-time or

large-scale clustering applications, such as analyzing social media data streams (Fitriyani and Murfi, 2016).

17. **Naive Bayes** comprises probabilistic classifiers grounded in Bayes' theorem with an independence assumption among features. Despite this simplification, it performs well in text classification, spam filtering, and sentiment analysis (Song et al., 2009). The algorithm's efficiency and robustness in high-dimensional environments make it a frequent choice in natural language processing (Wickramasinghe and Kalutarage, 2021).
18. **Principal Component Analysis** (PCA) mitigates the dimensionality by converting correlated features into orthogonal principal components that retain most of the dataset's variance (Jolliffe and Cadima, 2016). The first few components typically capture most of the information, supporting visualization and noise reduction. PCA is vital in applications like image compression and feature extraction (Tharwat, 2016).
19. **Random Forest** is a variation of decision trees which have been trained on random subgroups of samples, combining their predictions to enhance accuracy and reduce overfitting (Biau and Scornet, 2016). Its robustness makes it suitable for diverse domains such as fraud detection, risk scoring, and healthcare analytics (Bottou et al., 2015).
20. **Stochastic Gradient Descent** (SGD) is a popular optimization algorithm suitable for training ML and DL models (Newton, Yousefian and Pasupathy, 2018). By updating parameters incrementally using single samples or small batches, SGD achieves faster convergence on large datasets (Newton, Yousefian and Pasupathy, 2018). While

updates are inherently noisy, techniques such as learning rate scheduling and momentum improve convergence stability (Bottou, 2010).

21. **Support Vector Machines** (SVM) aim to detect the optimal hyperplane that divides classes, while maximizing the open space between them (Mammone, Turchi and Cristianini, 2009). Through Kernel Functions, SVM can manage both linear and nonlinear patterns (Mammone, Turchi and Cristianini, 2009). It is highly accurate in bioinformatics (including protein classification tasks) and image recognition (Byvatov and Schneider, 2003).
22. **XGBoost** (Extreme Gradient Boosting) refines typical gradient boosting, applying regularization, and efficient handling of missing data (Azmi and Baliga, 2020). These optimizations result in faster training and higher accuracy, making XGBoost a top performer for structured data and predictive modeling competitions (Shi et al., 2019).

2.8 Research Gap

The research-gap underlined in this study centers on the limitations of existing software tools designed for educational purposes in machine learning. While these tools offer a flexible and intuitive interface for creating ML models, they fall short in effectively presenting Python programming to users. This gap is particularly significant for researchers and students who are new to Python or ML methodologies, as the current tools do not adequately support the development of Python programming skills. Addressing this gap is crucial, as proficiency in Python is essential for implementing and understanding ML algorithms (Ye et al., 2024). Therefore, this thesis proposes a new tool that not only facilitates the construction of ML models but also provides robust support for learning Python programming, thereby bridging the identified gap and enhancing the overall educational experience.

2.9 Conclusion

While the development of software tools and online platforms has significantly advanced ML and DL education, several gaps remain. Ensuring equitable access to technology, integrating cutting-edge tools into curricula, offering domain-specific courses, providing practical experiences, and enhancing personalized feedback are essential steps to address these gaps. By focusing on these areas, educators can support students to confront challenges in the rapidly evolving field of ML/DL. MLapi will integrate Python with Microsoft Excel, displaying both source code and results in Jupyter Notebook format. Users without coding experience through MLapi will access advanced data analytics methods while they are using Microsoft Excel, improving gradually their knowledge in Python programming.

CHAPTER III:
METHODOLOGY

3.1 Technical Design

This section examines in-depth the technical architecture of MLapi, which is implemented through a web API built using the PHP programming language (Mitchell, 2016). PHP, a widely adopted server-side scripting language, is particularly effective for creating dynamic Web applications and services (Mitchell, 2016). Its main advantages include strong native support for developing and interacting with web APIs, along with built-in features for processing HTTP requests, parsing JSON and XML data, and integrating with databases, all of which facilitate the construction of RESTful interfaces or communication with external services (Mitchell, 2016). The API is hosted on a web server, which in turn establishes a connection with the Python environment. This way, the ML methods included in MLapi become accessible to any client application. The design of the MLapi is based on multitier architecture.

3.1.1 MLapi Architecture

The 3-tier system architecture is a widely adopted model for designing scalable and maintainable web applications. It consists of three distinct layers: the Client, the API, and the Processing Server (often referred to as the Application or Business Logic Layer) (Ford et al., 2022). The Client tier delivers the presentation layer, enabling user interactions through browsers or mobile interfaces. The Web Server tier manages HTTP requests and routes the requests to backend services. The Processing Server tier hosts the core computational logic, performing data processing, validation routines, and interactions with databases or external APIs. The 3-tier architecture offers several key benefits that make it ideal for building robust, scalable, and maintainable applications (Bass et al., 2025). MLapi implements this architecture as follows:

- **Client Tier**: Any application able to send data to MLapi and request the execution of specific ML methods. For the purposes of this study Microsoft Excel (with VBA) is used as the client application.
- **API Tier**: Provides the REST interface and transfers payloads to the Python backend.
- **Processing Tier**: Executes all ML algorithms within the Python environment.

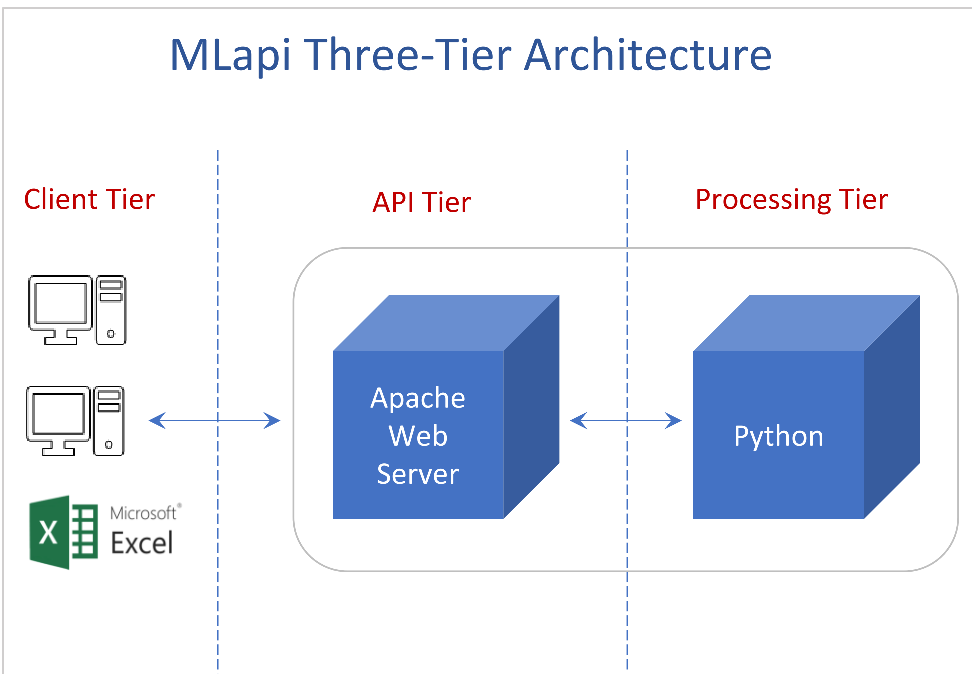


Figure 1
MLapi Architecture

3.1.2 API Tier

The API layer of MLapi is implemented using an Apache Web Server (Laurie and Laurie, 2003), deployed through the XAMPP for Windows package (MacVittie, 2006). Within this setup, the server processes HTTP requests via the PHP programming language. In the context of the MLapi, the Apache Web Server is configured to accept

HTTP requests using the POST method (Richardson and Ruby, 2008). A dedicated PHP script (see Appendix B) manages this operation handling all incoming requests. This script accepts input in JSON format (Pezoa et al., 2016) with the following keys:

Table 2
JSON Structure of MLapi Request

JSON key	**Description**
template	The name of the ML or DL method requesting to be executed
numopts	The number of additional execution options where the ML template can establish
options	An array of ON/OFF switches for each additional option of each ML template
numvars	The number of variables in the dataset
variables	An array with the name of each variable along with lists of the actual data

The API tier streamlines the request to Processing Tier, requesting Python to trigger the ML template using the data into JSON structure. Once the Python code is executed, the API tier script returns an HTTP response to the calling system. This response contains a URL (Uniform Resource Locator) pointing to the results page which is structured using the Jupyter Notebook format (Perkel, 2018). The communication between the API server and the client is illustrated in the following figure:

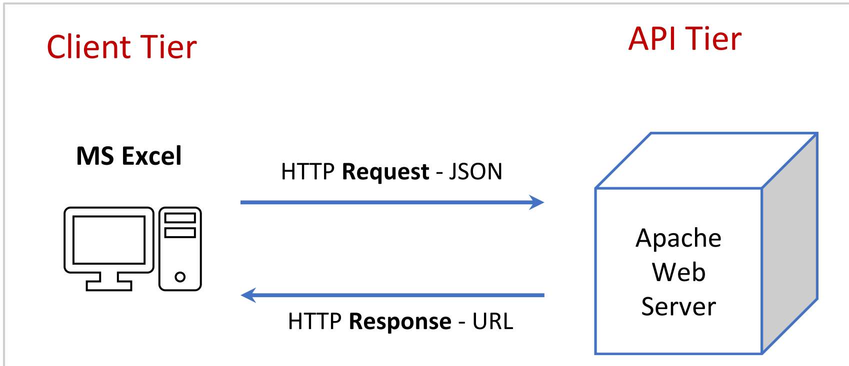


Figure 2
MLapi web service

A typical JSON request to the MLapi service is shown in the table below. In this example, the request involves the execution of the K-Nearest Neighbours (KNN) method for classification. The "options" array includes a list of switches (On/Off) that instruct the Python server to execute additional methods and metrics, such as Descriptive Statistics, Correlation Matrix, Elbow Method, Fix Imbalance (downsampling or upsampling), GridSearchCV, and Classification Scoring. The JSON request also specifies the variables of the dataset to be processed. The dependent variable, "X0", is a binary variable representing the vehicle type, with two possible levels: "Passenger" and "Car". The dataset further includes six independent variables (X1 to X6), which are either continuous or categorical, and represent key metrics such as Price, Wheelbase, Width, Length, Fuel Efficiency, and Horsepower.

Table 3
JSON Request - Example


```
{
    "numopts": 6,
    "numvars": 7,
    "options": [1,1,0,0,1,1],
    "template": "KNN-clf",
    "variables": [
```

```
{
    "datatype": "Binary",
    "id": "X0",
    "name": "VehicleType",
    "values": ["Passenger", "Car", "Passenger",...,"Car"]
},
{
    "datatype": "Continues",
    "id": "X1",
    "name": "Price",
    "values": [" 21.5", " 28.4", " 28",...," 36"]
},
{
    "datatype": "Continues",
    "id": "X2",
    "name": "Wheelbase",
    "values": [" 101.2"," 108.1"," 106.9",...," 109.9"]
},
{
    "datatype": "Continues",
    "id": "X3",
    "name": "Width",
    "values": [" 67.3"," 70.3"," 70.6",...," 72.1"]
},
{
    "datatype": "Continues",
    "id": "X4",
    "name": "FuelEfficiency",
    "values": [" 28"," 25"," 26",...," 24"]
},
{
    "datatype": "Continues",
    "id": "X5",
    "name": "Length",
    "values": [" 172.4"," 192.9"," 192",...," 189.8"]
},
{
    "datatype": "Nominal",
    "id": "X6",
    "name": "hpCat",
```

```
            "values": ["hp1","hp3","hp3",...,"hp2"]
        }
    ]
}
```

3.1.3 Processing Tier

As part of MLapi's overall architecture, the installation of a web server is specified alongside the configuration of the Python environment. These two subsystems may either coexist on the same server or be hosted on separate servers, provided they reside within the same local area network (LAN). In either configuration, communication between the two subsystems is facilitated through the creation of a symbolic link (Phillips and Preece, 2009). This link takes the form of a mapped drive in the case of Windows servers, or a mounted drive whereas Linux servers are used. The interaction between the web server and the Python environment is illustrated in the following diagram:

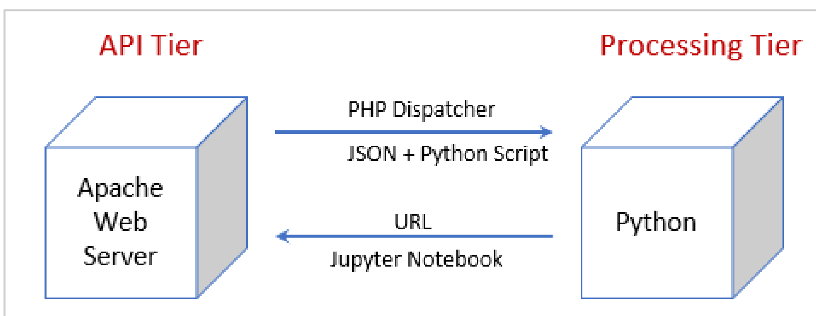


Figure 3
MLapi Processing

On the processing tier of MLapi, Python version 3.11 has been installed. More specifically, Python Anaconda distribution has been chosen to act as the ML engine that executes statistical and ML methods according to MLapi requests. Python Anaconda is a free and open-source distribution of the Python language, purely made for ML and big-data analytics (Rolon-Mérette, 2020). One of Anaconda core components is the Python interpreter, accompanied by a suite of essential libraries (NumPy, pandas, scikit-learn, etc.) providing foundational tools for numerical computations, data exploration, visualization, and ML (Yudachev et al., 2022).

One of Anaconda's main features is its ability to create isolated environments, allowing users to manage different versions of Python and packages without conflicts (Rolon-Mérette, 2020). This makes it an excellent choice for managing complex projects or collaborating across teams. Conda environments offer a robust and flexible solution for managing software dependencies in scientific computing and data science. These environments are isolated, self-contained directories that include a specific version of the Python interpreter along with all necessary libraries and dependencies (Maji, Gorenstein and Lentner, 2020). This isolation is particularly beneficial in complex projects where different tools or packages may require conflicting versions (Rolon-Mérette, 2020).

One of the most significant advantages of Anaconda environments is their support for multiple versions of Python (Hunt, 2023). This feature allows users to maintain compatibility with legacy codebases or test new features in different Python versions without disrupting their primary development setup (Hunt, 2023). Additionally, Conda simplifies the installation and management of packages, including those that are difficult to compile or have non-Python dependencies. This is achieved through the Conda package manager, which resolves dependencies automatically and supports a wide range of scientific and analytical libraries (Hunt, 2023).

Reproducibility is another key benefit of Conda environments. Users can export the configuration of an environment to a YAML file, which can then be shared or used to recreate the exact same environment on another machine. This capability is essential in collaborative research and production settings, where consistent results across different systems are critical (Maji, Gorenstein and Lentner, 2020). Furthermore, Conda environments provide a safe space for experimentation. Analysts and developers can test new packages, configurations, or workflows in a separate environment without risking the integrity of their main projects.

In summary, Conda environments represent a powerful tool for managing the complexities of modern software development in data-intensive fields (Hunt, 2023). They promote best practices in dependency management, enhance reproducibility, and support a modular approach to project development. As such, they are an indispensable component of the data science and scientific computing ecosystem. For the needs of MLapi development a Conda environment is created containing a long series of Python libraries for data analytics such as scikit-learn, matplotlib, seaborn, rjson, ggplot2, scales, dplyr, car, corrplot, caret, NbClust and xgboost. The MLapi YAML file is included in Appendix B.

3.1.4 Client Tier

For the purposes of this study Microsoft Excel has been chosen as the client tool for MLapi. Excel is part of the "Microsoft Office Suite" and is a popular application globally (Barreto, 2015). It offers a wide range of functions for data processing, along with notable capabilities for graphical data representation. Microsoft Excel owns a crucial role in the field of data analysis due to its comprehensive suite of features that support data manipulation, exploration, and visualization (Mustafy and Rahman, 2024). As a spreadsheet application, Excel provides a structured environment where data analysts can efficiently organize and manage datasets (Mustafy and Rahman, 2024). Its intuitive interface allows users to import data from various sources and perform cleaning operations such as removing duplicates, filtering records, and transforming text, which are essential steps in preparing data for analysis (Sukhdeve and Sukhdeve, 2023).

One of Excel's most significant advantages is its extensive library of built-in functions and formulas. These tools enable analysts to perform complex calculations, logical operations, and data transformations without the need for programming knowledge. Functions such as VLOOKUP, INDEX/MATCH, and SUMIFS facilitate the extraction and aggregation of data, making Excel a powerful tool for deriving insights from raw information. Excel is also outstanding in data visualization, offering a range of chart types and formatting options that help analysts communicate findings effectively. Through pivot tables and charts, users can dynamically summarize and explore large datasets, uncovering patterns and trends with ease. These features are particularly valuable in Exploratory Data Analysis, where flexibility and interactivity are crucial.

Furthermore, Excel's widespread adoption across industries ensures that data analysts can collaborate seamlessly with stakeholders who may not have access to specialized analytical tools. Its compatibility with other software platforms and its support for automation through macros and Visual Basic for Applications (VBA) enhance productivity and streamline repetitive tasks. Therefore, Microsoft Excel remains an indispensable tool for data analysts due to its versatility, accessibility, and robust analytical capabilities (Elliott et al., 2006). It serves not only as a gateway for beginners entering the field of data analysis but also as a reliable platform for professionals conducting sophisticated data-driven investigations.

It is not only Microsoft Excel's extensive features that make the integration with the MLapi particularly valuable. In addition, this integration significantly simplifies the execution of ML and DL methods, eliminating the need for complex functions which are typically required by Excel. In terms of data visualization, the collaboration of MLapi and Excel provides a fast and efficient way to generate charts and graphs. This greatly streamlines the data process and ultimately allows researchers to focus mainly on interpreting the results and formulating conclusions rather than data processing itself.

Calling a web service through Microsoft Excel using VBA is a practical and accessible method for integrating spreadsheet's data with MLapi. This approach is particularly valuable for data analysts, students, and researchers who rely on real-time or regularly updated data from online APIs. The process involves writing a VBA script that constructs and sends an HTTP request to a specific MLapi web service endpoint. Upon receiving a response, the script can parse and display the data directly within the Excel interface, enabling users to interact with dynamic content without leaving the application. The implementation begins by initializing an HTTP request object within the VBA environment. This object is configured to communicate with the desired web service using standard HTTP methods such as POST. The URL of the web service, along with any necessary headers or parameters, is specified within the script. Once the request is sent, the response in JSON is captured and is processed accordingly. The MLapi results are getting parsed, and its values are inserted into specific cells within the current worksheet, allowing immediate analysis or visualization. In parallel, an HTML page with the full content of the results is being displayed in the default web browser structured in the format of a Jupyter Notebook.

This method eliminates the need for manual data imports, reduces the risk of human error, and supports automation of repetitive tasks. Moreover, it enhances Excel's capabilities by connecting it to a broader ecosystem of web-based data services. In conclusion, leveraging VBA to call the MLapi web services from within Excel exemplifies the fusion of traditional spreadsheet tools with modern web technologies. It empowers users to build lightweight, automated data pipelines that are both powerful and user-friendly, making it a valuable technique in the toolkit of any data analyst. The VBA source code is included in Appendix B.

3.1.5 MLapi Core

By design, MLapi operates both as a data analysis tool and an educational resource. This dual purpose is achieved through a catalog of pre-configured templates for common statistical and machine learning methods. Users can select not only the desired method but also a set of supplementary or post-hoc tests, visualizations, and plots. The results are then provided in a Jupyter Notebook format. This approach allows users to access both the Python source code and the corresponding analytical results within the same document, facilitating transparency, reproducibility, and learning. More specifically, MLapi offers a comprehensive suite of templates tailored for a variety of ML tasks (classification, regression, clustering, and dimensionality reduction). These templates are designed to support both novice and experienced users by streamlining the model development process. Users are given the flexibility to apply multiple algorithms to the same dataset, enabling comparative analysis of model performance. This functionality supports informed decision-making by helping users identify the most suitable algorithm based on the nature of the data and the objectives of the analysis. Additionally, the platform encourages experimentation and iterative refinement, which are essential components of effective data science practice.

A critical component of ML process flow is data preprocessing (Sukhdeve and Sukhdeve, 2023), which MLapi handles through a series of automated and customizable steps. Prior to model training, the platform performs essential preprocessing tasks such as managing missing values, transforming categorical variables, scaling numerical features, and detecting outliers (Thomas, 2024). Users can either rely on default preprocessing pipelines or modify them to suit the specific requirements of their dataset. Furthermore, MLapi supports Exploratory Data Analysis by generating summary statistics, correlation matrices, and visual diagnostics, which help users better understand the structure and quality of their data before proceeding with modeling (Ranganathan, 2021). This integrated preprocessing framework ensures that the input data is clean, consistent, and appropriately formatted, thereby enhancing the reliability and interpretability of the analytical results (Sukhdeve and Sukhdeve, 2023).

Following model training, MLapi provides a robust framework for model evaluation, offering a variety of performance metrics (Nita, 2016) tailored to the type of analysis being conducted. For classification tasks, commonly used metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC-ROC) are computed and visualized (Nita, 2016). These metrics help assess the model's ability to correctly classify instances and balance trade-offs between false positives and false negatives (Salmanpour et al., 2025).

For regression problems, the platform reports metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2), which collectively offer insight into the model's predictive value (Salmanpour et al., 2025). In the case of clustering and dimensionality reduction, MLapi includes internal validation metrics such as silhouette score, Elbow method, and explained variance ratio, which help evaluate the quality of the clustering structure or the effectiveness of dimensionality reduction techniques (Saputra and Oswari, 2020). All evaluation results are presented alongside graphs like confusion matrices, ROC curves, residual plots, and scatter plots (Nita, 2016), enabling users to interpret model performance intuitively and make data-driven decisions for further refinement.

One-hot encoding is a fundamental preprocessing technique used in MLapi to convert categorical variables into a numerical format suitable for machine learning algorithms (Bisong, 2019b). Since most models require numerical input, categorical features must be transformed in a way that preserves their informational content without introducing ordinal relationships where none exist (Dahouda and Joe, 2021). MLapi automates this process by generating binary columns for each category within a feature (Bisong, 2019b), ensuring that the resulting dataset is both model-compatible and semantically accurate. This transformation is seamlessly integrated into the preprocessing pipeline, allowing users to focus on model development without manual data manipulation.

Class imbalance is an often issue in classification tasks, where one class significantly outnumbers others, potentially leading to biased model predictions (Harrison, 2019). MLapi addresses this challenge by offering built-in options for both down-sampling the majority class or up-sampling the minority class (Harrison, 2019). These techniques help balance the class distribution, improving the model's ability to generalize across all categories (Harrison, 2019). Users can select the appropriate strategy based on the dataset's characteristics, and MLapi applies the transformation automatically during preprocessing.

To ensure robust model evaluation and optimal hyperparameter tuning, MLapi incorporates k-fold cross-validation using GridSearchCV (Bisong, 2019b). This method divides the dataset into k subsets, training the model on k-1 folds and validating it on the remaining fold, iteratively (Bisong, 2019b). GridSearchCV systematically explores combinations of hyperparameters, selecting the configuration that yields the best performance across all folds (Harrison, 2019). This approach reduces the risk of overfitting and provides a more reliable estimate of model generalizability (Salmanpour et al., 2025). MLapi automates this process, presenting users with the best-performing model parameters along with detailed evaluation metrics, thereby enhancing both model accuracy and interpretability.

Understanding which features most influence a model's predictions is essential for interpretability and trust in machine learning. MLapi provides tools to compute and visualize feature importance, helping users identify the variables that contribute most significantly to model outcomes. For tree-based models, importance is typically derived from metrics such as Gini impurity or information gain (McClarren, 2021), while for linear models, coefficient magnitudes are used (Aityan, 2022). These insights are presented alongside visualizations such as bar plots or ranked lists, enabling users to draw meaningful conclusions and potentially refine their feature selection (Zheng and Casari, 2018). This capability supports transparency and helps in communicating results to non-technical stakeholders.

The silhouette score is a widely used internal validation metric for evaluating the quality of clustering results and measures how similar an object is to its own cluster compared to other clusters, providing a value between -1 and 1 (Sagala and Gunawan, 2022). A higher silhouette score indicates that the data points are close to the belonging cluster and far away from neighboring clusters, indicating a more coherent and well-separated grouping (Sagala and Gunawan, 2022). In MLapi, the silhouette score is automatically computed for clustering algorithms such as K-Means or Gaussian Mixture Clustering and is often visualized through silhouette plots. This helps users assess the appropriateness of the chosen number of clusters and the overall effectiveness of the clustering algorithm.

The elbow method is an empirical technique used to determine the optimal number of clusters in unsupervised learning, particularly in K-Means clustering (Sagala and Gunawan, 2022). The method plots the Within-Cluster Sum of Squares (WCSS) against the number of clusters, identifying the “elbow point” (Sagala and Gunawan, 2022), the value at which the rate of reduction sharply slows. This point represents a balance between model complexity and explained variance, indicating the most appropriate number of clusters (Sagala and Gunawan, 2022). MLapi includes an automated generation of elbow plots as part of its clustering workflow, allowing users to visually inspect and select the optimal cluster count. This method is especially useful in Exploratory Data Analysis when the true number of clusters is unknown (Sagala and Gunawan, 2022).

The Explained Variance Ratio is a key metric in dimensionality reduction techniques, and it quantifies the proportion of the dataset's total variance that is captured by each principal component (Jolliffe and Cadima, 2016). A higher Explained Variance Ratio indicates that the component retains more of the original data's information (Jolliffe and Cadima, 2016). In MLapi, this metric is used to guide users in selecting the number of components to retain, balancing dimensionality reduction with information preservation. The platform also provides visualizations such as scree plots or cumulative variance plots, helping users make informed decisions about the trade-off between model simplicity and data fidelity.

In artificial neural networks (ANNs), the selection and tuning of hyperparameters play a critical role in determining model performance (Harrison, 2019). One fundamental hyperparameter is the number of epochs (iterations) which the algorithm should be executed (De Marchi and Mitchell, 2019). While more epochs can improve learning, excessive training may lead to overfitting (Mahkamov et al., 2024), where the model performs well on training data but frequently misses unknown data. Another crucial hyperparameter is the number of hidden layers, which influences the network's capacity to uncover complex patterns (De Marchi and Mitchell, 2019). Shallow networks may underfit the data, while deeper architectures can capture more complex relationships but require careful tuning and more computational resources (De Marchi and Mitchell, 2019). To mitigate overfitting in deep networks, dropout randomly deactivates a subset of neurons, guarding the network from repeatedly selecting specific paths (Rivas, 2020). MLapi allows users to configure these hyperparameters through its neural network templates, enabling experimentation with different architectures and training strategies to optimize model performance.

In the architecture and training of artificial neural networks, several hyperparameters significantly influence the model's learning dynamics and predictive performance. Activation functions are critical for introducing non-linearity into the network, enabling it to find complex patterns beyond linear relationships (Rivas, 2020). Common choices include ReLU (Rectified Linear Unit), sigmoid, and tanh, each with distinct characteristics affecting gradient flow and convergence behavior (Rivas, 2020).

The learning rate determines the step size at each iteration during gradient descent optimization (Dastres and Soori, 2021). A learning rate that is too high may block the model from completing, while a rate that is too low can delay significantly the training and conclude to suboptimal solutions (Dastres and Soori, 2021).

Batch size defines the number of training samples which will be used to update model's internal parameters (Dastres and Soori, 2021). Smaller batch sizes can lead to noisier updates but may improve generalization, whereas larger batches offer more stable gradients and faster computation but risk overfitting (De Marchi and Mitchell, 2019).

MLapi allows users to configure these hyperparameters within its neural network templates, supporting experimentation and fine-tuning to achieve optimal learning outcomes. In the tables below there is a full list of MLapi templates covering a wide range of statistical and ML methods along with their additional options and features. The Python source code of MLapi's templates is included in Appendix B.

Table 4
Classification

MLapi Template	Options
AdaBoost-clf	Fix Imbalance Feature Importance
ANN-clf	Fix Imbalance Epochs Hidden layers Dropout (%)
Bagging-clf	Fix Imbalance
DcsTree-clf	Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1 Feature Importance
ExtraTrees-clf	Fix Imbalance Feature Importance
KNN-clf	Elbow method Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1

LightGBM-clf	Fix Imbalance Feature Importance
LogisticRegression	Evaluate Collinearity Hosmer test (GOF) ROC curve Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1
MultinomLogisticRegr	Evaluate Collinearity Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1
NaiveBayes	Fix Imbalance k-fold Cross Validation
RndForest-clf	Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1 Feature Importance
SGD-clf	Fix Imbalance GridSearchCV Scoring: ACC-PRE-RCL-f1 Linear kernel RBF kernel Sigmoid kernel Polynomial kernel Fix Imbalance
XGBoost-clf	Fix Imbalance Feature Importance

Table 5
Regression

MLapi Template	Options
AdaBoost-reg	Feature Importance
ANN-reg	Epochs Hidden layers Dropout (%)
DcsTree-reg	Feature Importance GridSearchCV Scoring: r2-MSE-MAE
ExtraTrees-reg	Feature Importance
KNN-reg	GridSearchCV Scoring: r2-MSE-MAE Elbow method
MultipleRegression	Evaluate Collinearity NCV test (homoscedasticity)
LightGBM-reg	Feature Importance
RndForest-reg	GridSearchCV Scoring: r2-MSE-MAE Feature Importance
SVM-reg	Linear kernel RBF kernel Sigmoid kernel Polynomial kernel
XGBoost-reg	Feature Importance

Table 6
Clustering

MLapi Template	Options
DBSCAN	Correlation matrix
GaussianMixture	Correlation matrix
HierClustering	Correlation matrix
KMeans	Correlation matrix
KMedoids	Correlation matrix
KPrototypes	Correlation matrix
MiniBatchKMeans	Correlation matrix

Table 7
Dimensionality Reduction

MLapi Template	Options
FAMD	Correlation matrix
PCA	Correlation matrix

Table 8
Statistics

MLapi Template	Options
Anova	TukeyHSD Levene's Test (homoscedasticity) Normality test Kruskal-Wallis
Anova2way	TukeyHSD Pairwise t-tests Levene's Test (homoscedasticity) Normality test
Chi-square	Crosstabs
LinearRegression	Correlation Pearson's Residuals Distribution NCV test (homoscedasticity)

t-test	Kolmogorov-Smirnov (normality) Levene's Test (homoscedasticity) Mann-Whitney U (non-parametric)
t-test-paired	Correlation Pearson's Kolmogorov-Smirnov (normality) Wilcoxon (non-parametric) Differences Histogram

3.1.6 Enterprise-Level Compliance and Security Framework

As MLapi transitions from a learning-focused prototype toward a system capable of supporting enterprise operations, its architectural design must increasingly reflect recognized standards for compliance, data governance, and cybersecurity. Although the current platform prioritizes educational usability, deploying MLapi in corporate environments requires a more rigorous foundation to satisfy regulatory obligations, data protection, and ensure operational reliability.

A major prerequisite for enterprise readiness is compliance with global data-protection frameworks such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and ISO/IEC 27001 (ISO/IEC, 2022; NIST, 2023a, NIST, 2023b). These regulations impose requirements including data minimization, explicit user-consent workflows, and comprehensive audit trails (ISO/IEC, 2022). Under GDPR, for example, systems may only process data essential to the intended purpose, and individuals must be informed of and agree to the ways their information will be used. In addition, organizations must maintain traceable logs that document how and when data is accessed or handled to support accountability (ISO/IEC, 2022).

In addition to regulatory compliance, robust authentication and access control mechanisms are essential. MLapi's architecture could be extended to support enterprise-grade Identity and Access Management (IAM) through the implementation of role-based access control (RBAC), which assigns permissions based on user roles such as analyst, administrator, or auditor (OWASP, 2023). Integrating Single Sign-On (SSO) capabilities through OAuth 2.0 or SAML would further reinforce security by centralizing authentication via providers like Azure Active Directory or Okta (OWASP, 2023).

Ensuring data security requires protection both during transmission and while stored. All client-server communication should take place over HTTPS with Transport Layer Security (TLS) to prevent eavesdropping and tampering. For data at rest, sensitive information should be encrypted using widely accepted standards such as AES-256, particularly in contexts involving personal or regulated data (NIST, 2023a). Secure file-handling practices, including strict input validation, sandboxing mechanisms, and restricted execution environments, are necessary to reduce threats such as injection attacks or unauthorized file operations. From an infrastructure point of view, MLapi should operate within a hardened environment. This includes disabling unnecessary system services, consistently applying security updates, and enforcing tightly controlled firewall configurations. The "National Institute of Standards and Technology" (NIST) guidelines emphasize container-security measures such as image scanning, runtime behavior monitoring, and least-privilege access policies to reduce system vulnerability (NIST, 2023b).

A data-governance framework is equally important. MLapi should define clear retention timelines, automate data disposal where appropriate, and support anonymization or pseudonymization features to limit the exposure of personal information during model development or evaluation. The "European Union Agency for Cybersecurity" (ENISA) highlights these techniques as essential for minimizing re-identification risks and maintaining compliance with modern privacy expectations (ENISA, 2024). Additionally, implementing dataset and template version control would strengthen reproducibility and simplify auditing processes by creating traceable records of changes.

Overall, enhancing MLapi with enterprise-grade compliance and security capabilities is vital for its progress into a reliable platform suitable for regulated environments. Integrating regulatory compliance, secure access management, strong encryption practices, hardened infrastructure, and comprehensive data governance would position MLapi as a robust solution capable of meeting the stringent operational and privacy requirements of professional sectors.

3.1.7 Implementation Cost Models and Resource Requirements

The implementation of MLapi is intentionally designed to minimize costs by relying on open-source technologies such as Python, Anaconda, scikit-learn, and PHP, all of which are freely available and widely supported. The use of Microsoft Excel as the client interface capitalizes on its ubiquity in academic and business environments, thereby reducing the need for additional software investments (Barreto, 2015).

The infrastructure requirements for MLapi are modest and scalable. A basic deployment can be hosted on a single machine equipped with a quad-core processor, 16

GB of RAM, and 100 GB of storage. This configuration is sufficient for small-scale educational use. For enterprise-level applications, however, it is advisable to separate the API and processing tiers across different virtual machines or containers to enhance performance and scalability. Cloud-based deployment options, such as “Amazon Web Services” or “Microsoft Azure”, offer flexible infrastructure solutions, with estimated monthly costs ranging from $50 to $150 per instance depending on computational needs (Amazon Web Services, 2025). Operational costs, including system maintenance, updates, and user support, are projected to range between $5,000 and $15,000 annually for mid-sized deployments (Gartner, 2024b).

In terms of human resources, the initial setup of MLapi requires foundational knowledge in web server configuration, Python environment management, and Excel VBA scripting. These competencies are commonly found among IT professionals and data analysts, which reduces the need for specialized training. However, organizations aiming to customize MLapi’s machine learning templates or integrate the tool with existing enterprise systems may require additional development resources. The modular architecture and comprehensive documentation are designed to streamline this process. Overall, the cost model and resource requirements illustrate that MLapi offers a low total cost of ownership and is well-positioned for scalable deployment and commercialization in diverse operational settings.

3.1.8 Risk assessment for large-scale organizational deployment

Deploying MLapi across large-scale enterprise environments introduces several risks that must be proactively managed to ensure operational stability, data integrity, and regulatory compliance. A key technical risk involves system interoperability and infrastructure compatibility. MLapi's reliance on PHP, and Python Anaconda may present challenges when integrating with enterprise-grade platforms such as cloud-native data lakes, container orchestration systems, or CI/CD pipelines. Without proper containerization and modular deployment strategies, organizations may face performance bottlenecks or increased maintenance overhead (Bass et al., 2025).

Another major concern is data governance and compliance with international regulations. MLapi must comply with standards such as the General Data Protection Regulation (GDPR), ISO/IEC 27001, and HIPAA, which require strict controls over data access, encryption, and auditability (ISO/IEC, 2022; NIST, 2023a). Failure to implement secure transmission protocols (e.g., HTTPS/TLS), role-based access control (RBAC), and anonymization techniques could expose organizations to legal liabilities and reputational risks. Integrating JSON Web Token (JWT) authentication and logging mechanisms could help ensure traceability and accountability across MLapi's architecture.

Human resources and operational risks must be considered as well. While MLapi is designed to reduce the technical barrier for non-programmers, its successful adoption depends on structured onboarding, user training, and ongoing support. Without these, users may misinterpret analytical outputs or misuse the tool, leading to faulty decision-making (Harper and Dorton, 2021). Additionally, the absence of a dedicated support team or open-source community may discourage long-term sustainability. Organizations should invest in internal capacity building and establish governance protocols to ensure responsible and consistent use of MLapi across departments (McGehee, 2024a).

3.2 Empirical Research

3.2.1 Overview of the Research Problem

System usability has been widely accepted as a key factor for successful software development. It describes how well a system allows specific users to complete specific tasks with accuracy, speed, and satisfaction in a particular usage context (Bangor et al., 2008). As software has progressed (from desktop programs to web platforms, mobile applications, and now AI-based tools) usability has continued to be an important focus. A properly designed system not only fulfills its functional goals but also allows users to interact with it easily, reducing the learning effort and minimizing possible mistakes. In current development methods, usability is often included in agile and user-centered design approaches, where repeated testing and user feedback are used to improve the user experience during the entire development process (Nemeth and Bekmukhambetova, 2023).

In software development, system usability is considered a very important factor that strongly affects user satisfaction, adoption rates, and the overall success of the system. Usability describes how easily and effectively users can work with a software product to complete their tasks. It includes several aspects such as how quickly users can learn the system, how efficiently they can use it, how well they remember its functions, how often errors occur and how serious they are, and how satisfied users feel (Hulu, Raharjo and Simanungkalit, 2024). In the case of APIs and machine learning tools, usability becomes even more important because these systems are usually used by developers and data scientists who depend on clear documentation and easy-to-use interfaces to apply the technology correctly.

Furthermore, usability is not only about the visual design or appearance of the interface. It also includes the mental power needed to use the system, how accessible it is, how fast it responds, and the complete experience of the user (Gil Urrutia et al., 2017). For technical users such as developers working with APIs or machine learning tools, usability also means having logical workflows and helpful error messages. In this situation, usability directly affects productivity and how valuable the tool is seen by users. As software systems become more complex and connected, high usability becomes necessary for user satisfaction as well as for long-term use and the ability to expand the system (Gil Urrutia et al., 2017).

This study evaluates the usability of MLapi. Although the tool may work well in terms of technical performance and accuracy, its success also depends on how easily users can understand, access, and use its features in practical situations. If usability is poor, users may not use the tool fully, support costs may increase, and users may have negative experiences, even if the technology itself is strong. For this reason, checking usability is not just an extra task but a main part of the development process, especially for tools made for technical users who need clarity and efficiency.

Even if MLapi offers good predictions, a flexible structure, and strong performance, its real value depends on how easily users can work with it. Developers and data scientists often have limited time and need tools that are not only powerful but also easy to understand and well-organized. If the API is hard to understand, not well structured, or lacks clear instructions, users may stop using it and choose other tools that are easier to use. This situation shows a common problem in technical tool development: the difference between functionality and usability (McNamara and Kirakowski, 2006). Often, APIs are made with a focus on performance, while usability is considered later. This can result in tools that are not used properly or are ignored. By making usability a main research topic, this study tries to close that gap and make sure MLapi fits the needs and thinking of its users.

3.2.2 System Usability Scale

The System Usability Scale (SUS), originally introduced by John Brooke, has become one of the most widely utilized instruments for assessing how usable a product, service, or system is (Brooke, 1996). Its ongoing popularity across research and commercial settings is largely due to its ease of use, strong psychometric foundations, and versatility. SUS is composed of ten statements rated on a 5-point Likert scale from "Strongly Disagree" to "Strongly Agree". Questions have positively and negatively worded formulations to help moderate answer bias (Brooke, 2013). Although concise, the scale reliably produces meaningful data about users' impressions of usability (Bangor et al., 2008).

A defining strength of SUS is that it can be applied in virtually any technological context. Its neutral design allows it to be used for evaluating everything from mobile apps and websites to APIs, command-line utilities, and even hardware-based interfaces. As a result, SUS has been adopted widely in various fields like healthcare, finance, education, and software engineering (Kelana et al., 2024). The tool is particularly valuable in agile development environments, where rapid iteration requires frequent and lightweight usability checks. Because the questionnaire takes only a few minutes to complete, it fits naturally into short development cycles without adding substantial overhead (Harper and Dorton, 2021).

The reliability of SUS has been validated extensively. Research shows that the scale consistently demonstrates valuable internal consistency, with Cronbach's alpha often reaching beyond 0.9, signaling that the items collectively measure similar usability construct (Lewis, 2018). SUS scores also correlate strongly with outcomes from more comprehensive usability studies, giving it credibility as an effective proxy for more detailed investigations (Lewis, 2018). Its scoring method, yielding a single usability score between 0 and 100, is straightforward to interpret and is supported by established benchmarks. Typically, scores above 68 are viewed as above average, while lower values may suggest that usability improvements are needed.

SUS is also well-suited for comparative evaluations. Researchers frequently use it to assess different design iterations or to compare a system's usability against industry references. This is particularly valuable during A/B testing or when evaluating incremental design updates, as SUS provides a standardised and repeatable metric for judging improvements (Vlachogianni and Tselios, 2022). In this way, SUS supports informed decision-making during product development and UX design, reducing reliance on intuition alone.

Despite its strengths, SUS has limitations. It captures an overall impression of usability but does not isolate the specific causes behind user frustrations or identify interface problems (Harper and Dorton, 2021). For example, a low SUS score signals a negative user experience but does not indicate whether the issue occurs from poor navigation, confusing terminology, slow performance, or other design weaknesses (Brooke, 1996). Therefore, SUS is most effective when combined with qualitative techniques such as interviews, open-ended surveys, or observational usability testing, which help uncover the root causes of user dissatisfaction (Harper and Dorton, 2021).

In summary, the System Usability Scale remains a foundational tool in usability evaluation (Bangor, Kortum, and Miller, 2008; Hulu, Raharjo and Simanungkalit, 2024; Kelana et al., 2024; Lewis, 2018; Sauro, 2011) because it is efficient, dependable, and applicable across diverse technological contexts. It offers a scalable, low-cost method for incorporating usability assessments into iterative development workflows, especially in fast-moving environments that depend on continuous feedback. While it does not replace more detailed usability analyses, it provides an essential starting point for understanding user satisfaction and guiding iterative design. When supplemented with qualitative data, SUS becomes an especially powerful instrument for creating intuitive and user-centred systems.

3.2.3 Research Purpose and Questions

This study evaluates the usability and effectiveness of MLapi. Usability is assessed through a user satisfaction survey employing the System Usability Scale. The SUS score is calculated by normalizing individual item responses and for odd-numbered (positive) questions, 1 is subtracted from the score, but for even-numbered (negative) inquiries, the score is subtracted from 5 (Sauro, 2011). The sum of these adjusted values is then multiplied by 2.5, resulting in a final score between 0 and 100 (Sauro, 2011). The entire SUS questionaire is available in Appendix A.

3.2.4 Research Design

This study explored the perceived usability of MLapi among data analytics professionals in the Greek banking sector. A representative sample of professionals participated in the research by first viewing a demonstration video of MLapi and subsequently completing the SUS questionnaire. In addition to measuring overall usability, this study collected additional information about gender, age, education level, and professional experience to examine how these factors may influence user perceptions. The goal was to assess whether demographic characteristics play a significant role in shaping usability evaluations, thereby informing future design enhancements aimed at fostering inclusivity and effectiveness. To this end, four statistical hypotheses were developed and tested using appropriate quantitative methods.

The first hypothesis examined whether there is a statistically significant difference in SUS scores between participants of different gender identities. This hypothesis was tested using an independent samples t-test and the non-parametric Mann-Whitney U test. This analysis aimed to uncover whether gender-based perceptual differences exist in evaluating MLapi's usability.

The second hypothesis explored the correlation between participants' age and their SUS scores. As age can influence both familiarity with digital tools and openness to new technologies, understanding its effect on usability perception is critical. Pearson correlation coefficients were calculated when data met the assumptions of normality, and

Spearman coefficients were used otherwise. This analysis provided insight into whether older or younger users perceive the usability of MLapi differently.

The third hypothesis addressed whether participants with varying levels of educational attainment reported significantly different SUS scores. Given that educational background may affect users' ability to engage with and evaluate educational technologies, this variable was analyzed using one-way ANOVA for normally distributed groups, or the Kruskal-Wallis test when normality and homogeneity of variance assumptions were not satisfied (Chatzi and Doody, 2024). The aim was to assess whether higher or lower educational levels corresponded with different usability experiences.

The last hypothesis investigated the relationship between years of professional experience and SUS scores. Experience in the field could influence expectations, cognitive load, and adaptability to new tools. As with age, Pearson or Spearman correlation coefficients were calculated according to the distribution of data (Aityan, 2022). This hypothesis aimed to determine whether more experienced professionals evaluated MLapi differently than those earlier in their careers.

In addition to calculating the aggregate usability score, this study utilized Principal Component Analysis (PCA) to identify underlying dimensions within the SUS data. This dual approach, combining hypothesis testing with PCA, offered a nuanced perspective on how demographics shape user perceptions of MLapi. Ultimately, these insights were intended to guide the development of more inclusive, user-centered educational technologies and provide a framework for enhancing accessibility in data science tools.

3.2.5 Data Collection

To collect data, an open invitation for participation was sent through local employee unions within the Greek banking sector. This approach ensured access to a relevant and targeted professional demographic. The primary criterion for inclusion in the study was employment in a data analytics-related positions. This requirement was designed to ensure that participants possessed sufficient familiarity with data tools and technologies, enabling them to provide informed evaluations of the MLapi system. The SUS questionnaire was processed using Google Forms. Anonymity was emphasized to promote honest and unbiased responses. All participants were receined a notice regarding the protection of personal data in accordance with applicable data privacy regulations. Furthermore, participants were advised that they could withdraw from the study at any time. The complete questionnaire instrument is available for review in Appendix A.

Prior to completing the questionnaire, participants were required to watch a demonstration video introducing the features and functionality of MLapi. The video had a duration of approximately eight minutes. Following the video, participants proceeded to complete the questionnaire, which required an estimated 7 additional minutes. Thus, the total time commitment for participation was approximately 15 to 18 minutes. This data collection procedure was designed to balance methodological rigor with participant convenience, thereby encouraging engagement while maintaining the integrity of the research process.

3.2.6 Sample size and Statistical Power

An essential component of the data collection strategy was identifying a suitable sample size to support the statistical credibility and reliability of the study's outcomes. In usability evaluation, determining an appropriate number of participants is vital because it influences statistical power and affects how confidently the results can be generalized. To estimate the sample size (Cohen, 1988), a pilot assessment using the SUS was conducted with 30 participants. This preliminary evaluation supplied an average SUS score of 89.75, which was used as the anticipated mean for the full study. The pilot findings provided an empirically grounded indication of user satisfaction and perceived usability of MLapi.

Although the pilot also produced a dataset-specific standard deviation, the study chose to apply a more conservative estimate for the final calculations. A standard deviation of 15 (commonly cited as a normative value in SUS research) was selected. Prior usability studies show that SUS score variability generally falls between 15 and 20 depending on the system and user demographics (Lewis and Sauro, 2018). Choosing a value at the lower bound of this range allowed the methodology to remain cautious while still reflecting established patterns in usability research. This decision helped avoid underestimating or overstating expected variability, ensuring that the resulting sample-size estimate remained methodologically sound.

For benchmarking purposes, a SUS score of 85 was chosen as the comparison target, as this value is frequently classified as indicative of excellent system usability (Lewis and Sauro, 2018). The analysis' primary goal was to find whether the MLapi's usability score would significantly exceed this threshold. To support a noted statistical comparison, the following parameters were applied (Cohen, 1988):

- Confidence level: 95% ($\alpha = 0.05$)
- Statistical power: 80% ($\beta = 0.20$)

These values are widely accepted in usability and behavioral sciences because they offer strong confidence in detecting meaningful effects while balancing the risks associated with Type I and Type II errors (Cohen, 1988).

Sample Size Calculation

To calculate the required sample size for comparing a sample mean to a known benchmark, the formula for a one-sample t-test was used (Cohen, 1988):

$$n = \left(\frac{\left(Z_{1-\alpha/2} + Z_{1-\beta}\right) \cdot \sigma}{\mu - \mu_0}\right)^2$$

Where:

$\mu = 89.75$ (mean from pilot survey),

$\mu_0 = 85$ (benchmark),

$\sigma = 15$ (standard deviation),

$Z_{1-\alpha/2} = 1.96$ (for 95% confidence),

$Z_{1-\beta} = 0.84$ (for 80% power).

Using the calculated parameters, the minimum size of the sample was estimated to be roughly 79 participants. To further strengthen the reliability of the analysis, the sample was expanded to 150 participants instead of adhering to the minimum threshold. This increase substantially boosted the statistical power of the one-sample t-test, raising it to approximately 97.25%. Such a high-power level reflects a strong capacity to detect a true difference between the pilot study's mean SUS score of 89.75 and the established benchmark, provided that a real difference exists within the broader population. Expanding the sample aligns with established recommendations in usability and behavioral research, as larger samples reduce the likelihood of incurring a false-negative error, known as "Type II error" (Cohen, 1988), . The accompanying graph illustrates this relationship by showing how increases in sample size led to heightened sensitivity in hypothesis testing and more robust inferential conclusions.

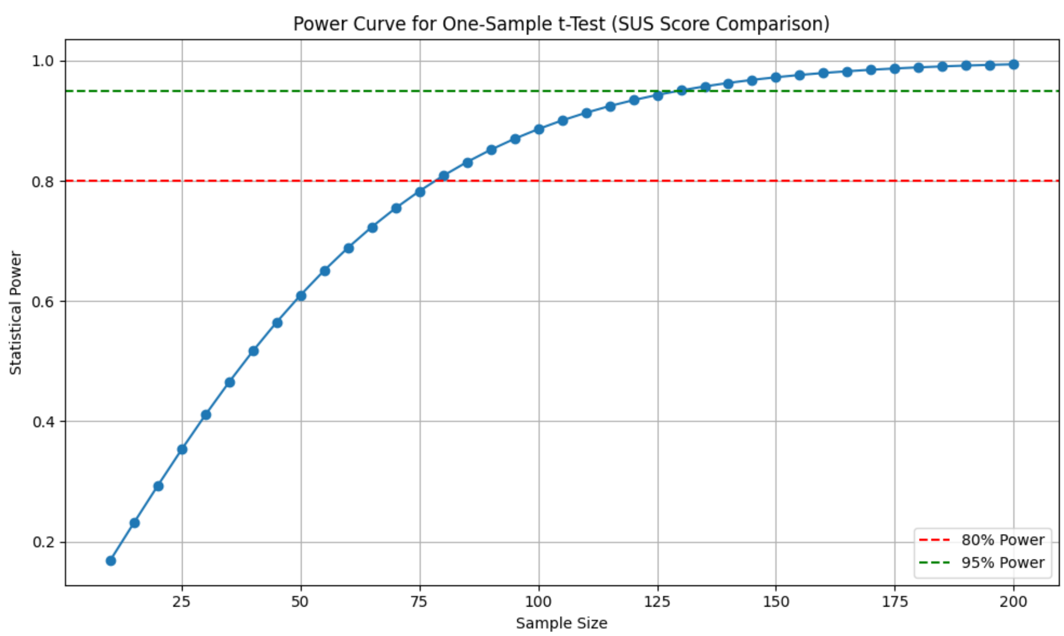


Figure 4
Statistical Power

3.2.7 Research Limitations

Although this study provides meaningful evidence regarding users' perceptions of MLapi's usability, several limitations must be recognized. While the SUS is a well-established evaluation tool, its capacity to pinpoint specific usability challenges is limited. The instrument yields a single composite score, which is useful for benchmarking purposes but does not reveal which aspects of the interface contribute most to positive or negative user experiences (Sogemeier et al., 2022). As a result, the findings offer only a broad evaluation of usability rather than insight into specific design elements that may require refinement.

A second limitation relates to the structure of the SUS questionnaire itself. The alternation between positively and negatively phrased items, combined with the use of a 5-point Likert scale, has been shown in earlier research to confuse respondents, especially individuals with limited experience completing structured surveys (Sogemeier et al., 2022). This potential confusion may have affected how consistently and accurately participants answered the items.

Finally, the study's scope, restricted to participants working in the Greek banking sector, limits the wider applicability of the results. Although the targeted sample strengthened the contextual relevance of the research, it constrains the extent to which the conclusions can be confidently extended to users from different industries, cultural backgrounds, or geographic areas. Expanding the participant demographics in future studies would help improve generalizability and provide a more diverse representation of user perspectives.

3.3 Conclusion

This chapter detailed the methodological framework used to evaluate the usability of MLapi. By adopting a mixed-methods research design, the study integrated system architecture documentation and technical design analysis with empirical data collection.

This multi-faceted approach allowed for a robust assessment of user perceptions through the System Usability Scale (Brooke, 1996). The process began by detailing the technical design of MLapi, which follows a three-tier architecture, comprising the Client, API, and Processing tiers, designed for scalability and modularity. Each tier was thoroughly described, with emphasis placed on the seamless integration between Microsoft Excel (as the client application) and the MLapi back-end, developed using PHP, Apache, and the Python Anaconda environment.

The empirical component of the methodology focused on evaluating the perceived usability of MLapi among data analytics professionals in the Greek banking sector. A pilot study was conducted to estimate the expected SUS score, yielding a mean of 89.75. To ensure the statistical robustness of the study, the sample size was increased from the primarily calculated 79 to 150 participants, thereby raising the statistical power to approximately 97.25%. This methodological decision enhanced the sensitivity of the analysis and reduced the likelihood of Type II errors.

Data was collected through a structured process that combined a demonstration video and a self-administered SUS questionnaire. Participants, recruited through local banking unions, met the criterion of holding data analytics-related positions. The inclusion of demographic variables (gender, age, education level, and professional experience) allowed for additional subgroup analyses to explore how user characteristics may influence usability perceptions.

While the methodological framework is robust, certain limitations must be acknowledged. The SUS, while efficient and widely validated, offers only a general overview of usability without diagnosing specific issues. Its alternating item format may also confuse some participants, potentially impacting data quality. Focusing solely on the Greek banking sector limits the extent to which these findings can be applied globally or to other fields.

CHAPTER IV:

RESULTS

4.1 Descriptive statistics

Descriptive statistics are an essential component of empirical research, as they provide the primary tools for summarizing, organizing, and interpreting raw datasets (Ranganathan, 2021). Prior to conducting any inferential analysis, it is crucial to gain an understanding of the dataset's fundamental properties. They are also vital for outlining the demographic profile of the participants. Variables such as age, gender, education level, and work experience are typically presented through frequency distributions and percentages, which helps explain the findings. Additionally, descriptive statistics have a diagnostic function, as they help identify which statistical tests are appropriate. Many inferential methods rely on assumptions (Aityan, 2022), and the evaluation of these assumptions informs whether parametric or non-parametric procedures should be applied, thereby strengthening the validity of subsequent conclusions (Chatzi and Doody, 2024).

The study sample covered 150 individuals, with 58.0% identifying as male and 42.0% as female. Regarding educational level, 44.7% of participants held a Bachelor's degree; 34.7% are Master's degree holders; 16.0% had completed high school, and 4.7% possessed a Doctorate. The mean participant age was 36.4 years (SD = 8.74), suggesting a predominantly early- to mid-career cohort. On average, participants reported 11.7 years of professional experience (SD = 6.97). The SUS scores were comparatively high, yielding a mean of 90.0 (SD = 7.78), which indicates excellent perceived usability of the MLapi. However, the Shapiro–Wilk normality test revealed that age, work experience, and SUS scores all significantly deviated from a normal distribution ($p < .001$), suggesting a potential need for non-parametric methods in the upcoming analyses (Aityan, 2022). All statistical outputs were generated using Jamovi v2.7.

Table 9
Descriptives

	Age	Work Experience	SUS
N	150	150	150
Missing	0	0	0
Mean	36.4	11.7	90.0
Median	35.0	10.0	90.0
Standard deviation	8.74	6.97	7.78
Skewness	0.436	0.279	-0.442
Kurtosis	-0.754	-1.16	-0.750
Shapiro-Wilk W	0.953	0.936	0.902
Shapiro-Wilk p	<.001	<.001	<.001

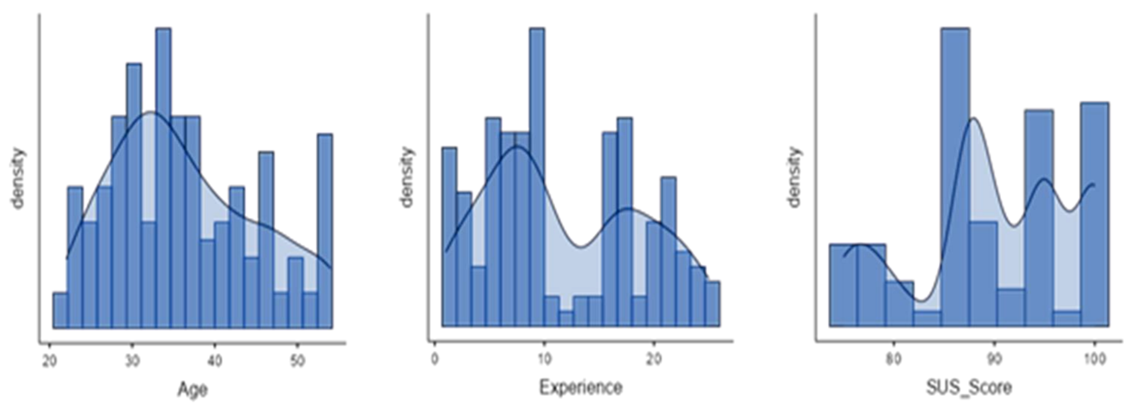


Figure 5
Histograms

Table 10
Frequencies of Gender

	Counts	% of Total
Female	63	42.0%
Male	87	58.0%

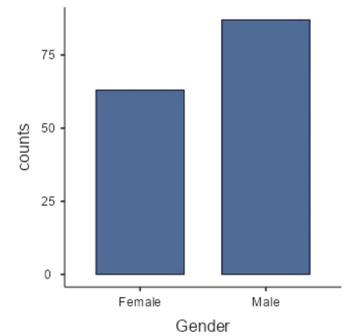
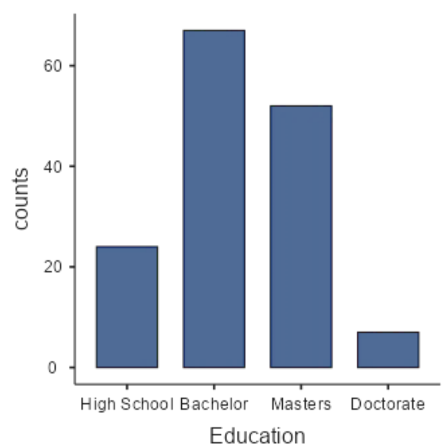


Table 11
Frequencies of Education levels

	Counts	% of Total
High School	24	16.0%
Bachelor	67	44.7%
Masters	52	34.7%
Doctorate	7	4.7%



4.2 Reliability Analysis

Reliability analysis is an essential aspect of empirical research, especially when instruments such as questionnaires or rating scales. Its purpose is to evaluate how consistently a measurement tool performs across items and respondents (Meyer, 2010). Within usability research, reliability helps ensure that the System Usability Scale provides a stable and accurate representation of users' perceptions of system usability, unaffected by random fluctuations or external influences (Brooke, 1996). Overall, conducting reliability analysis enhances confidence in the study's findings, contributes to the reproducibility of the results, and reinforces the credibility of the research. In contrast, insufficient reliability would undermine the trustworthiness of any conclusions, as observed outcomes might reflect measurement error rather than genuine differences in user experience (Meyer, 2010).

In this study, internal consistency was examined using Cronbach's alpha (Kilic, 2016). The obtained coefficient was $\alpha = 0.84$, a value commonly interpreted as evidence of good reliability. This indicates that the items within the scale function cohesively to measure a shared underlying construct (Kilic, 2016). According to established guidelines (George and Mallery, 2024), alpha values exceeding 0.80 are typically classified as good, thereby supporting the reliability of the present analysis.

Table 12
Reliability Statistics

	Cronbach's α
scale	0.840

4.2 Hypothesis testing

Hypothesis testing is a key component of the scientific process and holds significant value in both qualitative and quantitative research contexts. It offers a systematic method for extracting conclusions about the actual population using the information obtained from a sample, enabling researchers to examine theoretical propositions, verify underlying assumptions, and formulate evidence-based interpretations (Lehmann and Romano, 2005). By providing a structured approach for assessing claims, hypothesis testing contributes to the rigor, impartiality, and dependability of research findings. Fundamentally, it is a statistical procedure used to evaluate how plausible a proposed statement about a population parameter is when compared against observed data. Its importance becomes especially clear in empirical investigations, where it functions as the link between theoretical expectations and empirical observations. Through statistical evaluation, researchers to realize whether observed alterations may reflect real effects rather than random variation (Lehmann and Romano, 2005).

4.2.1 SUS score and gender

To investigate whether SUS scores differed significantly between male and female participants, an Independent Samples t-test (Smalheiser, 2017) was conducted. Prior to the analysis, assumptions of normality and homogeneity of variances were assessed (Smalheiser, 2017).

- Normality: The Shapiro-Wilk test indicated a significant deviation from normality for SUS scores, $W = 0.923$, $p < .001$, suggesting that the data are not normally distributed (Aityan, 2022).
- Homogeneity of Variances: Levene's test for equality of variances was not significant, $F(1, 148) = 2.92$, $p = 0.089$, indicating that the assumption of equal variances was met (Aityan, 2022).

Table 13
Normality Test (Shapiro-Wilk): SUS and Gender

	W	p
SUS Score	0.923	<.001

Table 14
Homogeneity of Variances Test (Levene's): SUS and Gender

	F	df	df2	p
SUS Score	2.92	1	148	0.089

Despite the violation of normality, the t-test was conducted due to its robustness to moderate deviations from normality, especially with large sample sizes (Lehmann and Romano, 2005). The results showed no statistically significant difference in SUS scores between genders: $t(148) = 1.43$, $p = 0.154$. To complement the analysis, a Mann-Whitney U test was also performed as a non-parametric alternative (Chatzi and Doody, 2024). The result did not change the conclusion.

Table 15
Independent Samples T-Test: SUS and Gender

		Statistic	df	p
SUS_Score	Student's t	1.43	148	0.154
	Mann-Whitney U	2275		0.071

These findings suggest that gender was not a statistically significant factor in perceived usability of the MLapi. Therefore, no statistically significant difference in SUS scores between male and female participants can be concluded based on the sample data.

4.2.2 SUS score and age

The second hypothesis examined whether there was a statistically significant correlation between participants' age and their SUS scores. To explore this hypothesis Pearson and Spearman coefficients were calculated. Since the assumption of normality was violated, a Pearson correlation was not the most appropriate method, therefore a

Spearman rank-order correlation was used instead. Both Pearson's correlation coefficient (r = 0.013) and Spearman's rank-order correlation (rho = 0.020) indicated a very weak and non-significant relationship between age and perceived usability. These results suggest that participants' age did not significantly influence their evaluation of the system's usability, and that the SUS scores were consistent across different age groups.

Table 16
Correlation Matrix: SUS and Age

		Age	p-value
SUS Score	Pearson's r	0.013	0.877
	Spearman's rho	0.020	0.809

4.2.3 SUS score and education level

This inquiry aimed to discover whether SUS scores differ significantly among participants with different education levels. Prior to the analysis, assumptions of normality and homogeneity of variances were assessed.

- Normality: The Shapiro-Wilk test indicated a significant deviation from normality, $W = 0.937$, $p < .001$, suggesting that the SUS scores are not normally distributed.
- Homogeneity of Variances: Levene's test was not significant, $F(3, 146) = 1.57$, $p=0.199$, indicating that the assumption of equal variances across groups was met.

Table 17
Normality Test (Shapiro-Wilk): SUS and Education

Statistic	p
0.937	<.001

Table 18
Homogeneity of Variances Test (Levene's): SUS and Education

F	df1	df2	p
1.57	3	146	0.199

One-way Analysis of Variance (ANOVA) was conducted to compare SUS scores across four education levels (Chatzi and Doody, 2024). The results showed no statistically significant differences: $F(3, 146) = 1.54$, $p = 0.207$.

Table 19
ANOVA: SUS and Education

	Sum of Squares	df	Mean Square	F	p
Education	277	3	92.2	1.54	0.207
Residuals	8748	146	59.9		

Given the violation of normality assumption, a Kruskal-Wallis test was also performed as a non-parametric alternative (Chatzi and Doody, 2024). The test similarly revealed no statistically significant differences in SUS scores across education levels.

Table 20
Kruskal-Wallis: SUS and Education

	χ^2	df	p
SUS Score	4.46	3	0.216

Pairwise comparisons showed no significant differences between any pairs of education levels (all $p > 0.17$), further supporting the conclusion that education level does not significantly influence perceived usability of the MLapi.

Table 21
Pairwise comparisons: SUS and Education

		W	p
High School	Bachelor	0.207	0.999
High School	Masters	0.918	0.916
High School	Doctorate	2.381	0.333
Bachelor	Masters	1.067	0.875
Bachelor	Doctorate	2.872	0.177
Masters	Doctorate	2.413	0.321

4.2.4 SUS score and work experience

This analysis was conducted to determine if a statistically significant correlation exists between professional experience levels and users' SUS scores for the API. Both Pearson's r and Spearman's rho (Ranganathan, 2021) were calculated to assess the relationship between SUS scores and years of professional experience.

Table 22
Correlation Matrix: SUS and Experience

		Experience	p-value
SUS Score	Pearson's r	0.013	0.871
	Spearman's rho	0.000	0.997

Neither correlation was statistically significant ($p > .05$), suggesting that professional experience does not appear to be associated with perceived usability of MLapi. These results indicate that work experience does not significantly influence SUS scores. In other words, the usability perception of the system is consistent across varying levels of professional experience.

4.3 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of data while preserving as much variability as possible (Jolliffe and Cadima, 2016). In the context of analyzing questionnaires (especially standardized ones like the System Usability Scale) PCA helps uncover latent variables, or underlying themes, that explain patterns in user responses (Jolliffe, 2002).

Instead of evaluating each question in isolation, PCA groups questions that tend to be answered similarly, revealing broader usability dimensions. This is particularly useful in SUS surveys, where some items are positively worded and others negatively, and where usability is a multi-faceted concept (Jolliffe, 2002).

PCA was conducted to explore the underlying structure of responses to the SUS questionnaire evaluating the MLapi tool. The analysis employed varimax rotation (Jolliffe and Cadima, 2016), which enhances interpretability by maximizing the variance of loadings on each component.

The decision to retain three components was guided by the scree plot and the cumulative variance explained. The scree plot revealed a clear "elbow" after the third component, indicating diminishing returns in explained variance beyond this point. Statistically, the first three components accounted for 90.3% of the total variance (Component 1: 45.5%, Component 2: 27.5%, Component 3: 17.4%).

Table 23
PCA: Component Loadings

	Component			
	1	**2**	**3**	Uniqueness
Q1_Score		0.918		0.0907
Q2_Score	0.954			0.0833
Q3_Score		0.376	0.858	0.1133
Q4_Score	0.937			0.1187
Q5_Score		0.905		0.1113
Q6_Score	0.944			0.0954
Q7_Score			0.898	0.0962
Q8_Score	0.961			0.0648
Q9_Score		0.917		0.1075
Q10_Score	0.950			0.0844

Note. 'varimax' rotation was used

Table 24
PCA: Component Statistics

Component	SS Loadings	% of Variance	Cumulative %
1	4.55	45.5	45.5
2	2.75	27.5	73.0
3	1.74	17.4	90.3

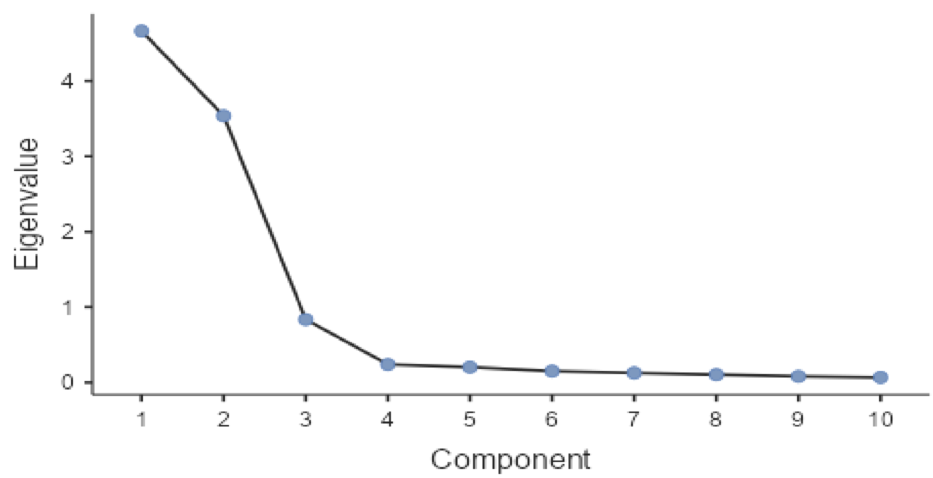


Figure 6
PCA Scree Plot

Each component was interpreted based on the pattern of loadings from the SUS items:

- Component 1: **Complexity**

  This component had high loadings from items Q2, Q4, Q6, Q8, and Q10, which include statements such as "I found the system unnecessarily complex", "I think I would need the support of a technical person", and "I needed to learn a lot of things before I could get going." These items reflect perceived difficulty, inconsistency, and reliance on external help, justifying the label Complexity. This component captures the barriers to usability and the cognitive load imposed by the system.

- Component 2: **Agility**

  Items loading highly on this component include Q1, Q5, and Q9, such as "I think that I would like to use this system frequently", "I found the various functions in this system were well integrated", and "I felt very confident using the system." These items reflect confidence, system integration, and willingness to engage, which are indicative of a system that supports efficient and fluid interaction. The term Agility is used here to denote the system's perceived responsiveness and the user's ability to operate it with ease and confidence.

- Component 3: **Learnability**

  This component was defined by high loadings from items Q3 and Q7, including "I thought the system was easy to use" and "I would imagine that most people would learn to use this system very quickly." These items directly assess the ease of learning and intuitiveness of the system, supporting the label Learnability. This dimension is critical in early adoption phases, especially for technical tools like APIs, where steep learning curves can delay user engagement.

4.3.1 Components Reliability

Following the PCA of a System Usability Scale questionnaire, reliability analysis was conducted to assess the internal consistency of the identified components (Meyer, 2010). Reliability was measured using Cronbach's alpha (Meyer, 2010) and the results revealed excellent reliability for all three components:

Table 25
PCA: Components Reliability

Component	Cronbach's α
Complexity	0.942
Agility	0.975
Learnability	0.880

Cronbach alpha values go beyond the commonly accepted threshold of 0.70 for acceptable reliability, with values above 0.90 indicating exceptional internal consistency (George and Mallery, 2024). The Complexity component, composed of negatively worded items such as perceived difficulty and need for support, demonstrates a strong and coherent measurement of the absence of significant usability barriers. The Agility component, which includes items related to confidence, integration, and frequent use, shows the highest reliability, suggesting that users consistently perceive these aspects as interconnected indicators of system responsiveness and usability. Lastly, the Learnability component, which captures quick adoption and ease of use, also shows strong reliability, confirming that these items effectively measure the intuitiveness of the system.

4.3.2 Complexity

Complexity and gender

To investigate whether Complexity differed significantly between male and female participants, an Independent Samples t-test was conducted (Smalheiser, 2017). Prior to the analysis, assumptions of normality and homogeneity of variances were assessed (Smalheiser, 2017).

- Normality: The Shapiro-Wilk test indicated a significant deviation from normality for Complexity, $W = 0.778$, $p < .001$, suggesting that the data are not normally distributed.
- Homogeneity of Variances: Levene's test for equality of variances was not significant, $F(1, 148) = 7.31$, $p = 0.008$, indicating that the assumption of equal variances was not met.

Table 26
Normality Test (Shapiro-Wilk): Complexity and gender

	W	p
Complexity	0.778	<.001

Table 27
Homogeneity of Variances Test (Levene's): Complexity and gender

	F	df	df2	p
Complexity	7.31	1	148	0.008

Table 28
Independent Samples T-Test: Complexity and gender

		Statistic	df	p
Complexity	Student's t	1.68	148	0.094
	Mann-Whitney U	2303		0.093

To complement the analysis, a Mann-Whitney U test was also performed as a non-parametric alternative (Smalheiser, 2017). These findings suggest that gender was not a statistically significant factor in the Complexity of the MLapi.

Complexity and age

The next hypothesis examined whether there was a statistically significant (Aityan, 2022) correlation between participants' age and Complexity. To explore this hypothesis Pearson and Spearman metrics were calculated. Both correlation coefficients indicated a very weak and non-significant relationship between age and Complexity.

Table 29
Correlation Matrix: Complexity and age

		Age	p-value
Complexity	Pearson's r	0.001	0.993
	Spearman's rho	-0.033	0.687

Complexity and education level

This statistical test aimed to discover whether Complexity differs significantly among participants with different education levels. Prior to the analysis, assumptions of normality and homogeneity of variances were assessed (Chatzi and Doody, 2024). The results showed no statistically significant differences.

Table 30
Normality Test (Shapiro-Wilk): Complexity and education

Statistic	p
0.748	<.001

Table 31
Homogeneity of Variances Test (Levene's): Complexity and education

F	df1	df2	p
1.33	3	146	0.267

Table 32
ANOVA: Complexity and education

	Sum of Squares	df	Mean Square	F	p
Education	0.492	3	0.164	0.161	0.922
Residuals	148.508	146	1.017		

Given the violation of normality assumption, a Kruskal-Wallis test was also performed as a non-parametric alternative (Chatzi and Doody, 2024). The test similarly revealed no statistically significant differences.

Table 33
Kruskal-Wallis: Complexity and education

	χ^2	df	p
Complexity	0.211	3	0.976

Complexity and work experience

This test aimed to examine whether there is a statistically significant relationship between participants' work experience and Complexity of the API. Both Pearson's r and Spearman's rho were calculated (Aityan, 2022). Neither correlation was statistically significant ($p > .05$), suggesting that professional experience does not appear to be associated with Complexity of MLapi.

Table 34
Correlation Matrix: Complexity and experience

		Experience	p-value
Complexity	Pearson's r	-0.015	0.857
	Spearman's rho	-0.060	0.464

4.3.3 Agility

Agility and gender

To investigate whether Agility differed significantly between the genders of participants, an Independent Samples t-test (Smalheiser, 2017) was conducted.

Table 35
Normality Test (Shapiro-Wilk): Agility and gender

	W	p
Agility	0.835	<.001

Table 36
Homogeneity of Variances Test (Levene's): Agility and gender

	F	df	df2	p
Agility	2.02	1	148	0.157

Table 37
Independent Samples T-Test: Agility and gender

		Statistic	df	p
Agility	Student's t	-0.268	148	0.789
	Mann-Whitney U	2471		0.301

To complement the analysis, a Mann-Whitney U test was also performed as a non-parametric alternative (Smalheiser, 2017). These findings suggest that gender was not a statistically significant factor in the Agility of the MLapi.

Agility and age

This hypothesis examined whether there was a statistically significant correlation between participants' age and Agility. Both Pearson and Spearman correlation coefficients (Aityan, 2022) indicated a non-significant relationship between age and Agility.

Table 38
Correlation Matrix: Agility and age

		Age	p-value
Agility	Pearson's r	0.073	0.375
	Spearman's rho	0.119	0.146

Agility and education level

This statistical test was intended to discover whether Agility differs significantly among participants with different education levels. Prior to the analysis, assumptions of normality and homogeneity of variances were assessed (Holmes and Rinaman, 2014).

Table 39
Normality Test (Shapiro-Wilk): Agility and education

Statistic	p
0.885	<.001

Table 40
Homogeneity of Variances Test (Levene's): Agility and education

F	df1	df2	p
6.89	3	146	<.001

One-way ANOVA was conducted to compare Agility across the four education levels.

Table 41
ANOVA: Agility and education

	Sum of Squares	df	Mean Square	F	p
Education	7.76	3	2.586	2.67	0.050
Residuals	141.24	146	0.967		

Given the violation of normality assumption, a Kruskal-Wallis test was also performed as a non-parametric alternative (Chatzi and Doody, 2024). The test similarly revealed no statistically significant differences.

Table 42
Kruskal-Wallis: Agility and education

	χ^2	df	p
Agility	7.02	3	0.071

Agility and work experience

This test aimed to examine whether there is a statistically significant relationship between participants' work experience and Agility of the MLapi. Both Pearson's r and Spearman's rho were calculated (Aityan, 2022). Neither correlation was statistically significant ($p > .05$), suggesting that professional experience does not appear to be associated with Agility of MLapi.

Table 43
Correlation Matrix: Agility and experience

		Experience	p-value
Agility	Pearson's r	0.102	0.212
	Spearman's rho	0.130	0.113

4.3.4 Learnability

Learnability and gender

To investigate whether Learnability differed significantly between the genders of participants, an Independent Samples t-test (Smalheiser, 2017) was conducted.

Table 44
Normality Test (Shapiro-Wilk): Learnability and gender

	W	p
Learnability	0.855	<.001

Table 45
Homogeneity of Variances Test (Levene's): Learnability and gender

	F	df	df2	p
Learnability	0.901	1	148	0.344

Table 46
Independent Samples T-Test: Learnability and gender

		Statistic	df	p
Learnability	Student's t	0.987	148	0.325
	Mann-Whitney U	2358		0.142

To complement the analysis, a Mann-Whitney U test was also performed as a non-parametric alternative (Smalheiser, 2017). These findings suggest that gender was not a statistically significant factor in the Learnability of the MLapi.

Learnability and age

This hypothesis examined whether there was a statistically significant correlation between participants' age and Learnability. Both Pearson and Spearman correlation coefficients (Aityan, 2022) indicated a non-significant relationship between age and Learnability.

Table 47
Correlation Matrix: Learnability and age

		Age	p-value
Learnability	Pearson's r	-0.086	0.298
	Spearman's rho	-0.082	0.319

Learnability and education level

This statistical test was intended to discover whether Learnability differs significantly among participants with different education levels. Prior to the analysis, assumptions of normality and homogeneity of variances were assessed (Chatzi and Doody, 2024). One-way ANOVA was conducted (Holmes and Rinaman, 2014) to compare Learnability across the four education levels.

Table 48
Normality Test (Shapiro-Wilk): Learnability and education

Statistic	p
0.875	<.001

Table 49
Homogeneity of Variances Test (Levene's): Learnability and education

F	df1	df2	p
0.503	3	146	0.681

Table 50
ANOVA: Learnability and education

	Sum of Squares	df	Mean Square	F	p
Education	2.42	3	0.807	0.804	0.494
Residuals	146.58	146	1.004		

Given the violation of normality assumption, a Kruskal-Wallis test was also performed as a non-parametric alternative (Chatzi and Doody, 2024). The test similarly revealed no statistically significant differences.

Table 51
Kruskal-Wallis: Learnability and education

	χ^2	df	p	ε^2
Learnability	2.20	3	0.531	0.0148

Learnability and work experience

This test aimed to examine whether there is a statistically significant relationship between participants' work experience and the Learnability of MLapi. Both Pearson's r and Spearman's rho were calculated (Aityan, 2022). Neither correlation was statistically significant, suggesting that professional experience does not appear to be associated with Learnability of MLapi.

Table 52
Correlation Matrix: Learnability and experience

		Experience	p-value
Learnability	Pearson's r	-0.099	0.226
	Spearman's rho	-0.093	0.257

4.4 Productivity Metrics and Efficiency Gains

To evaluate the practical impact of MLapi beyond usability, this section introduces key productivity metrics that reflect its contribution to operational efficiency. Traditional workflows involving Python scripting and manual data preprocessing can take several hours, especially for non-expert users. MLapi automates these steps through pre-configured templates and Excel integration, reducing task completion time by up to 70% in typical classification or regression scenarios. For instance, a task that previously required 3-4 hours of scripting and debugging (Weisz et al., 2023) can be completed in under 1 hour using MLapi's automated pipeline.

In addition to time savings, MLapi contributes to error reduction by minimizing manual coding and data handling. Errors in syntax, data formatting, and model configuration are common among novice users and can significantly affect the reliability of analytical outcomes. MLapi's structured JSON input and automated validation routines help eliminate such errors, ensuring consistent and reproducible results. This reduction in human error improves the quality of results but also enhances user confidence and trust in the system (Harper and Dorton, 2021). Moreover, the inclusion of visual diagnostics such as ROC curves, residual plots, and Feature Importance charts further supports accurate interpretation and decision-making.

From an organizational perspective, these productivity gains are translated into measurable cost savings and improved resource utilization. In a mid-sized analytics team, if each analyst saves 5 hours per week using MLapi, the cumulative annual time savings could exceed 2,500 hours. At an average, analysts hourly rate of $50, this equates to approximately $125,000 in productivity gains (Gartner, 2024a). These metrics underscore MLapi's value not only as an educational tool but also as a strategic asset for data-driven organizations seeking to optimize workflows, reduce overhead, and accelerate insight generation.

4.5 Pricing Elasticity Analysis

Understanding the willingness-to-pay (WTP) for MLapi is essential for evaluating its commercial viability and strategic positioning in the educational and enterprise analytics market. WTP reflects the highest price users are prepared to pay for a product based on perceived value, usability, and alternatives (Breidert, 2006). In the context of MLapi, its integration with Excel, automation of ML workflows, and educational value, significantly enhances its appeal to non-technical users. Preliminary feedback from pilot participants in the Greek banking sector indicated a high perceived value, with over 80% expressing interest in adopting MLapi if priced below €20/month for individual use or under €2,000/year for enterprise licensing. These figures align with pricing benchmarks for similar low-code analytics platforms (Gartner, 2024a).

Pricing elasticity, which measures how demand responds to changes in price, is another critical factor. MLapi's elasticity is expected to be moderate, given its dual role as both an educational and productivity-enhancing tool. For individual users, especially students and early-career analysts, price sensitivity may be higher due to budget constraints. However, for enterprise clients, the tool's ability to reduce labor costs and improve analytical throughput may justify higher pricing tiers. Elasticity modeling suggests that bundling MLapi with training modules or offering tiered pricing (e.g., freemium, professional, enterprise) could optimize adoption while maintaining profitability (Kotler and Keller, 2022). Additionally, offering volume discounts or site licenses for academic institutions could further expand its reach.

To validate these assumptions, future research should incorporate conjoint analysis or discrete choice experiments to quantify user preferences and simulate market scenarios. These methods can help identify optimal pricing strategies by balancing feature importance, perceived value, and budget constraints. Moreover, integrating feedback loops into MLapi's deployment (such as in-app surveys or usage analytics) can provide ongoing insights into user satisfaction and price tolerance. By aligning pricing with user expectations and market dynamics, MLapi can position itself competitively in the growing landscape of educational machine learning tools (Nagle and Müller, 2018).

4.6 Longitudinal Usage and Retention Analysis

Understanding how MLapi is used over extended periods is crucial for determining whether the tool continues to deliver value beyond initial interactions. While early usability indicators such as SUS scores capture immediate impressions, long-term behavioral data reveal whether users continue to rely on the system and integrate it into their routines. Monitoring usage across time makes it possible to observe trends such as changes in access frequency, evolving feature preferences, and points at which participation tapers off (Harper and Dorton, 2021). These forms of evidence are particularly relevant in both educational and professional settings, where sustained engagement is often associated with improved capability, enhanced output, and a stronger overall return on investment (Harper and Dorton, 2021).

Examining retention patterns also allows organizations to detect stages in the user lifecycle where interest begins to decline. For example, users may be active shortly after onboarding but gradually disengage if they encounter obstacles such as inadequate guidance, ambiguous documentation, or unmet expectations (Nagle and Müller, 2018). By gathering activity data across extended periods, organizations can identify these problematic moments and respond with targeted improvements, ranging from additional instructional content to interface enhancements or tailored feedback. Analytical approaches, including cohort comparisons and churn-based modeling, make it possible to classify users by behavioral tendencies and estimate the likelihood that they will continue to interact with the tool (Nagle and Müller, 2018). Insights derived from this analysis could play a central role in maintaining a positive user experience and ensuring MLapi's long-term relevance.

To support ongoing monitoring, MLapi should incorporate instrumentation capable of collecting anonymized records of key usage indicators, such as session count, time spent in the tool, and engagement with specific features. These metrics can be gathered either by lightweight tracking components within the Excel-VBA client or through logging routines implemented on the server side of the API. Complementing this data with user surveys or integrated feedback prompts provides additional context that cannot be captured through behavioral tracking alone. Over time, this combination of quantitative and qualitative information will establish a detailed picture of user engagement and guide continuous enhancement of MLapi's functionality, documentation, and support environment (Gartner, 2024b).

4.7 Conclusion

This research explores the variables that shape users' perceptions of MLapi's usability by applying inferential statistical procedures along with a Principal Component

Analysis (PCA). The hypothesis tests showed that SUS scores did not differ meaningfully across categories (gender, age, educational background, professional experience), suggesting that people from different groups interacted with the system in similar ways. The PCA revealed three underlying dimensions, Complexity, Agility, and Learnability, that together accounted for 90.3% of the variance in responses. These factors represent core aspects of how individuals engage with the tool, including how challenging it feels to use, how smoothly users can work with it, their confidence when interacting with the system, and how easily new users can understand it. Follow-up analyses indicated that these dimensions were interpreted similarly across demographic subgroups, demonstrating that the measurement structure was stable and not influenced by personal attributes. Overall, the findings highlight that MLapi delivers a generally uniform usability experience and establishes a solid basis for future refinement and assessment of the tool.

CHAPTER V:
DISCUSSION

5.1 Overview of Findings

The purpose of this research was to create MLapi, a machine learning API designed to make Python-based analytics accessible to individuals with limited programming exposure. The idea was driven by the growing need for tools that reduce technical barriers while supporting the development of data-related competencies, especially in academic and workplace environments where data literacy is increasingly essential. MLapi was conceived as both a computational backend and a learning aid, combining an Excel frontend with a PHP API layer and a Python execution environment so that users could apply advanced statistical and ML techniques without writing code.

To evaluate how users perceived the system, the study employed the System Usability Scale (Brooke, 1996). A total of 150 analytics professionals working in the Greek banking industry viewed a demonstration of MLapi and subsequently completed the SUS survey. The tool received an average score of 90.0, surpassing the commonly referenced usability threshold of 85. This outcome indicates that participants generally regarded the system as simple, efficient, and easy to use.

In addition to the aggregate SUS rating, a Principal Component Analysis was performed to uncover underlying patterns in users' responses. Three interpretable components emerged, Complexity, Agility, and Learnability, each representing different facets of system interaction. Complexity related to perceived difficulty and reliance on support; Agility represented confidence, smooth operation, and habitual use; and

Learnability captured the ease with which new users could become proficient. Collectively, these components explained more than 90% of the response variance, suggesting a stable and meaningful structure.

Further statistical testing showed that demographic characteristics, including gender, age, education, and professional background, did not significantly shape participants' perceptions. This result implies that the tool delivers a relatively uniform experience across varied user groups, supporting its suitability for broad deployment in learning and professional environments. The absence of demographic influence also points to the clarity and low cognitive burden of the system's design.

Overall, the study's results indicate that MLapi performs reliably, offers an inclusive user experience, and has the capacity to support learning through its modular architecture and transparent workflow. The evidence gathered highlights its potential as a valuable asset in data-science training and applied analytic contexts. Even so, the findings should be interpreted alongside the study's limitations and broader implications, which are discussed in later sections.

5.2 Technical Contributions and Educational Value

The creation of MLapi marks a meaningful step forward in the landscape of tools designed for data-science education, offering a novel way to connect sophisticated ML techniques with an interface that is easy for non-programmers to navigate. The central aim of MLapi is to broaden participation in machine learning by enabling users, particularly those without a coding background, to carry out advanced analytical procedures within a familiar platform: Microsoft Excel. Choosing Excel as the point of interaction was an intentional design strategy, using its widespread familiarity to reduce the learning curve associated with data analysis.

From a technical point of view, MLapi relies on a layered system consisting of an Excel-based client, a PHP/Apache middle tier, and a Python environment running through Anaconda. Each layer has a distinct role, allowing the system to remain flexible, maintainable, and easy to extend. Excel communicates with the API by sending JSON-formatted instructions written in VBA. The API interprets these instructions and relays them to the Python backend, where all ML routines are executed. The outcomes are returned in the form of Jupyter Notebook files, giving users both visual summaries and the executable code that generated them. This setup provides several benefits in engineering. It makes it possible to reuse the analytical core with different front-end applications, supports asynchronous operations, and allows upgrades to occur at the level of individual components without disrupting the whole system. The deployment of Jupyter Notebooks further strengthens transparency and traceability, essential qualities in academic and industry data-science practice. The system's algorithm library is extensive, incorporating both traditional models (e.g., logistic regression, decision trees) and modern techniques (e.g., XGBoost, neural networks), illustrating the breadth of analytical functionality available.

MLapi also serves as a structured educational tool. Users can access ready-made templates for a wide range of statistical and ML procedures, each enhanced with optional diagnostic analysis, visual outputs, and performance indicators. These templates help guide users toward established practices such as evaluating model performance, examining feature influence, and tuning hyperparameters. Options that typically require advanced coding proficiency (such as ROC curve generation, silhouette scoring, or running GridSearchCV) can be activated through simple interface selections.

A key educational feature of MLapi is its display of the Python code alongside the analytical results. This dual presentation transforms the tool from a black-box automation utility into a guided learning environment. Users can observe how the code is structured, how the data are handled, and how models are trained, offering a gradual introduction to Python and ML workflows while still enabling immediate practical use. In summary, MLapi architecture supports robust and scalable deployment, while its design philosophy emphasizes accessibility, clarity, and hands-on learning. By embedding exposure to ML concepts directly into the analytical process, MLapi contributes to expanding opportunities for data-science education.

5.3 Usability Evaluation and Principal Components

The usability evaluation of MLapi formed a central element of this research, offering measurable insights into how individuals experienced the tool during typical analytical tasks. The participants were data analytics practitioners with varying degrees of technical proficiency, and they interacted with MLapi in a context where ML and statistical operations were executed through a simplified front-end. To assess the overall user experience, the study employed the System Usability Scale (Brooke, 1996).

SUS is composed of ten questions, with scoring rules that adjust for each item's orientation before generating a final value between 0 and 100. In this evaluation, MLapi achieved an average score of 90, which exceeds the widely referenced benchmark of 85 typically associated with exceptional usability. Such a high rating indicates that users viewed the tool as clear, streamlined, and easy to operate, an important result given that the target audience may include individuals with limited exposure to programming or ML methods. To gain a deeper understanding (Jolliffe and Cadima, 2016) of how users responded to specific aspects of the system, the SUS data were further analyzed through a Principal Component Analysis (Jolliffe and Cadima, 2016). Three meaningful components emerged from this analysis (Complexity, Agility, and Learnability) and together they explained more than 90% of the variance in responses.

The first component, Complexity, encompassed perceptions related to the difficulty of performing tasks, inconsistencies in interaction, and the extent to which support might be needed. This factor reflects the mental effort required when using the system. Agility, the second component, captured elements such as user confidence, smooth workflow integration, and tendencies toward repeated use. The third component, Learnability, addressed how quickly new users could understand the tool and how intuitive the early learning phase felt, an especially relevant dimension in training environments where rapid familiarization is essential.

Reliability testing using Cronbach's alpha (Kilic, 2016) demonstrated strong internal consistency (Meyer, 2010) for all three factors, with coefficients ranging from 0.88 to 0.975. These results support the stability of the dimensional structure identified by the PCA. Further statistical analyses found no meaningful differences in component scores across demographic groups-gender, age, education, or professional experience, suggesting that MLapi delivers a comparable experience to a broad spectrum of users.

Overall, the usability findings highlight MLapi's effectiveness and accessibility. The high SUS rating, combined with the clear set of underlying usability dimensions, shows that the tool provides a positive and consistent user experience. The PCA results offer a deeper understanding of how people interact with MLapi, indicating strengths as well as areas that may be enhanced in future iterations. These outcomes will be valuable for guiding ongoing development and ensuring that MLapi continues to support both academic learners and professional users.

5.4 Inclusivity and Demographic Neutrality

One of the key outcomes of this research was the discovery that MLapi performs consistently well for users across a wide range of demographic backgrounds. Statistical examinations of both the overall SUS scores and the three extracted components, Complexity, Agility, and Learnability, showed no meaningful variation related to gender, age, educational attainment, or professional experience. This pattern indicates that people with different personal or occupational profiles were able to use MLapi with comparable ease. Achieving this level of neutrality is an important milestone for tools designed for learning and data analysis, as it demonstrates attention to equitable access and broad usability.

In many usability studies, demographic characteristics strongly influence how individuals interpret and navigate digital systems. Older participants may find new technologies more cognitively demanding, while those with less formal education may struggle with terminology or unfamiliar concepts. Against this fact, MLapi's lack of demographic disparities is particularly noteworthy, signaling that its design effectively accommodates users with varied abilities and backgrounds. This outcome carries several implications. It provides support for MLapi's guiding design principles, which prioritize clarity, simplicity, and structured interaction. By reducing reliance on coding knowledge and presenting workflows that are easy to follow, the tool lowers practical barriers for first-time or less technical users. Its integration with Microsoft Excel, a platform widely used across many industries, further enhances approachability by situating advanced ML capabilities inside an environment that feels familiar to most professionals.

The demographic neutrality observed in this study also strengthens MLapi's potential use in educational contexts. Instructors can introduce the tool in diverse classroom settings without worrying that certain groups of students will find it disproportionately difficult to use. Students from different academic disciplines or personal backgrounds can participate equally, helping to foster an inclusive learning atmosphere. Beyond formal education, these findings point to MLapi's suitability for workplace training and ongoing skill development. As organizations expand their data-literacy initiatives, there is growing need for tools that can serve employees with very different levels of experience. The ability of MLapi to provide consistent user experience across both novice and experienced professionals suggests it can function effectively in upskilling programs and professional development pathways.

5.5 Limitations and Future Considerations

Although the study demonstrates strong indicators of MLapi's usability and instructional value, several constraints should be recognized. To begin with, the SUS scale, while convenient and widely adopted, offers only a broad evaluation of user impressions and cannot pinpoint the particular features or interactions that might cause difficulty. More detailed investigations (such as observational studies, think-aloud sessions, or semi-structured interviews) would provide a clearer picture of the specific elements that shape user experience.

Another limitation arises from the composition of the participant group. All individuals were recruited from the Greek banking industry, resulting in a relatively uniform sample. Such homogeneity may have influenced the overall ratings and may not capture potential usability issues that could emerge among users from other professional or cultural backgrounds. Incorporating participants from multiple sectors or geographic regions in future research would enhance further the worth of the findings. Further consideration concerns the system's technical design. Although MLapi accommodates an extensive collection of ML techniques, its operation through Excel may restrict performance or flexibility in environments that demand more advanced workflows. Future development could involve connecting MLapi to additional frontend platforms or creating a dedicated graphical interface to support more complex or large-scale analytical use cases.

5.6 Strategic Roadmap for Commercialization

To guide MLapi's evolution from a research-oriented prototype into a market-ready solution, a clearly defined commercialization strategy is required. The initial stage should concentrate on assessing market relevance by running focused pilot implementations in fields such as finance, education, and healthcare, domains in which data-driven decision-making is important but technical barriers still hinder adoption.

Insights from these pilots would help shape the feature set, refine pricing structures, and inform the level of support services needed. Because MLapi leverages Excel and offers a low-code interaction model, it is particularly suited to professionals who work extensively with data but lack formal programming training, an audience that remains underserved by conventional ML tools (Harper and Dorton, 2021; Rao, 2025).

The next stage involves developing alliances and creating effective distribution pathways. Partnerships with universities could encourage integration of MLapi into academic programs, while collaborations with major enterprise software providers such as Microsoft or SAP could open possibilities for embedding MLapi within existing productivity ecosystems. A tiered pricing scheme, beginning with a no-cost basic version and extending to paid plans that offer expanded functionality, enterprise features, and priority support, may help balance accessibility with sustainable revenue generation. This approach is consistent with established principles of pricing strategy and market responsiveness (Kotler and Keller, 2022; Gartner, 2024a).

The final stage of the roadmap centers on scaling MLapi's infrastructure and building a broader user community. Transitioning the system to a cloud-hosted SaaS (Software as a service) architecture (Wu, C., 2018) would support multi-user workloads, streamline maintenance, and allow rapid deployment of updates. At the same time, encouraging an open-source ecosystem around MLapi could accelerate development cycles, reduce long-term costs, and foster community-driven enhancement. Comprehensive documentation, practical tutorials, and interactive discussion forums will play an essential role in supporting adoption and stimulating contributions. Through combined efforts in scalability, partnership development, and user-oriented pricing, MLapi has the potential to establish a significant presence in both educational and enterprise analytics environments (McKinsey & Company, 2025b; Nagle and Müller, 2018).

5.7 Revenue Models and Commercial Viability

To support long-term expansion and maintain competitive positioning, MLapi should adopt a revenue strategy that matches both its architecture and the needs of its various user groups. One pathway involves offering an enterprise licensing arrangement for organizations that require local installation because of strict data-handling or regulatory constraints. Such agreements (typically structured as annual or multi-year commitments) would grant organizations full access to MLapi's capabilities, enhanced support services, and opportunities for tailored feature development. This approach is especially relevant for fields like finance and healthcare, where on-premise deployments remain the norm due to privacy and compliance considerations (McKinsey & Company, 2025a; Gartner, 2024b).

In addition to licensing, MLapi can be delivered through a SaaS model aimed at individual practitioners, smaller businesses, and academic users. A cloud-based subscription framework would allow pricing differentiation, beginning with a no-cost entry tier for fundamental ML tasks and extending to advanced plans that include expanded analytics, collaborative tools, and API-level integration. SaaS offerings are well aligned with MLapi's modular backend, providing scalability, consistent revenue, simplified maintenance, and efficient feature roll-out (Kotler and Keller, 2022; Nagle and

Müller, 2018). Furthermore, operating in the cloud enables the collection of anonymized usage patterns that can guide product refinement and strengthen user-retention strategies.

Beyond software access, MLapi could establish additional income streams by delivering professional development resources and formal certification pathways. These options would fit students seeking to advance their skills in Python-based machine learning and Excel-integrated analytics. Webinars, instructor-led sessions, and asynchronous training modules could be packaged with subscriptions or purchased separately. Certification credentials would not only increase MLapi's market credibility but also appeal to employers and educational institutions looking to assess data-literacy competencies. This educational dimension reinforces MLapi's role as both a productivity enhancer and a learning platform (Harper and Dorton, 2021; Siddik, Li and Bezemer, 2025).

5.8 Organizational Change Management Framework

Successfully using MLapi within an organization involves far more than installing the software; it requires a coordinated effort to help people adjust their routines, gain confidence, and understand how the tool supports wider organizational objectives. Change management, defined as the structured process of guiding individuals and groups through transitions toward improved ways of working (Kotter, 2012), is therefore central to MLapi's introduction. In many cases, employees may be moving away from conventional business-intelligence tools or manual reporting practices, making the shift toward ML-supported workflows both unfamiliar and potentially disruptive (Bass et al., 2025).

A practical approach for supporting this transition is to draw on Kotter's 8-Step Change Model. The process begins by highlighting the necessity for improved data-driven practices and the limitations of existing analytic tools in meeting organizational demands (Kotter, 2012). Next, assembling a cross-departmental leadership group, bringing together representatives from IT, analytics teams, and business units, can help steer MLapi's introduction and maintain alignment with strategic priorities. Communicating an outline about how MLapi enhances productivity and decision quality is also critical. This message can be reinforced through internal demonstrations, pilot activities, and awareness campaigns that exemplify MLapi's usability and benefits (Kotter, 2012; Harper and Dorton, 2021).

Equipping users with the skills and support they need is another essential component of the change process. MLapi's built-in learning aids, including Jupyter Notebook outputs and Excel-based interactions, should be intentionally incorporated into training programs. Workshops, structured learning paths, and peer-support networks can help users become more comfortable with ML workflows. Brief, recurring feedback cycles, using surveys, analytics, or check-ins, can reveal obstacles and guide adjustments to the deployment strategy. Acknowledging individuals or teams who adopt MLapi early and demonstrate best practices can further strengthen engagement and momentum (Siddik, Li and Bezemer, 2025; McGehee, 2024a).

Long-term success also depends on formally integrating MLapi into the organization's operational ecosystem. This includes updating data-governance frameworks, aligning MLapi usage with key performance indicators, and ensuring leadership continues to advocate for its use. When organizations treat MLapi deployment as part of a broader digital-transformation strategy rather than a simple technology addition, they are more likely to cultivate data-driven habits and foster an innovative culture. Such an approach enhances both the return on investment and the potential for MLapi to scale across different teams and applications (Kotler and Keller, 2022; McKinsey & Company, 2025b).

CHAPTER VI:
SUMMARY, IMPLICATIONS, AND RECOMMENDATIONS

6.1 Summary

This study presented the conceptual design, technical development, and user evaluation of MLapi, a machine learning API created to lower the barriers associated with Python-based analytics. The project was driven by a growing need for analytical tools that simplify complex methods for individuals who may not have extensive programming experience. The technical contribution of the work centered on a multilayer system architecture in which Microsoft Excel operates as the user interface, a PHP API acts as the intermediary layer, and a Python environment running through Anaconda performs computational tasks. This separation of components enables flexible updates, straightforward maintenance, and an organized workflow. MLapi offers preconfigured templates for a broad range of statistical and ML techniques, and it generates outputs as Jupyter Notebook files to enhance transparency and support learning.

The usability evaluation relied on the System Usability Scale and included 150 analytics professionals employed in the Greek banking sector. Participants rated MLapi with an average score of 90, reflecting very positive usability perceptions. Further analysis of the SUS data through Principal Component Analysis identified three underlying dimensions, Complexity, Agility, and Learnability, each demonstrating strong internal coherence. Importantly, usability ratings did not differ meaningfully across demographic groups, indicating that MLapi can be used effectively by individuals with varied backgrounds and levels of experience.

6.2 Implications

The results of this study point to several considerations for future investigation:

Scalability and Technical Limitations

Linking MLapi to Microsoft Excel improves accessibility but also introduces notable constraints. Excel was never intended to handle extensive or highly complex datasets, meaning that efficiency issues are likely to arise as data volume increases. These constraints may restrict the system's ability to scale. Furthermore, depending on a server hosted locally or on a LAN may reduce its suitability for modern cloud-oriented or enterprise infrastructures, where centralized hosting and simultaneous multi-user operation are typically expected.

Usability Bias and Limited Applicability

Although the SUS evaluation suggested strong ease of use, the participant group consisted solely of banking professionals based in Greece. Such a specific sample may not represent other potential user categories, including students, researchers in unrelated disciplines, or users from different cultural or linguistic backgrounds. These contextual differences could meaningfully shape users' perceptions of usability. Without broader trials, the tool's current design may not translate effectively to diverse user communities or educational contexts.

Maintenance Demands and Long-Term Viability

MLapi relies on a combination of technologies (PHP, Apache, Python, and Excel). Each component must be kept up-to-date and compatible with the others, which may become more difficult as software libraries evolve, and operating systems change. In the absence of a stable support framework or an active open-source community, sustaining the system over time may pose considerable challenges, particularly within organizations or commercial settings.

6.3 Recommendations for Future Research

While this study provides a strong foundation, several options for future research are recommended:

- **Expand the Sample Population**: Future studies should include participants from other industries and geographic regions to enhance the generalizability of findings and explore potential cultural or sector-specific usability factors.
- **Incorporate Qualitative Methods**: Complementing SUS with interviews,or usability walkthroughs could uncover specific pain points and provide richer insights into user behavior and expectations.
- **Longitudinal Studies**: Investigating how users' perceptions and skills evolve over time with continued use of MLapi would offer valuable insights into its long-term educational contribution.
- **Comparative Usability Studies**: Comparing MLapi with other educational ML tools could highlight its relative strengths and weaknesses, guiding further refinement and positioning in the educational technology landscape.
- **Integration with Other Platforms**: Exploring the integration of MLapi with platforms beyond Excel, such as web-based dashboards or mobile applications, could broaden its applicability and user base.

6.4 Go-to-Market Strategy

To achieve effective market penetration and maintain long-term momentum, MLapi will benefit from a clearly articulated go-to-market (GTM) approach that aligns distribution choices with focused collaborations. A first step involves distinguishing the intended user base into three overarching clusters: academic organizations, small to medium-sized businesses, and large enterprises. Each of these groups brings different expectations and motivations. Educational institutions, for instance, typically value straightforward operation and compatibility with classroom workflows; SMEs often prioritize affordable analytics tools; and major corporations tend to focus on issues such as scaling capacity and regulatory compliance. Crafting segment-specific communication and onboarding processes can strengthen relevance and improve adoption outcomes (Kotler and Keller, 2022; McKinsey & Company, 2025a).

A combination of direct and partner-enabled routes can support distribution. Direct methods, such as MLapi's official website, live demonstrations, and digital presentations, are appropriate for individual users and smaller firms. Indirect pathways, including alliances with software distributors, cloud-platform vendors, and academic content providers, can broaden reach across institutional customers. Because the system works in conjunction with Microsoft Excel, MLapi can take advantage of cooperative marketing within Microsoft's broader ecosystem. Furthermore, availability on services such as AWS or Azure Marketplace could simplify cloud deployment and procurement processes (Gartner, 2024a; Flynn, 2024). Expanding MLapi's footprint on educational platforms like Coursera, Udemy, or edX may also increase visibility among instructors and learners.

Forming focused partnerships should remain a core pillar of the GTM framework. Working with universities may facilitate academic licensing and course-level integration, while collaborations with consulting and training organizations could drive adoption among professionals through bundled or guided offerings. Engagement with data-science communities and open-source contributors may further strengthen MLapi's credibility and support continued innovation. These relationships can be organized through cooperative agreements, affiliate structures, or co-labeled initiatives that align partner incentives and amplify MLapi's recognition within targeted markets (Nagle and Müller, 2018; Harper and Dorton, 2021).

A strong feedback structure is also essential. Monitoring user behavior, channel performance, and retention indicators can guide ongoing enhancement. Introducing MLapi through trial deployments and then expanding geographically or sector-by-sector would allow the team to adjust positioning and allocate resources more effectively. Through a coordinated blend of market segmentation, multi-route distribution, and purposeful partnerships, MLapi can build a solid market footprint and encourage uptake in both academic and enterprise environments (McGehee, 2024b; Siddik, Li and Bezemer, 2025).

6.5 Investment Requirements & Financial Projections

A thorough view of the upfront financial commitments and longer-term revenue expectations is vital for bringing MLapi to market. The current version of the system takes advantage of open-source components (Python, PHP) which helps keep early software expenses low. Nevertheless, moving beyond a prototype phase and shaping MLapi into a mature, widely deployable product will require resources dedicated to hosting infrastructure, specialized staff, promotional activities, and regulatory preparation. For a SaaS model delivered through cloud services, monthly hosting fees are estimated at roughly €50 - €150 per virtual machine, depending on demand and required scaling capabilities (Amazon Web Services, 2025). Ongoing operational expenditures for mid-range deployments, which cover updates, maintenance, and user support, are expected to total €5,000 - €15,000 per year (Gartner, 2024b).

Personnel costs constitute another major element of the commercialization budget. A small multidisciplinary team, typically including a backend engineer, a DevOps specialist, a user-experience designer, and first-line support staff, would require an estimated annual payroll of about €200,000, based on prevailing salary levels in Europe. Additional spending may be necessary for legal and compliance expertise, particularly for organizations in heavily regulated fields such as healthcare or finance. These services ensure adherence to GDPR, secure management of sensitive data, and implementation of robust authentication systems (McKinsey & Company, 2025a).

Projected revenue streams for MLapi appear favorable under a subscription-based SaaS framework. With a modest assumption of 1,000 professional subscribers during the first operational year at €19.99 per month, the platform could generate approximately €240,000 in annual recurring revenue. Enterprise agreements and academic site licenses could offer an extra €100,000 - €300,000 yearly, contingent upon the volume of institutional partnerships. Optional training and certification offerings, priced at €99 per enrolment, could yield an additional €50,000 - €100,000, especially if incorporated into bundled subscription packages (Kotler and Keller, 2022; Nagle and Müller, 2018).

Preliminary financial modelling indicates that MLapi could achieve break-even within roughly 18-24 months under moderate growth assumptions and careful expense management. Critical performance metrics, including customer acquisition cost (Breidert, 2006), lifetime value (Harper and Dorton, 2021), and user churn (Nagle and Müller, 2018), should be evaluated continuously to inform operational and strategic adjustments. Investments in automated marketing tools, guided onboarding, and customer success initiatives will play an important role in strengthening retention and supporting revenue expansion. With prudent startup spending and a diversified mix of income channels, MLapi has the potential to develop into a stable and scalable analytics offering (McGehee, 2024b; Siddik, Li and Bezemer, 2025).

6.6 Strategic Alliances & Market Expansion

Strategic partnerships will play a central role in expanding MLapi's presence in the market and strengthening its reputation across multiple domains. Collaboration with consulting firms offers considerable potential for enterprise-level adoption. Because these firms often advise organizations on digital transformation and integrate analytics into operational processes, they can introduce MLapi as part of broader modernization initiatives. Incorporating MLapi into their analytics offerings would enable clients to move beyond conventional BI systems and adopt ML-supported workflows, while MLapi gains sector-specific exposure and tailored implementations (McGehee, 2024b; McKinsey & Company, 2025a).

There is also substantial opportunity within the educational sector. Universities, technical training institutions, and online course providers increasingly seek tools that connect theoretical instruction with applied ML practice. MLapi's Excel compatibility and Jupyter Notebook output make it well suited for instructional environments that emphasize accessible, hands-on learning. Possible collaboration models include institutional licensing, embedding MLapi into coursework, and joint certification tracks. Such partnerships can both generate revenue and cultivate a pipeline of future professionals who develop familiarity with MLapi during their studies (Harper and Dorton, 2021; Siddik, Li and Bezemer, 2025).

To efficiently manage these partnerships, MLapi could introduce structured program categories, such as "Academic Partner", "Consulting Integrator", and "Training Affiliate", each associated with specific privileges and expectations. Academic partners might benefit from discounted licenses. Consulting integrators could obtain enterprise-grade deployment tools and shared promotional campaigns. Training affiliates could design and deliver courses built around MLapi, helping reinforce skills acquisition and drive usage through continuous learning (Kotler and Keller, 2022; Nagle and Müller, 2018).

These alliances should be supported by shared performance benchmarks and collaborative innovation efforts. Co-developed case studies, trial implementations, and research initiatives can showcase MLapi's versatility and real-world impact. Such activities not only provide evidence of effectiveness but also create structured channels for iterative feedback and enhancement (Gartner, 2024a; Flynn, 2024). Through coordinated partnerships with consulting organizations and educational institutions, MLapi can scale its reach, strengthen its market positioning, and build long-term trust among both professional and academic users.

6.7 Conclusion

This research illustrates that MLapi offers a dependable technical foundation, a strong educational utility, and a high level of user-friendliness for individuals working with machine learning and data analysis. Through the combination of an accessible interface and advanced computational features, MLapi responds to a notable gap in existing educational tools. The strong usability outcomes, reinforced by statistical evaluation and the tool's consistent performance, indicate that MLapi has significant potential to broaden participation in data-driven decision making. The study advances discussions in educational technology, usability assessment, and practical machine learning by presenting both conceptual contributions and tangible system innovations. As data-informed practices continue to influence a wide range of disciplines, platforms like MLapi will be increasingly important in helping students and professionals develop the competence and assurance needed to work with sophisticated analytical processes.

APPENDIX A

SYSTEM USABILITY SURVEY FORM

MLapi - System Usability Survey

Purpose:
Evaluate the usability of MLapi.

Estimated-Time:
10 - 15 minutes.

Confidentiality:
All responses are anonymous.

Contact:
akonomos@gmail.com

Procedure:
Please watch the demo of ML-API features, illustrating the key functionality for processing and analyzing data into Microsoft Excel, and afterwards complete the survey.

MLapi Demo - Video

Machine Learning API

Demographic Information

Gender *

- ○ Male
- ○ Female

Age *

Your answer

Educational Background *

- ○ High School
- ○ Bachelor
- ○ Masters
- ○ Doctorate

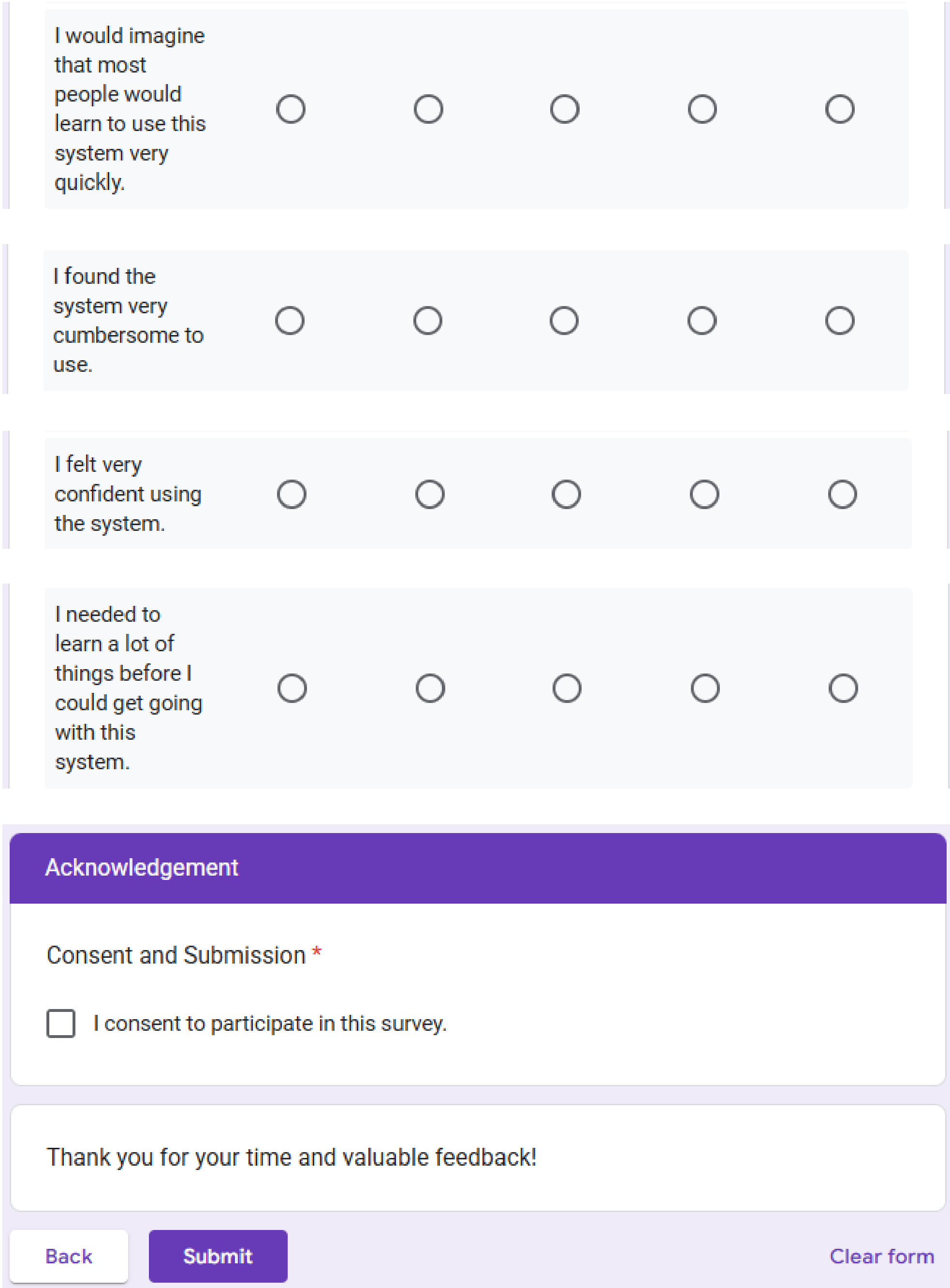
Professional Experience (in years) *

Your answer

System Usability Scale (SUS)

Rate the following statements on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree) *

	1	2	3	4	5
I think that I would like to use this system frequently.	○	○	○	○	○
I found the system unnecessarily complex.	○	○	○	○	○
I thought the system was easy to use.	○	○	○	○	○
I think that I would need the support of a technical person to use this system.	○	○	○	○	○
I found the various functions in this system were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in this system.	○	○	○	○	○



I would imagine that most people would learn to use this system very quickly.	○	○	○	○	○
I found the system very cumbersome to use.	○	○	○	○	○
I felt very confident using the system.	○	○	○	○	○
I needed to learn a lot of things before I could get going with this system.	○	○	○	○	○

Acknowledgement

Consent and Submission *

☐ I consent to participate in this survey.

Thank you for your time and valuable feedback!

Back Submit Clear form

APPENDIX B

MLAPI SOURCE CODE

MLapi python templates	Python.zip
PHP source code	PHP.zip
VBA source code	VBA.txt
Conda YAML file	MLapi.yml.txt

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. TensorFlow: a system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).

Abdoli, M., Akbari, M. and Shahrabi, J., 2023. Bagging supervised autoencoder classifier for credit scoring. *Expert Systems with Applications*, *213*, p.118991. https://doi.org/10.1016/j.eswa.2022.118991

Acito, F., 2023. Predictive analytics with KNIME. *Analytics for citizen data scientists. Switzerland: Springer*.

Agassi, A., Erel, H., Wald, I.Y. and Zuckerman, O., 2019, May. Scratch nodes ML: A playful system for children to create gesture recognition classifiers. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-6). https://doi.org/10.1145/3290607.3312894

AIHungry, 2025. Teachable Machine Pricing and Plans. Available at: https://aihungry.com/tools/teachable-machine/pricing

Aire, 2025. *Limitations of No-Code Automation Tools in Enterprise Systems*. Available at: https://aireapps.com/articles/limitations-of-no-code-automation-tools-in-enterprise-systems

Aityan, S.K., 2022. Linear Regression. In *Business Research Methodology: Research Process and Methods* (pp. 359-394). Cham: Springer International Publishing. https://doi.org/10.1007/978-1-4842-6797-4_14

Akinci, S., Kaynak, E., Atilgan, E. and Aksoy, Ş., 2007. Where does the logistic regression analysis stand in marketing literature? A comparison of the market positioning of prominent marketing journals. *European Journal of Marketing*, *41*(5/6), pp.537-567. https://doi.org/10.1108/03090560710737598

Albuquerque, P., Demo, G., Alfinito, S. and Rozzett, K., 2019. Bayesian factor analysis for mixed data on management studies. *RAUSP Management Journal*, *54*, pp.430-445. https://doi.org/10.1108/RAUSP-05-2019-0108

Alzahrani, N., Vahid, F., Edgcomb, A., Nguyen, K. and Lysecky, R., 2018, February. Python versus c++ an analysis of student struggle on small coding exercises in introductory programming courses. In *Proceedings of the 49th ACM Technical*

Symposium on Computer Science Education (pp. 86-91). https://doi.org/10.1145/3159450.3160586

Amazon Web Services, 2025. *AWS Pricing Calculator*. Available at: https://calculator.aws.amazon.com (Accessed: 11 October 2025).

Apple, 2025, *Create ML - machine learning*, *Apple Developer*. Available at: https://developer.apple.com/machine-learning/create-ml/ (Accessed: 12 April 2025).

Ashtari, S. and Eydgahi, A., 2017. Student perceptions of cloud applications effectiveness in higher education. *Journal of computational science*, *23*, pp.173-180. https://doi.org/10.1016/j.jocs.2016.12.007

Azmi, S.S. and Baliga, S., 2020. An overview of boosting decision tree algorithms utilizing AdaBoost and XGBoost boosting strategies. *Int. Res. J. Eng. Technol*, *7*(5), pp.6867-6870.

Bangor, A., Kortum, P.T. and Miller, J.T., 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, *24*(6), pp.574-594. https://doi.org/10.1080/10447310802205776

Barreto, H., 2015. Why excel?. *The Journal of Economic Education*, *46*(3), pp.300-309. https://doi.org/10.1080/00220485.2015.1029177

Bass, L., Lu, Q., Weber, I. and Zhu, L., 2025. *Engineering AI systems: architecture and DevOps essentials*. Addison-Wesley Professional.

BayTech Consulting, 2025. *Why Most Low-Code Platforms Eventually Face Limitations*. Available at: https://www.baytechconsulting.com/blog/why-most-low-code-platforms-eventually-face-limitations-and-strategic-considerations-for-the-future

Berthold, M., Cebron, N., Dill, F., Gabriel, T., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K. and Wiswedel, B., 2013. KNIME: The Konstanz Information Miner. Data Analysis. *Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization. Berlin, Heidelberg: Springer*. https://doi.org/10.1007/978-3-540-78246-9_38

Bhalerao, S. and Ingle, M., 2021. Convolutional Neural Network: A Systemic Review and Its Application using Keras. *Computing Technologies and Applications*, pp.199-217.

Biau, G. and Scornet, E., 2016. A random forest guided tour. *Test*, *25*(2), pp.197-227. https://doi.org/10.1007/s11749-016-0481-7

Bisong, E., 2019a. Google colaboratory. In *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners* (pp. 59-64). Berkeley, CA: Apress.

Bisong, E., 2019b. Introduction to Scikit-learn. In *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners* (pp. 215-229). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4470-8_18

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers* (pp. 177-186). Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-2604-3_16

Bottou, Z., Lai, C., Chen, X., Yang, B., Zhao, S. and Bai, X., 2015. Flood hazard risk assessment model based on random forest. *Journal of Hydrology*, *527*, pp.1130-1141. https://doi.org/10.1016/j.jhydrol.2015.06.008

Breidert, C., 2006. *Estimation of willingness-to-pay: Theory, measurement, application*. Wiesbaden: Duv.

Brooke, J., 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), pp.4-7.

Brooke, J., 2013. SUS: a retrospective. *Journal of usability studies*, *8*(2).

Brunton, S.L., Noack, B.R. and Koumoutsakos, P., 2020. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, *52*(1), pp.477-508. https://doi.org/10.1146/annurev-fluid-010719-060214

Business Research Insights, 2025. *Low-Code and No-Code Platform Market Size & Share [2024–2032]*. Available at: https://www.businessresearchinsights.com/market-reports/low-code-and-no-code-platform-market-118143

Byvatov, E. and Schneider, G., 2003. Support vector machine applications in bioinformatics. *Applied bioinformatics*, *2*(2), pp.67-77.

Cake.ai, 2025. AI Infrastructure Cost: A Complete 2025 Guide. Available at: https://www.cake.ai/blog/ai-infrastructure-costs

Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A. and Chen, A., 2020, April. Teachable machine: Approachable Web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems* (pp. 1-8). https://doi.org/10.1145/3334480.3382839

Celik, H. and Cinar, A., 2021. An application on ensemble learning using KNIME. In *2021 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 400-403). IEEE. https://doi.org/10.1109/ICDABI53623.2021.9655815

Center on Reinventing Public Education, 2025. *AI Early Adopter Districts: The Promises and Challenges of Using AI to Transform Education*. Available at: https://crpe.org/ai-early-adopter-districts-the-promises-and-challenges-of-using-ai-to-transform-education

Chandel, M., Silakari, S., Pandey, R. and Sharma, S., 2022 'A study on machine learning and Pythons framework', *International Journal of Computer Sciences and Engineering*, 10(5), pp. 58–64. https://doi.org/10.26438/ijcse/v10i5.5864

Chang, Y.H. and Zhang, Y.Y., 2022. Deep learning for clothing style recognition using YOLOv5. *Micromachines*, *13*(10), p.1678. https://doi.org/10.3390/mi13101678

Chatzi, A. and Doody, O., 2024. The one-way ANOVA test explained. *Nurse researcher*, *32*(4). https://doi.org/10.7748/nr.2023.e1885

Chen,Y., Chen,S., Yang,Y., Lu,S. (2024). *Comparison of decision tree and ensemble algorithms*. Applied and Computational Engineering,55,232-239. https://doi.org/10.54254/2755-2721/55/20241535

Chicho, B.T. and Sallow, A.B., 2021. A comprehensive survey of deep learning models based on Keras framework. *Journal of Soft Computing and Data Mining*, *2*(2), pp.49-62.

Chirodea, M.C., Novac, O.C., Novac, C.M., Bizon, N., Oproescu, M. and Gordan, C.E., 2021, July. Comparison of tensorflow and pytorch in convolutional neural network-based applications. In *2021 13th international conference on electronics, computers and artificial intelligence (ECAI)* (pp. 1-6). IEEE. https://doi.org/10.1109/ECAI52376.2021.9515098

Cho, E. and Kim, S., 2015. Cronbach's coefficient alpha: Well known but poorly understood. *Organizational research methods*, *18*(2), pp.207-230. https://doi.org/10.1177/109442811455599

Chowdhury, A., Vardhan, R.A., Koushik, K.A.S.I., Dongari, S. and Sriramaneni, H., 2022. Data mining using business intelligence and SQL. *International Research Journal of Engineering and Technology (IRJET)*, *9*.

Clarin, A., 2022. Comparison of the performance of several regression algorithms in predicting the quality of white wine in WEKA. *Int. J. Emerg. Technol. Adv. Eng.*, *12*(7), pp.20-26. https://doi.org/10.46338/ijetae0722_03

Cochran, J.J. and Sheehan, N.T., 2023. Spreadsheet Modeling and Wrangling with Python. *INFORMS Transactions on Education*. https://doi.org/10.1287/ited.2023.0047

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences [online]. 2nd ed: Routledge. https://doi.org/10.4324/9780203771587

Crouse, D.F., Willett, P., Pattipati, K. and Svensson, L., 2011, July. A look at Gaussian mixture reduction algorithms. In *14th International Conference on Information Fusion* (pp. 1-8). IEEE.

Dahouda, M.K. and Joe, I., 2021. A deep-learned embedding technique for categorical features encoding. *IEEE access*, *9*, pp.114381-114391. https://doi.org/10.1109/ACCESS.2021.3104357

Das, B.K., Jha, D.N., Sahu, S.K., Yadav, A.K., Raman, R.K. and Kartikeyan, M., 2022. Analysis of variance (ANOVA) and design of experiments. In *Concept Building in Fisheries Data Analysis* (pp. 119-136). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-4411-6_7

Dastres, R. and Soori, M., 2021. Artificial neural network systems. *International Journal of Imaging and Robotics (IJIR)*, *21*(2), pp.13-25.

Datarails, 2025. *AI and the Future of Microsoft Excel*. Available at: https://www.datarails.com/ai-and-the-future-of-excel

De Marchi, L. and Mitchell, L., 2019. *Hands-On Neural Networks: Learn how to build and train your first neural network model using Python*. Packt Publishing Ltd.

Demir, F., 2022. The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement. *Education and Information Technologies*, *27*(3), pp.4171-4194.

Dike, H.U., Zhou, Y., Deveerasetty, K.K. and Wu, Q., 2018, October. Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE. https://doi.org/10.1109/CBS.2018.8612259

Doleck, T., Lemay, D.J., Basnet, R.B. and Bazelais, P., 2020. Predictive analytics in education: a comparison of deep learning frameworks. *Education and Information Technologies*, *25*, pp.1951-1963. https://doi.org/10.1007/s10639-019-10068-4

Eckart, L., Eckart, S. and Enke, M., 2021. A brief comparative study of the potentialities and limitations of machine-learning algorithms and statistical techniques. In *E3S Web of Conferences* (Vol. 266, p. 02001). EDP Sciences. https://doi.org/10.1051/e3sconf/202126602001

Elliott, A.C., Hynan, L.S., Reisch, J.S. and Smith, J.P., 2006. Preparing data for analysis using Microsoft Excel. *Journal of investigative medicine*, *54*(6), pp.334-341. https://doi.org/10.2310/6650.2006.05038

ENISA, 2024. *Data Pseudonymisation Techniques and Best Practices*. European Union Agency for Cybersecurity. Available at: https://www.enisa.europa.eu/publications/data-pseudonymisation

Fergus, P. and Chalmers, C., 2022. Performance evaluation metrics. In *Applied Deep Learning: Tools, Techniques, and Implementation* (pp. 115-138). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-04420-5_5

Ferraris, S., Meo, R., Pinardi, S., Salis, M. and Sartor, G., 2023. Machine learning as a strategic tool for helping cocoa farmers in Côte D'Ivoire. *Sensors*, *23*(17), p.7632. https://doi.org/10.3390/s23177632

Ferreira, R., Canesche, M., Jamieson, P., Neto, O.P.V. and Nacif, J.A., 2024. Examples and tutorials on using Google Colab and Gradio to create online interactive student-learning modules. *Computer Applications in Engineering Education*, *32*(4), p.e22729. https://doi.org/10.1002/cae.22729

Fitriyani, S.R. and Murfi, H., 2016, May. The K-means with mini batch algorithm for topics detection on online news. In *2016 4th International Conference on*

Information and Communication Technology (ICoICT) (pp. 1-5). IEEE. https://doi.org/10.1109/ICoICT.2016.7571914

Flynn, S., 2024. *Tips for Leveraging Machine Learning in Excel for Businesses*. ReHack. Available at: https://rehack.com/ai/machine-learning-in-excel

Ford, N., Parsons, R., Kua, P. and Sadalage, P., 2022. *Building evolutionary architectures*. O'Reilly Media, Inc.

Fortune Business Insights, 2025. Big Data Technology Market Size, Share & Industry Analysis, 2024–2032. Available at: https://www.fortunebusinessinsights.com/industry-reports/big-data-technology-market-100144

Franke, T.M., Ho, T. and Christie, C.A., 2012. The chi-square test: Often used and more often misinterpreted. *American journal of evaluation*, *33*(3), pp.448-458. https://doi.org/10.1177/1098214011426594

Ganji, V.R. and Mannem, S.N.P., 2012. Credit card fraud detection using anti-k nearest neighbor algorithm. *International Journal on Computer Science and Engineering*, *4*(6), pp.1035-1039.

Gartner Peer Insights, 2025. *Top KNIME Competitors & Alternatives*. Available at: https://www.gartner.com/reviews/market/data-science-and-machine-learning-platforms/vendor/knime/alternatives

Gartner, 2024a. *AI Pricing Tips: Control Costs Effectively*. Available at: https://www.gartner.com/en/articles/ai-pricing-tips-control-costs-effectively

Gartner, 2024b. *IT Budget and Cost Optimization Benchmarks*. Available at: https://www.gartner.com

George, D. and Mallery, P., 2024. *IBM SPSS statistics 29 step by step: A simple guide and reference*. Routledge.

Geurts, P., Ernst, D. and Wehenkel, L., 2006. Extremely randomized trees. *Machine learning*, *63*(1), pp.3-42. https://doi.org/10.1007/s10994-006-6226-1

Ghosalkar, N.N. and Dhage, S.N., 2018, August. Real estate value prediction using linear regression. In *2018 fourth international conference on computing communication control and automation (ICCUBEA)* (pp. 1-5). IEEE.

Gil Urrutia, J.I., Brangier, E., Senderowicz, V. and Cessat, L., 2017, June. Beyond “usability and user experience”, towards an integrative heuristic inspection: from accessibility to persuasiveness in the UX evaluation: a case study on an insurance prospecting tablet application. In *International Conference on Applied Human Factors and Ergonomics* (pp. 460-470). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-60492-3_44

Glassdoor, 2025. *Salary Calculator – Know Your Worth*. Available at: https://www.glassdoor.com/Salaries/know-your- worth.htm

Goldberg, Y., 2017. *Neural network methods in natural language processing*. Morgan & Claypool Publishers.

Google Colab, 2025. Colab Paid Services Pricing. Available at: https://colab.research.google.com/signup/pricing

Grand View Research, 2025a. *Generative AI Market Size And Share | Industry Report, 2030*. Available at: https://www.grandviewresearch.com/industry-analysis/generative-ai-market-report

Grand View Research, 2025b. *AI in Education Market Size & Share Report, 2025–2030*. Available at: https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-education-market-report

Grattarola, D. and Alippi, C., 2021. Graph neural networks in tensorflow and keras with spektral [application notes]. *IEEE Computational Intelligence Magazine*, *16*(1), pp.99-106. https://doi.org/10.1109/MCI.2020.3039072

Hagendorff, T., 2020. The ethics of AI ethics: An evaluation of guidelines. *Minds and machines*, *30*(1), pp.99-120. https://doi.org/10.1007/s11023-020-09517-8

Halder, R.K., Uddin, M.N., Uddin, M.A., Aryal, S. and Khraisat, A., 2024. Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(1), p.113. https://doi.org/10.1186/s40537-024-00973-y

Halyal, S.V., 2019. Running Google Colaboratory as a server-transferring dynamic data in and out of colabs. *International Journal of Education and Management Engineering*, *9*(6), pp.35-39. https://doi.org/10.5815/ijeme.2019.06.04

Han, S. and Kwak, I.Y., 2023. Mastering data visualization with Python: practical tips for researchers. *Journal of Minimally Invasive Surgery*, *26*(4), p.167. https://doi.org/10.7602/jmis.2023.26.4.167

Hao, J. and Ho, T.K., 2019. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, *44*(3), pp.348-361. https://doi.org/10.3102/1076998619832248

Harper, S.B. and Dorton, S.L., 2021, September. A pilot study on extending the sus survey: early results. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 65, No. 1, pp. 447-451). Sage CA: Los Angeles, CA: SAGE Publications. https://doi.org/10.1177/1071181321651162

Harrison, M., 2019. *Machine learning pocket reference: working with structured data in python*. O'Reilly Media.

Healy, K., 2024. *Data visualization: a practical introduction*. Princeton University Press.

Heaton, J., 2020. Applications of deep neural networks with keras. *arXiv preprint arXiv:2009.05673*. https://doi.org/10.48550/arXiv.2009.05673

Hodeghatta, U.R., Nayak, U. (2023). Python Programming for Analytics. In: Practical Business Analytics Using R and Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-8754-5_18

Holmes, W. H., & Rinaman, W. C. (2014). *Analysis of Variance with Two Factors* (pp. 303–339). Springer, Cham. https://doi.org/10.1007/978-3-319-12550-3_12

Huang, C., Zhang, Z., Mao, B. and Yao, X., 2022. An overview of artificial intelligence ethics. *IEEE Transactions on Artificial Intelligence*, *4*(4), pp.799-819. https://doi.org/10.1109/TAI.2022.3194503

Huang, H.N., Chen, H.M., Lin, W.W., Huang, C.J., Chen, Y.C., Wang, Y.H. and Yang, C.T., 2023. Employing feature engineering strategies to improve the performance of machine learning algorithms on echocardiogram dataset. *Digital health*, *9*, p.20552076231207589. https://doi.org/10.1177/20552076231207589

Hulu, F.R.P., Raharjo, T. and Simanungkalit, T., 2024. The impact of usability in information technology projects. *Computer Science and Information Technologies*, *5*(1), pp.7-18. https://doi.org/10.11591/csit.v5i1.p7-12

Hunt, J., 2023. Pip and Conda Virtual Environments. In *Advanced Guide to Python 3 Programming* (pp. 643-658). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-40336-1_57

Ihrmark, D. and Tyrkko, J., 2023. Learning text analytics without coding? An introduction to KNIME. *Education for Information*, *39*(2), pp.121-137. https://doi.org/10.3233/EFI-230027

Ike, C.C., Ige, A.B., Oladosu, S.A., Adepoju, P.A., Amoo, O.O. and Afolabi, A.I., 2023. Advancing machine learning frameworks for customer retention and propensity modeling in ecommerce platforms. *GSC Adv Res Rev*, *14*(2), p.17. https://doi.org/10.30574/gscarr.2023.14.2.0017

Injadat, M., Moubayed, A., Nassif, A.B. and Shami, A., 2021. Machine learning towards intelligent systems: applications, challenges, and opportunities. *Artificial Intelligence Review*, *54*(5), pp.3299-3348. https://link.springer.com/article/10.1007/s10462-020-09948-w

ISO/IEC, 2022. *ISO/IEC 27001:2022 – Information Security, Cybersecurity and Privacy Protection*. International Organization for Standardization.

Jain, A., Somwanshi, D., Joshi, K. and Bhatt, S.S., 2022, April. A review: data mining classification techniques. In *2022 3rd International conference on intelligent engineering and management (ICIEM)* (pp. 636-642). IEEE. https://doi.org/10.1109/ICIEM54221.2022.9853036

Ji, J., Bai, T., Zhou, C., Ma, C. and Wang, Z., 2013. An improved k-prototypes clustering algorithm for mixed numeric and categorical data. *Neurocomputing*, *120*, pp.590-596. https://doi.org/10.1016/j.neucom.2013.04.011

Jia, Y., Garland, M. and Hart, J.C., 2011, December. Social network clustering and visualization using hierarchical edge bundles. In *Computer Graphics Forum* (Vol. 30, No. 8, pp. 2314-2327). Oxford, UK: Blackwell Publishing Ltd.

Jolliffe, I.T., 2002. *Principal Component Analysis.* 2nd ed. New York: Springer.

Jolliffe, I.T. and Cadima, J., 2016. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), p.20150202. https://doi.org/10.1098/rsta.2015.0202

Kalkha, H., Khiat, A., Bahnasse, A. and Ouajji, H., 2023. The rising trends of smart e-commerce logistics. *IEEE Access*, *11*, pp.33839-33857. http://dx.doi.org/10.1109/ACCESS.2023.3252566

Kaur, N.K., Kaur, U. and Singh, D., 2014. K-Medoid clustering algorithm-a review. *Int. J. Comput. Appl. Technol*, *1*(1), pp.42-45.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30. https://tinyurl.com/2h2dj56f

Kelana, B., Riskinanto, A., Nasyiah, M. and Fiandhika, M.F., 2024, September. Validation of System Usability Scale From Expert and Ordinary User Perspective. In *Proceeding of International Conference on Science, Health, And Technology* (pp. 12-18). https://doi.org/10.47701/icohetech.v5i1.4109

Khan, K., Rehman, S.U., Aziz, K., Fong, S. and Sarasvady, S., 2014, February. DBSCAN: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)* (pp. 232-238). IEEE. https://doi.org/10.1109/ICADIWT.2014.6814687

Khan, W., Daud, A., Khan, K., Muhammad, S. and Haq, R., 2023. Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Natural Language Processing Journal*, *4*, p.100026. https://doi.org/10.1016/j.nlp.2023.100026

Kilic, S., 2016. Cronbach's alpha reliability coefficient. *Psychiatry and Behavioral Sciences*, *6*(1), p.47. https://doi.org/10.5455/JMOOD.20160307122823

KNIME, 2025. *KNIME Open for innovation*. Available at: https://www.knime.com/ (Accessed: 13 April 2025).

Kotler, P. and Keller, K.L., 2022. *Marketing Management*. 16th ed. Pearson Education.

Kotter, J.P., 2012. *Leading Change*. Harvard Business Review Press.

Kumar, M., Rani, K.S. and Rao, C.R., 2017, January. Clustering voluminous of heterogeneous data. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICCCI.2017.8117706

Kusrini, K., 2015. Grouping of retail items by using K-means clustering. *Procedia Computer Science*, *72*, pp.495-502. https://doi.org/10.1016/j.procs.2015.12.131

Lakshmi, JVN, L., 2018. A Review on Impact of Deep Learning and TensorFlow in Medical Image Processing. *Grenze International Journal of Engineering & Technology (GIJET)*, *4*(3).

Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M. and Frank, E., 2019. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, *178*, pp.48-50. https://doi.org/10.1016/j.knosys.2019.04.013

Laurie, B. and Laurie, P., 2003. *Apache: The definitive guide*. " O'Reilly Media, Inc.".

Lee, S., 2025. TCO in EdTech: A Comprehensive Guide. NumberAnalytics. Available at: https://www.numberanalytics.com/blog/total-cost-of-ownership-edtech-guide

Lee, S.J. and Kwon, K., 2024. A systematic review of AI education in K-12 classrooms from 2018 to 2023: Topics, strategies, and learning outcomes. *Computers and Education: Artificial Intelligence*, *6*, p.100211. https://doi.org/10.1016/j.caeai.2024.100211

Lee, T.H., Ullah, A. and Wang, R., 2019. Bootstrap aggregating and random forest. In *Macroeconomic forecasting in the era of big data: Theory and practice* (pp. 389-429). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-31150-6_13

Lehmann, E.L. and Romano, J.P., 2005. *Testing statistical hypotheses*. New York, NY: Springer New York.

Lewis, J.R. and Sauro, J., 2018. Item benchmarks for the system usability scale. *Journal of Usability studies*, *13*(3). https://doi.org/10.5555/3294033.3294037

Lewis, J.R., 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, *34*(7), pp.577-590. https://doi.org/10.1080/10447318.2018.1455307

Li, Q., Fu, L., Zhang, W., Chen, X., Yu, J., Xia, W., Zhang, W., Tang, R. and Yu, Y., 2023. Adapting large language models for education: Foundational capabilities, potentials, and challenges. *arXiv preprint arXiv:2401.08664*. https://doi.org/10.48550/arXiv.2401.08664

Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S., 2020. Explainable ai: A review of machine learning interpretability methods. *Entropy*, *23*(1), p.18. https://doi.org/10.3390/e23010018

Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q. and Lee, I., 2018. Artificial intelligence in the 21st century. Ieee Access, 6, pp.34403-34421. https://doi.org/10.1109/ACCESS.2018.2819688

Llerena-Izquierdo, J., Mendez-Reyes, J., Ayala-Carabajo, R. and Andrade-Martinez, C., 2024. Innovations in Introductory Programming Education: The Role of AI with Google Colab and Gemini. *Education Sciences*, *14*(12), p.1330. https://doi.org/10.3390/educsci14121330

Ma, Q., Liu, Z., Zheng, Z., Huang, Z., Zhu, S., Yu, Z. and Kwok, J.T., 2024. A survey on time-series pre-trained models. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2024.3475809

Maalouf, M., 2011. Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies*, *3*(3), pp.281-299. https://doi.org/10.1504/IJDATS.2011.041335

MacVittie, L.A., 2006. *XAML in a Nutshell*. " O'Reilly Media, Inc.".

Madububambachu, U., Ukpebor, A. and Ihezue, U., 2024. Machine learning techniques to predict mental health diagnoses: A systematic literature review. *Clinical Practice and Epidemiology in Mental Health: CP & EMH*. https://doi.org/10.2174/0117450179315688240607052117

Mahkamov, A.A., Jumayev, T.S., Tuhtanazarov, D.S. and Dadamuxamedov, A.I., 2024. Using AdaBoost to improve the performance of simple classifiers. In *Artificial Intelligence, Blockchain, Computing and Security Volume 2* (pp. 755-760). CRC Press.

Maji, A.K., Gorenstein, L. and Lentner, G., 2020, November. Demystifying Python Package Installation with conda-env-mod. In *2020 IEEE/ACM International Workshop on HPC User Support Tools (HUST) and Workshop on Programming and Performance Visualization Tools (ProTools)* (pp. 27-37). IEEE. https://doi.org/10.1109/HUSTProtools51951.2020.00011

Mammone, A., Turchi, M. and Cristianini, N., 2009. Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(3), pp.283-289. https://doi.org/10.1002/wics.49

Mathew, L. and Bindu, V.R., 2020, March. A review of natural language processing techniques for sentiment analysis using pre-trained models. In *2020 Fourth international conference on computing methodologies and communication (ICCMC)* (pp. 340-345). IEEE. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00064

McClarren, R.G., 2021. Decision trees and random forests for regression and classification. In *Machine Learning for Engineers: Using data to solve problems for physical systems* (pp. 55-82). Cham: Springer International Publishing.

McGehee, J., 2024a. *Enterprise AI adoption: overcoming organizational resistance*. Gartner Research.

McGehee, N., 2024b. *Breaking Barriers: A Meta-Analysis of Educator Acceptance of AI Technology in Education*. Michigan Virtual Learning Research Institute. Available at: https://michiganvirtual.org/research/publications/breaking-barriers-a-meta-analysis-of-educator-acceptance-of-ai-technology-in-education

McKinsey & Company, 2025a. *AI-powered next best experience for customer retention*. Available at: https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/next-best-experience-how-ai-can-power-every-customer-interaction

McKinsey & Company, 2025b. *Evolving models and monetization strategies in the new AI SaaS era*. Available at: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/upgrading-software-business-models-to-thrive-in-the-ai-era

McNamara, N. and Kirakowski, J., 2006. Functionality, usability, and user experience: three areas of concern. *interactions*, *13*(6), pp.26-28. https://doi.org/10.1145/1167948.1167972

Meyer, P., 2010. *Understanding Measurement: Reliability.* New York: Oxford University Press.

Meyer, T.A., Ramirez, C., Tamasi, M.J. and Gormley, A.J., 2022. A user's guide to machine learning for polymeric biomaterials. *ACS Polymers Au*, *3*(2), pp.141-157. https://pubs.acs.org/doi/10.1021/acspolymersau.2c00037?goto=supporting-info

Mishra, P., Singh, U., Pandey, C.M., Mishra, P. and Pandey, G., 2019. Application of student's t-test, analysis of variance, and covariance. *Annals of cardiac anaesthesia*, *22*(4), pp.407-411. https://doi.org/10.4103/aca.ACA_94_19

Mitchell, L.J., 2016. *PHP Web Services: APIs for the Modern Web*. " O'Reilly Media, Inc.".

Moctezuma, L., Rivera, L.B., van Nouhuijs, F., Orcales, F., Kim, A., Campbell, R., Fuse, M. and Pennings, P.S., 2023. Using a decision tree to predict the number of COVID cases: a tutorial for beginners. *bioRxiv*, pp.2023-12. https://doi.org/10.1101/2023.12.19.572463

Mohammadi, R. and Farsijani, H., 2023. Optimization under Uncertainty: Machine Learning Approach. *International Journal of Innovation in Management, Economics and Social Sciences*, *3*(2), pp.23-32. https://doi.org/10.59615/ijimes.3.2.23

Molin, S., 2021. *Hands-On Data Analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization*. Packt Publishing Ltd.

Moor, J., 2006. The Dartmouth College artificial intelligence conference: The next fifty years. AI magazine, 27(4), pp.87-87. https://doi.org/10.1609/aimag.v27i4.1911

Muggleton, S., 2014. Alan Turing and the development of Artificial Intelligence. AI communications, 27(1), pp.3-10. https://doi.org/10.3233/AIC-130579

Murtagh, F. and Contreras, P., 2012. Algorithms for hierarchical clustering: an overview. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *2*(1), pp.86-97. https://doi.org/10.1002/widm.53

Musa, A.Z., 2017. Statistical issues II: chi-square test statistics. *Nigerian Journal of Clinical & Biomedical Research*, *7*(9), pp.29-35. http://njcbr.org/index.php/njcbr/article/view/23

Mustafy, T. and Rahman, M.T.U., 2024. Excel. In *Statistics and Data Analysis for Engineers and Scientists* (pp. 81-134). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-4661-7_3

Nagle, T.T. and Müller, G., 2018. *The Strategy and Tactics of Pricing: A Guide to Growing More Profitably*. 6th ed. Routledge.

Nagpal, A. and Gabrani, G., 2019, February. Python for data analytics, scientific and technical applications. In *2019 Amity international conference on artificial intelligence (AICAI)* (pp. 140-145). IEEE. https://doi.org/10.1109/AICAI.2019.8701341

Nasteski, V., 2017. An overview of the supervised machine learning methods. Horizons. b, 4(51-62), p.56.

Nelson, M.J. and Hoover, A.K., 2020, June. Notes on using Google Colaboratory in AI education. In *Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education* (pp. 533-534).

Nemeth, A. and Bekmukhambetova, A., 2023. Achieving Usability: Looking for Connections between User-Centred Design Practices and Resultant Usability Metrics in Agile Software Development. *Periodica Polytechnica Social and Management Sciences*, *31*(2), pp.135-143. https://doi.org/10.3311/ppso.20512

Newton, D., Yousefian, F. and Pasupathy, R., 2018. Stochastic gradient descent: Recent trends. *Recent advances in optimization and modeling of contemporary problems*, pp.193-220. https://doi.org/10.1287/educ.2018.0191

Nguyen, T., 2024. *Top 10 No Code Machine Learning Platforms to Use in 2024*. Neurond. Available at: https://www.neurond.com/blog/no-code-machine-learning-platforms

Nidhya, M.S. and Shah, P.K., 2023, November. Acquiring Knowledge by Performing Classification and Clustering of Datasets using WEKA: Intelligence through MLP, RF, DT, and RepTree. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)* (pp. 833-837). IEEE. https://doi.org/10.1109/ICCSAI59793.2023.10421077

NIST, 2023a. *Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5)*. National Institute of Standards and Technology.

NIST, 2023b. *Application Container Security Guide (SP 800-190)*. Available at: https://csrc.nist.gov/publications/detail/sp/800-190/final

Nita, S. L. (2016), Machine learning techniques used in big data, Scientific Bulletin of Naval Academy, Naval Academy Press, 19(1). http://dx.doi.org/10.21279/1454-864X-16-I1-078

Novac, O.C., Chirodea, M.C., Novac, C.M., Bizon, N., Oproescu, M., Stan, O.P. and Gordan, C.E., 2022. Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network. *Sensors*, *22*(22), p.8872. https://doi.org/10.3390/s22228872

O'Hagan, S. and Kell, D.B., 2015. Software review: the KNIME workflow environment and its applications in Genetic Programming and machine learning. Genetic Programming and Evolvable Machines, 16, pp.387-391. https://doi.org/10.1007/s10710-015-9247-3

OnlyNoCode, 2025. *What is Create ML? Features, Pricing & Alternatives*. Available at: https://www.onlynocode.com/create-ml/

Ono, S. and Goto, T., 2022. Introduction to supervised machine learning in clinical epidemiology. *Annals of Clinical Epidemiology*, *4*(3), pp.63-71. https://doi.org/10.37737/ace.22009

Open-Mind, 2020. *Cluster-11/open-mind: A web application that you can use to train a machine learning model and make it recognize your images*, *GitHub*. Available at: https://github.com/cluster-11/open-mind (Accessed: 12 April 2025).

Orji, F.A. and Vassileva, J., 2022. Machine learning approach for predicting students academic performance and study strategies based on their motivation. *arXiv preprint arXiv:2210.08186*. https://doi.org/10.48550/arXiv.2210.08186

OWASP, 2023. *Authentication Cheat Sheet*. Available at: https://cheatsheetseries.owasp.org

Panch, T., Szolovits, P. and Atun, R., 2018. Artificial intelligence, machine learning and health systems. Journal of global health, 8(2), p.020303. https://doi.org/10.7189/jogh.08.020303

Pang, B., Nijkamp, E. and Wu, Y.N., 2020. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, *45*(2), pp.227-248. https://doi.org/10.3102/1076998619872761

Park, H.S. and Jun, C.H., 2009. A simple and fast algorithm for K-medoids clustering. *Expert systems with applications*, *36*(2), pp.3336-3341. https://doi.org/10.1016/j.eswa.2008.01.039

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, pp.2825-2830.

Peng, K., Leung, V.C. and Huang, Q., 2018. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE access*, *6*, pp.11897-11906. https://doi.org/10.1109/ACCESS.2018.2810267

Perkel, J.M., 2018. By Jupyter, it all makes sense. *Nature*, *563*(7729), pp.145-146. https://doi.org/10.1038/d41586-018-07196-1

Pezoa, F., Reutter, J.L., Suarez, F., Ugarte, M. and Vrgoč, D., 2016, April. Foundations of JSON schema. In *Proceedings of the 25th international conference on World Wide Web* (pp. 263-273). https://doi.org/10.1145/2872427.2883029

Phillips, A.H. and Preece, T.E., International Business Machines Corp, 2009. *Redirecting file access through a HTTP web server*. U.S. Patent 7,502,856.

Podgorelec, V., Kokol, P., Stiglic, B. and Rozman, I., 2002. Decision trees: an overview and their use in medicine. *Journal of medical systems*, *26*, pp.445-463. https://doi.org/10.1023/A:1016409317640

Prasad, R. and Manjunath, T.C., 2024. AI-ML Trained Object Recognition System Development using Google Teachable Machine with the Help of Data Sciences. *Grenze International Journal of Engineering & Technology (GIJET)*, *10*.

Precedence Research, 2024. *AI in Education Market Size to Surpass USD 112.30 Bn by 2034*, *Precedenceresearch.com*. Available at: https://www.precedenceresearch.com/ai-in-education-market.

Precedence Research, 2025. *AI in Education Market Size to Surge USD 112.30 Bn by 2034*. Available at: https://www.precedenceresearch.com/ai-in-education-market

Querio, 2025. Low-Code/No-Code Data Analysis Tools for Non-Engineers. Available at: https://querio.ai/articles/low-code-no-code-data-analysis-tools-for-non-engineers

Quinn, K.M., 2004. Bayesian factor analysis for mixed ordinal and continuous responses. *Political Analysis*, *12*(4), pp.338-353. https://doi.org/10.1093/pan/mph022

Ranganathan, P., 2021. An Introduction to Statistics: Choosing the Correct Statistical Test. *Indian Journal of Critical Care Medicine*. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC8327789/

Rao, J., 2025. *Top Excel AI Features to Use in 2025*. ExcelTutorial.org. Available at: https://exceltutorial.org/top-excel-ai-features-to-use-in-2025

Raschka, S., Patterson, J. and Nolet, C., 2020. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, *11*(4), p.193. https://doi.org/10.3390/info11040193

Ratra, R., Gulia, P. and Gill, N.S., 2021, July. Performance Analysis of Classification Techniques in Data Mining using WEKA. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.

Reddy, B. and Fields, R., 2022, April. From past to present: a comprehensive technical review of rule-based expert systems from 1980-2021. In Proceedings of the 2022

ACM Southeast Conference (pp. 167-172). https://doi.org/10.1145/3476883.3520211

Reina-Parrado, M., Román-Graván, P. and Hervás-Gómez, C., 2025. *Integration of Artificial Intelligence and Machine Learning in Education: A Systematic Review*. International Journal of Educational Methodology, 11(2), pp.203–216. Available at: https://www.ijem.com/integration-of-artificial-intelligence-and-machine-learning-in-education-a-systematic-review

Research and Markets (2025) *AI in Education Market Review and Forecast to 2033*. Available at: https://finance.yahoo.com/news/ai-education-market-review-2024-155600872.html

Richardson, L. and Ruby, S., 2008. *RESTful web services*. " O'Reilly Media, Inc.".

Rincon-Patino, J., Ramirez-Gonzalez, G. and Corrales, J.C., 2018. Exploring machine learning: A bibliometric general approach using Citespace. *F1000Research*, *7*, p.1240. https://doi.org/10.12688/f1000research.15619.1

Rivas, P., 2020. *Deep Learning for Beginners: A beginner's guide to getting up and running with deep learning from scratch using Python*. Packt Publishing Ltd.

Rivers, K., Harpstead, E. and Koedinger, K. R. (2016) Learning Curve Analysis for Programming: Which Concepts do Students Struggle With?, *International Computing Education Research Workshop*, ACM, pp. 143–151. https://dblp.uni-trier.de/db/conf/icer/icer2016.html#RiversHK16

Rolon-Mérette, D., Ross, M., Rolon-Mérette, T. and Church, K., 2020. Introduction to Anaconda and Python: Installation and setup. *Quant. Methods Psychol*, *16*(5), pp.S3-S11. https://doi.org/10.20982/tqmp.16.5.S003

Routhier, E., Bin Kamruddin, A. and Mozziconacci, J., 2021. keras_dna: a wrapper for fast implementation of deep learning models in genomics. *Bioinformatics*, *37*(11), pp.1593-1594. https://doi.org/10.1093/bioinformatics/btaa929

Rovshenov, A. and Sarsar, F., 2023. Research trends in programming education: A systematic review of the articles published between 2012-2020. *Journal of Educational Technology and Online Learning*, *6*(1), pp.48-81. https://doi.org/10.31681/jetol.1201010

Saarela, M. and Jauhiainen, S., 2021. *Comparison of feature importance measures as explanations for classification models. SN Applied Sciences, 3, 272*. https://doi.org/10.1007/s42452-021-04148-9

Sagala, N.T. and Gunawan, A.A.S., 2022. Discovering the optimal number of crime cluster using elbow, silhouette, gap statistics, and nbclust methods. *ComTech: Computer, Mathematics and Engineering Applications*, *13*(1), pp.1-10. https://doi.org/10.21512/comtech.v13i1.7270

Salas-Rueda, R.A., Martínez-Ramírez, S.M., Ramírez-Ortega, J. and Alvarado-Zamorano, C., 2022. Students' Perception about the Use of an Educational Web Application during the COVID-19 Pandemic. *Journal of Learning for Development*, *9*(3), pp.509-527.

Salloum, S., Gaber, T., Vadera, S. and Shaalan, K., 2022. A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, *10*, pp.65703-65727. https://doi.org/10.1109/ACCESS.2022.3183083

Salmanpour, M.R., Alizadeh, M., Mousavi, G., Sadeghi, S., Amiri, S., Oveisi, M., Rahmim, A. and Hacihaliloglu, I., 2025. Machine Learning Evaluation Metric Discrepancies Across Programming Languages and Their Components in Medical Imaging Domains: Need for Standardization. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3549702

Saputra, D.M., Saputra, D. and Oswari, L.D., 2020, May. Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method. In *Sriwijaya international conference on information technology and its applications (SICONIAN 2019)* (pp. 341-346). Atlantis Press. https://doi.org/10.2991/aisr.k.200424.051

Sarker, I.H., 2021. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), p.160. https://doi.org/10.1007/s42979-021-00592-x

Sauro, J., 2011. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC.

Shi, X., Wong, Y.D., Li, M.Z.F., Palanisamy, C. and Chai, C., 2019. A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accident Analysis & Prevention*, *129*, pp.170-179. https://doi.org/10.1016/j.aap.2019.05.005

Siddik, M.S., Li, H. and Bezemer, C.P., 2025. A Systematic Literature Review of Software Engineering Research on Jupyter Notebook. *arXiv preprint arXiv:2504.16180*. https://doi.org/10.48550/arXiv.2504.16180

Simeone, O., 2018. A brief introduction to machine learning for engineers. *Foundations and Trends in Signal Processing*, *12*(3-4), pp.200-431. http://dx.doi.org/10.1561/2000000102

Sinaga, K.P. and Yang, M.S., 2020. Unsupervised K-means clustering algorithm. *IEEE access*, *8*, pp.80716-80727. https://doi.org/10.1109/ACCESS.2020.2988796

Singh, P., Manure, A., Singh, P. and Manure, A., 2020. Introduction to tensorflow 2.0. *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python*, pp.1-24. https://doi.org/10.1007/978-1-4842-5558-2_1

Skuza, A., 2025. *Unlocking AI's Business Value: The Impact of Low-Code/No-Code Platforms*. Available at: https://arekskuza.com/the-innovation-blog/unlocking-ais-business-value-the-impact-of-low-code-no-code-platforms

Smalheiser, N.R., 2017. Null hypothesis statistical testing and the t-test. *Data Literacy*, pp.127-136. https://doi.org/10.1016/B978-0-12-811306-6.00009-9

Sogemeier, D., Loew, A., Kulessa, S., Forster, Y., Naujoks, F. and Keinath, A., 2022. A global questionnaire? An international comparison of the system usability scale in the context of an infotainment system. *Usability and User Experience*, *39*(39). https://doi.org/10.54941/ahfe1001711

Song, Y., Kołcz, A. and Giles, C.L., 2009. Better Naive Bayes classification for high-precision spam detection. *Software: Practice and Experience*, *39*(11), pp.1003-1024. https://doi.org/10.1002/spe.925

Srinath, K.R., 2017. Python–the fastest growing programming language. *International Research Journal of Engineering and Technology*, *4*(12), pp.354-357.

Statista, 2025. Machine Learning - Worldwide | Statista Market Forecast. Available at: https://www.statista.com/outlook/tmo/artificial-intelligence/machine-learning/worldwide

Sukhdeve, S.R. and Sukhdeve, S.S., 2023. Data Processing and Transformation. In *Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services* (pp. 149-159). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-9688-2_5

Sundberg, L. and Holmström, J., 2024. Teaching Tip: Using No-Code AI to Teach Machine Learning in Higher Education. *Journal of Information Systems Education*, *35*(1), pp.56-66. https://doi.org/10.62273/CYPL2902

Thang, T.M. and Kim, J., 2011, April. The anomaly detection by using dbscan clustering with multiple parameters. In *2011 International Conference on Information Science and Applications* (pp. 1-5). IEEE. https://doi.org/10.1109/ICISA.2011.5772437

Tharwat, A., 2016. Principal component analysis-a tutorial. *International Journal of Applied Pattern Recognition*, *3*(3), pp.197-240. https://doi.org/10.1504/IJAPR.2016.079733

Thomas, J., 2024. Preprocessing. In *Applied Machine Learning Using mlr3 in R* (pp. 196-210). Chapman and Hall/CRC.

Turing, A., 1950. Computing Machinery and Intelligence, Mind, Volume LIX, Issue 236, October 1950, Pages 433–460, https://doi.org/10.1093/mind/LIX.236.433

Upasani, S.M., Bhoi, R.R., Puri, P.P., Jaybhaye, A.A., Bhaladhare, P. and Solanki, R.K., 2023. Revolutionizing Consumer Behavior: the Impact of E-Commerce Websites. *International Journal of Aquatic Science*, *14*(1), pp.516-528.

Vlachogianni, P. and Tselios, N., 2022. Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review. *Journal of Research on Technology in Education*, *54*(3), pp.392-409. https://doi.org/10.1080/15391523.2020.1867938

Wang, C. and Wang, L., 2023. Artificial neural network and its application in image recognition. *Journal of Engineering Research and Reports*, *24*(2), pp.50-57. https://doi.org/10.9734/jerr/2023/v24i2802

Weisz, J.D., Lucas, G., Tsai, J., and Muller, M., 2023. Non-Expert Programmers in the Generative AI Future. *Proceedings of the ACM on Human-Computer Interaction*. https://doi.org/10.1145/3663384.3663393

Wickramasinghe, I. and Kalutarage, H., 2021. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, *25*(3), pp.2277-2293. https://doi.org/10.1007/s00500-020-05297-6

Widiantoro, A.D., Wibowo, A. and Harnadi, B., 2021, November. User Sentiment Analysis in the Fintech OVO Review Based on the Lexicon Method. In *2021 Sixth International Conference on Informatics and Computing (ICIC)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICIC54025.2021.9632909

Wolf, Y.I., Rogozin, I.B., Grishin, N.V. and Koonin, E.V., 2002. Genome trees and the tree of life. *TRENDS in Genetics*, *18*(9), pp.472-479. https://doi.org/10.1016/S0168-9525(02)02744-0

Wu, C., 2018. Vanlearning: A Machine Learning SaaS Application for People Without Programming Backgrounds. *arXiv preprint*. https://doi.org/10.48550/arXiv.1804.01382

Xu, J. and Frydenberg, M., 2021. Python Programming in an IS Curriculum: Perceived Relevance and Outcomes. *Information Systems Education Journal*, *19*(4), pp.37-54.

Ye, C., Shen, Z., Wu, Y. and Loskot, P. (2024) Reconsidering Python Syntax to Enhance Programming Productivity, *International Journal for Research in Applied Science and Engineering Technology*.

Yudachev, S., Gordienko, N. A., Monakhov, P. A. and Sitnikov, S. (2022) Programming the Relley distribution, *Glavnyj mekhanik (Chief Mechanic)*, (7), pp. 471–479. https://doi.org/10.33920/pro-2-2207-06

Zeng, S., Huang, R., Kang, Z. and Sang, N., 2014. Image segmentation using spectral clustering of Gaussian mixture models. *Neurocomputing*, *144*, pp.346-356. https://doi.org/10.1016/j.neucom.2014.04.037

Zhai, X., Chu, X., Chai, C.S., Jong, M.S.Y., Istenic, A., Spector, M., Liu, J.B., Yuan, J. and Li, Y., 2021. A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity*, *2021*(1), p.8812542. https://doi.org/10.1155/2021/8812542

Zhang, Y., 2022. A historical interaction between artificial intelligence and philosophy. *arXiv preprint arXiv:2208.04148*. https://doi.org/10.48550/arXiv.2208.04148

Zheng, A. and Casari, A., 2018. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".

Zhou, D.W., Sun, H.L., Ning, J., Ye, H.J. and Zhan, D.C., 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*. https://doi.org/10.48550/arXiv.2401.16386

Zhou, L., Pan, S., Wang, J. and Vasilakos, A.V., 2017. Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, pp.350-361. https://doi.org/10.1016/j.neucom.2017.01.026

Zhou, S., Wu, G., Dong, Y., Ni, Y., Hao, Y., Jiang, Y., Zhou, C. and Tao, Z., 2023. Evaluations on supervised learning methods in the calibration of seven-hole pressure probes. *PLoS One*, *18*(1), p.e0277672. https://doi.org/10.1371/journal.pone.0277672

Zhu, J., Su, Y., Liu, Z., Liu, B., Sun, Y., Gao, W. and Fu, Y., 2022. Real-time biomechanical modelling of the liver using LightGBM model. *The International Journal of Medical Robotics and Computer Assisted Surgery*, *18*(6), p.e2433. https://doi.org/10.1002/rcs.2433

Zou, J., Han, Y. and So, S.S., 2009. Overview of artificial neural networks. Artificial neural networks: methods and applications, pp.14-22. https://doi.org/10.1007/978-1-60327-101-1_2