“LOWERING TECHNICAL BARRIERS IN MACHINE LEARNING EDUCATION: A USABILITY EVALUATION OF AN INTEGRATED ANALYTICS FRAMEWORK”

Research Paper

Antonios Konomos, SSBM Geneva, Switzerland, akonomos@gmail.com

Georgios Konomos, Athens University of Economics and Business, Greece, gkonomos@yahoo.gr

“Abstract”

The increasing role of data analytics and machine learning (ML) across different fields has created a strong need for educational tools that support both theoretical understanding and hands-on practice. Although Python is widely used as the main programming language for data analytics, many students and early-stage researchers find it difficult to use effectively because of steep learning curves in programming, statistics, and ML frameworks. This study introduces MLapi, an API-based machine learning tool designed to reduce technical barriers and make ML methods more accessible in educational and applied analytics settings. MLapi uses a three-tier architecture that links Microsoft Excel as the user interface with a Python-based analytics engine, allowing users to run statistical analysis and ML algorithms while viewing the underlying Python code in Jupyter Notebook format. To assess usability, an empirical study was conducted using the System Usability Scale (SUS) with data analytics professionals. The evaluation included reliability testing, hypothesis testing across demographic groups, and Principal Component Analysis to identify usability dimensions. Results show high overall usability, strong internal consistency, and no significant differences between demographic groups, indicating that MLapi provides an inclusive and user-friendly learning environment. These findings suggest that MLapi can serve as both an educational tool and a practical analytics solution, helping to create accessible ML learning environments and encouraging broader participation in data analytics education.

Keywords: Machine learning, Data Analytics, System Usability Scale, API-based Architecture, HumanComputer Interaction, Principal Component Analysis

1. Introduction

The growing use of data-driven research and decision-making in both academic and professional fields has made machine learning (ML) a core skill rather than a niche expertise. Modern developments in artificial intelligence (AI) and ML are based on long-standing theoretical studies of human reasoning and logic, which trace back to classical thinkers like Aristotle (Zhang, 2022). However, these ideas only became practical with the rise of digital computing in the mid-20th century. Key contributions by Turing (1950) and the formal launch of AI as a research discipline at the Dartmouth Conference in 1956 marked a major milestone in the development of intelligent systems (Moor, 2006).

Early AI systems mainly used symbolic reasoning and clearly defined rules to handle very specific tasks. These methods worked well in controlled settings but could not adapt or generalize beyond the conditions they were designed for (Reddy and Fields, 2022). Because of these limitations, researchers developed machine learning techniques that allow systems to learn patterns directly from data (Mohammadi and Farsijani, 2023; Panch, Szolovits and Atun, 2018). Statistical learning and neural networks became popular in the late 20th century, and later, deep learning grew rapidly thanks to more computing power and larger datasets (Liu et al., 2018; Zou, Han and So, 2009; Brunton, Noack and Koumoutsakos, 2019; Injadat et al., 2021). Today, machine learning and deep learning are widely used in predictive analytics, helping organizations make data-driven decisions, improve efficiency, and cut costs by extracting valuable insights from their data (Lindéus and Shetty, 2023; Balaji and Silic, 2022). These advances have turned ML into a key technology for applications like predictive analytics, pattern recognition, and automated decision-making across many industries.

In contemporary data analytics, machine learning techniques are commonly categorized into supervised and unsupervised approaches. Supervised methods, including classification and regression, rely on labeled data to generate predictive models, whereas unsupervised techniques such as clustering and dimensionality reduction uncover latent structures within unlabeled datasets (Dike et al., 2018; Nasteski, 2017). The widespread adoption of these methods has been facilitated by the Python programming language, which has emerged as the dominant platform for data analytics due to its readable syntax and extensive ecosystem of open-source libraries (Nagpal and Gabrani, 2019; Srinath, 2017). Libraries such as scikit-learn, pandas, and matplotlib have become standard tools in both academic research and industry practice (Hao and Ho, 2019; Molin, 2021).

Despite these advances, the effective use of Python-based ML tools remains challenging for many learners and novice researchers. Applying machine learning methods typically requires not only programming proficiency but also a solid understanding of statistics, data preprocessing, and model evaluation. These requirements can significantly hinder engagement with data-driven methodologies, particularly among individuals without prior technical backgrounds (Alzahrani et al., 2018; Sundberg and Holmström, 2023). As a result, a skills gap persists between the availability of powerful analytical tools and the ability of learners to use them confidently and productively (Wu, 2018). This gap is especially evident in educational settings, where learners may struggle to transition from conceptual understanding to practical implementation.

To address these challenges, numerous educational platforms and software tools have been developed to support the teaching of machine learning concepts. While many of these tools offer user-friendly interfaces and lower the barrier to entry, they often abstract away the underlying programming logic, limiting opportunities for learners to develop transferable Python skills. Consequently, learners may gain conceptual familiarity with ML processes without acquiring the coding competence necessary for advanced or independent analytical work.

In response to this limitation, the present study introduces **MLapi**, an API-based machine learning tool designed to support both data analytics practice and education. MLapi enables users to execute statistical and ML methods through a familiar spreadsheet-based interface while simultaneously exposing the underlying Python source code in a Jupyter Notebook structure. This dual design aims to reduce technical complexity while promoting transparency, reproducibility, and progressive skill development. By integrating usability-focused system design with established analytical workflows, MLapi seeks to bridge the gap between accessibility and technical depth in machine learning education.

The study evaluates the usability of MLapi using the System Usability Scale (SUS) and examines whether demographic factors influence user perceptions of the system. Through quantitative analysis and dimensionality reduction techniques, the research contributes empirical evidence on the design of inclusive and user-centered ML educational tools. In doing so, it aligns with ongoing efforts to enhance accessibility and effectiveness in data analytics and machine learning education.

2. Related Work and Theoretical Background

2.1 Machine learning in practice

Machine learning has become a core part of data analytics, helping to identify patterns, make predictions, and support decision-making from increasingly large and complex datasets. In business contexts, successful adoption of analytics depends not only on technology but also on organizational, cultural, and process factors. Critical success factors include combining technical, business, and soft skills, standardizing data practices, and using advanced analytical tools (Rao and Provodnikova, 2021). In education, ML plays a dual role, as a subject of study and as a practical tool for analyzing data.

Python has become the leading language for teaching and applying machine learning (Nagpal and Gabrani, 2019; Hodeghatta and Nayak, 2023). Although Python is more accessible than many other programming languages, using ML libraries effectively still requires technical skills that many learners lack in the early stages of their education (Ye et al., 2024). This gap between understanding concepts and applying them in practice has led to the development of specialized educational tools and platforms.

2.2 Educational software tools for machine learning

A wide range of software tools have been developed to support the teaching and learning of machine learning concepts, each emphasizing different learning priorities such as accessibility, visualization, or computational power. Cloud-based platforms like Google Colab have become particularly popular in educational settings due to their ease of access and integration with Jupyter Notebooks. Colab allows learners to execute Python code and experiment with ML models without local installation, fostering engagement and rapid experimentation (Bisong, 2019; Nelson and Hoover, 2020; Ferreira et al., 2024). However, reliance on internet connectivity and external infrastructure can introduce accessibility constraints, especially for learners with limited resources (Llerena-Izquierdo et al., 2024).

Frameworks such as TensorFlow and Keras provide powerful environments for developing and deploying ML and DL models. TensorFlow's scalability and extensive ecosystem make it suitable for industrial applications, while Keras offers a high-level API that simplifies model prototyping and experimentation (Abadi et al., 2016; Heaton, 2020). Although these frameworks are widely used in research and practice, studies indicate that their complexity often limits their effective integration into introductory educational curricula, particularly where institutional support and curriculum modernization are lacking (Demir, 2022; Rovshenov and Sarsar, 2023).

Other tools adopt low-code or no-code approaches to reduce the technical demands placed on learners. Teachable Machine, Create ML and Open Mind represent this category by enabling users to build ML models through graphical interfaces without writing code (Carney et al., 2020; Apple, 2025). These tools are effective for introducing ML concepts and fostering early engagement, yet they abstract away the programming logic underlying model construction. As a result, learners may gain conceptual familiarity while remaining unable to transfer knowledge to Python-based analytical environments.

Traditional data mining platforms such as WEKA and KNIME occupy an intermediate position between accessibility and analytical depth. WEKA provides a comprehensive suite of algorithms and visualization tools through a graphical interface, supporting experimentation and research workflows (Pedregosa et al., 2011; Jain et al., 2022). KNIME extends this approach with visual workflows and optional integration with Python and R, allowing users to combine drag-and-drop analytics with scripting when desired (Berthold et al., 2013; O'Hagan and Kell, 2015). While these platforms are effective for demonstrating analytical pipelines, they do not consistently expose or generate Python code in a manner that supports systematic programming skill development.

2.3 Research gap

Although existing educational tools for machine learning successfully lower initial barriers to entry, a persistent gap remains between usability and skill transfer. Many platforms prioritize ease of use by concealing implementation details, thereby limiting learners' opportunities to understand how ML algorithms are executed in Python. Conversely, professional ML frameworks expose full programming complexity but often overwhelm novice users. This dichotomy creates a challenge for educators and learners seeking tools that are simultaneously accessible, transparent, and educationally effective.

The literature suggests a need for educational ML systems that support gradual skill development by combining intuitive interfaces with explicit exposure to underlying code and analytical workflows. More specifically, tools that integrate familiar environments, such as spreadsheets commonly used in data analysis, with Python-based ML execution may offer a promising pathway for bridging this gap (Konomos, 2026). Addressing this need requires not only technical innovation but also empirical evaluation of usability and inclusivity.

In response, the present study proposes MLapi as an API-based machine learning tool designed to balance accessibility and technical transparency. The following sections describe the system design and present an empirical evaluation of its usability within a professional data analytics context.

3. MLapi: System Design and Architecture

3.1 Design and objectives

MLapi was designed to function simultaneously as a data analytics tool and an educational resource for machine learning. Its design is guided by four core objectives: accessibility, transparency, reproducibility, and educational value. Accessibility is addressed by allowing users to initiate machine learning analysis from a familiar spreadsheet-based environment, reducing the need for advanced programming expertise. Transparency is achieved by exposing the underlying Python source code alongside analytical results, enabling learners to observe and progressively understand ML implementations. Reproducibility is supported through standardized analytical workflows and notebook-based reporting, while educational value is embedded through structured templates that align statistical reasoning with machine learning practice.

Rather than replacing existing ML frameworks, MLapi acts as an intermediary layer that orchestrates established Python libraries and presents their functionality in a form suitable for educational and applied analytics contexts. This approach enables learners to focus on analytical reasoning and interpretation while gradually developing technical proficiency.

3.2 System architecture

MLapi adopts a three-tier architecture, a widely used design paradigm for scalable and maintainable systems (Ford et al., 2022; Bass et al., 2025). The architecture separates presentation, communication, and processing concerns into distinct layers, facilitating modularity and extensibility.

The C**lient tier** serves as the presentation layer and is responsible for user interaction. In this study, Microsoft Excel is used as the primary client application due to its widespread adoption in data analytics practice. Excel allows users to prepare datasets, configure analytical requests, and initiate machine learning tasks without requiring direct interaction with programming environments.

The **API tier** provides a RESTful interface that receives analytical requests from the client. Implemented using PHP and hosted on an Apache web server, this tier is responsible for request validation, data serialization, and communication with the backend processing environment. Data are transmitted in JSON format, enabling structured and platform-independent exchange of analytical parameters and datasets.

The **Processing tier** hosts the core analytical logic and executes all statistical and machine learning computations. This tier is implemented in Python using the Anaconda distribution, which provides a stable and reproducible environment for scientific computing and machine learning. Communication between the API tier and the processing tier is facilitated through a secure file-based interface within a shared network environment.

This architectural separation ensures that MLapi remains flexible and adaptable to different client interfaces and computational environments, while maintaining consistency in analytical execution.

3.3 Analytical engine and supported methods

The processing tier of MLapi integrates a comprehensive set of statistical and machine learning methods commonly used in data analytics education and practice. These include descriptive statistics, inferential tests (such as t-tests and ANOVA), and reliability measures, as well as machine learning algorithms for classification, regression, clustering, and dimensionality reduction. The selection of methods reflects both educational relevance and practical applicability.

MLapi leverages established Python libraries such as scikit-learn, pandas, matplotlib, and related packages to ensure methodological robustness and alignment with industry-standard practices. Analytical execution is organized through predefined templates that encapsulate complete workflows, including data preprocessing, model training, evaluation, and visualization. Users may select optional post-hoc analysis and diagnostic outputs, allowing for flexible yet structured experimentation.

3.4 Data preprocessing and model evaluation

Data preprocessing is a critical component of reliable machine learning analysis and is handled within MLapi through automated and configurable pipelines. These pipelines include handling of missing values, encoding of categorical variables, feature scaling, and class imbalance adjustment. One-hot encoding is applied to categorical features to ensure compatibility with numerical ML algorithms, while class imbalance can be addressed through up-sampling or down-sampling strategies depending on the analytical context (Dahouda and Joe, 2021; Harrison, 2019).

Model evaluation procedures are tailored to the type of analysis performed. For classification tasks, MLapi reports standard performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve. Regression analysis include error-based metrics and coefficients of determination, while clustering and dimensionality reduction methods are accompanied by internal validation measures such as silhouette scores, Elbow method diagnostics, and explained variance ratios. These evaluation outputs are presented alongside visual representations, supporting intuitive interpretation and comparative analysis.

To enhance robustness and generalizability, MLapi incorporates k-fold cross-validation and automated hyperparameter optimization using GridSearchCV. This functionality reduces overfitting risks and familiarizes users with best practices in model tuning and validation.

3.5 Educational output and learning support

A distinctive feature of MLapi is its use of Jupyter Notebook–formatted reports as the primary output medium. Each analytical request generates a response containing the executed Python code, textual explanations, numerical results, and visualizations. This format supports transparency and reproducibility while serving as an instructional artifact that learners can review, modify, and reuse.

By presenting code and results together, MLapi encourages a learning-by-observation approach, allowing users to gradually transition from interface-driven interaction to direct engagement with Python-based ML workflows. This design supports incremental skill acquisition and aligns with constructivist learning principles, where understanding is built through active exploration and reflection.

4. Research Methodology

4.1 Research design

The present study adopted a quantitative research design to evaluate the usability of MLapi as a machine learning tool for data analytics and educational use. The focus of the evaluation was on perceived usability rather than learning outcomes, as usability is a critical prerequisite for the effective adoption of educational technologies.

A demonstration-based evaluation approach was employed. Participants were introduced to MLapi through a standardized video presentation that illustrated the system architecture, analytical workflow, supported statistical and machine learning methods, and output format. This approach ensured consistent exposure to system functionality across participants and minimized variability arising from differences in prior programming experience or individual usage strategies.

4.2 Sample and participant characteristics

The study population consisted of professionals working in data analytics–related roles within the Greek banking sector. This population was selected due to its frequent engagement with data analysis tasks and its increasing exposure to machine learning methods in professional practice. Participants represented a heterogeneous group in terms of gender, age, educational background, and professional experience, allowing for the examination of usability perceptions across diverse user profiles.

The demographic variables collected included gender, age, highest level of education attained, and years of professional experience. These variables were incorporated into the analysis to test predefined hypotheses regarding their potential influence on perceived usability.

4.3 Sample size determination and power analysis

To estimate the sample size, a pilot assessment using the SUS was conducted with 30 participants. This preliminary evaluation produced 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).

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 goal of the analysis was to determine whether the MLapi's usability score would significantly exceed this threshold. To support a noted statistical comparison, Confidence level of 95% ($\alpha = 0.05$) and Statistical power of 80% ($\beta = 0.20$) were applied.

Using these parameters, the required sample size 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 ttest, 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 Type II error, that is, failing to detect an actual effect (Cohen, 1988).

4.4 System usability scale

Usability was assessed using the System Usability Scale, a standardized instrument widely used for evaluating perceived usability across different types of systems (Brooke, 1996; Brooke, 2013). The SUS consists of ten items rated on a five-point Likert scale, with alternating positively and negatively worded statements.

SUS scores were calculated according to established scoring procedures, resulting in a composite score ranging from 0 to 100. Higher scores indicate better perceived usability. The SUS was selected due to its simplicity, robustness, and extensive use in usability research, including evaluations of information and educational systems (Bangor et al., 2008).

4.5 Data collection procedure

Data collection was conducted electronically. Participants were first exposed to a video demonstration of MLapi, which illustrated the complete analytical workflow, including data submission via Excel, execution of statistical and machine learning algorithms through the API, and presentation of results. Following the demonstration, participants completed the SUS questionnaire along with a brief demographic survey. Participation was voluntary and anonymous, and no personally identifiable information was collected.

4.6 Data analysis

Data analysis was performed using quantitative statistical methods. Descriptive statistics were computed to summarize SUS scores and participant demographics. The internal consistency of the SUS instrument was assessed using Cronbach's alpha, with values above the commonly accepted threshold indicating satisfactory reliability.

Inferential statistical analysis was conducted to test the four predefined hypotheses. Independent samples *t*-tests were used to examine gender-based differences in usability perceptions. Correlation analysis was employed to assess relationships between SUS scores and continuous variables such as age and professional experience. Differences across educational attainment levels were examined using one-way analysis of variance (ANOVA). Statistical significance was evaluated at the 0.05 level.

In addition to hypothesis testing, Principal Component Analysis (PCA) was conducted to explore the underlying structure of usability perceptions and to identify latent dimensions within the SUS responses.

4.7 Methodological limitations

While the selected methodology enabled a robust evaluation of perceived usability, certain limitations should be acknowledged. The demonstration-based evaluation did not capture extended hands-on interaction or longitudinal usage patterns. Furthermore, the professional context of the sample may limit generalizability to purely academic or novice learner populations. These limitations are addressed in the discussion and future research sections.

5. Results

5.1 Descriptive statistics and reliability analysis

The study sample covered 150 individuals, with 58.0% identifying as male and 42.0% as female. Regarding educational level, most participants held a bachelor's degree (44.7%), followed by master's degree holders (34.7%); 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 analysis.

The internal consistency of the SUS instrument was assessed using Cronbach's alpha. The analysis yielded a coefficient of $\alpha = 0.84$, indicating strong reliability and confirming that the SUS items measured a coherent usability construct within the context of MLapi.

5.2 Hypothesis testing

Four hypotheses were formulated to examine whether demographic variables influenced perceived usability of MLapi.

Hypothesis 1 (H1): Gender Differences

There is a statistically significant difference in SUS scores between male and female participants.

An independent samples t-test was conducted to compare SUS scores across gender groups. 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. The result approached significance but did not reach the conventional threshold: $U = 2275$, $p = 0.071$. Thus, H1 was not supported, suggesting that gender did not influence perceived usability.

Hypothesis 2 (H2): Age

There is a statistically significant relationship between participant age and SUS score.

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. Consequently, H2 was not supported, indicating that usability perceptions were not associated with age.

Hypothesis 3 (H3): Educational Level

There are statistically significant differences in SUS scores across educational attainment levels.

A one-way analysis of variance (ANOVA) was conducted to compare SUS scores among participants with undergraduate, postgraduate, and doctoral-level education. The results showed no statistically significant differences between groups ($F(3, 146) = 1.54$, $p = 0.207$). Given the violation of normality assumption, a Kruskal-Wallis test was also performed as a non-parametric alternative. The test similarly revealed no statistically significant differences in SUS scores across education levels: $\chi^2(3) = 4.46$, $p = 0.216$. Therefore, H3 was not supported, suggesting that educational background did not affect usability evaluations.

Hypothesis 4 (H4): Professional Experience

There is a statistically significant relationship between years of professional experience and SUS score.

Both Pearson's r ($r = 0.013$, $p = 0.871$) and Spearman's rho ($rho < 0.001$, $p = 0.997$) were calculated to assess the relationship between SUS scores and years of professional experience. Neither correlation was statistically significant ($p > .05$). As a result, H4 was not supported, indicating that usability perceptions were consistent regardless of professional experience level.

5.3 Principal component analysis

To explore the underlying structure of usability perceptions, a Principal Component Analysis (PCA) was conducted on the ten SUS items. The PCA yielded three components, collectively explaining 90.3**%** of the total variance. Varimax rotation was applied to enhance interpretability.

- **Component 1: Complexity** (explained variance: 45.5%)
  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** (explained variance: 27.5%)
  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** (explained variance: 17.4%)
  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.

Reliability analysis of the extracted components demonstrated excellent internal consistency, with high Cronbach's alpha values (Complexity: 0.942, Agility: 0.975, Learnability: 0.880), supporting the stability and interpretability of the identified usability dimensions.

5.4 Summary of results

The statistical analysis indicates that MLapi achieved high perceived usability, strong reliability, and consistent evaluations across demographic groups. None of the four statistical hypotheses were supported, demonstrating demographic neutrality in usability perception. The PCA revealed three underlying dimensions, Complexity, Agility, and Learnability, that together accounted for 90.3% of the variance in responses. Follow-up analysis 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.

6. Discussion

The objective of this study was to evaluate the usability of MLapi and to examine whether demographic factors influenced users' perceptions of the system. The findings provide empirical support for MLapi as a usable and inclusive machine learning tool for data analytics and educational settings. In this section, the results are interpreted in relation to the research objectives, the identified research gap, and existing approaches to machine learning education.

6.1 Interpretation of usability findings

The overall SUS score indicates that MLapi achieved a level of usability classified as good to excellent. This result suggests that the system successfully balances functionality with ease of use, despite supporting complex statistical and machine learning methods. High perceived usability is particularly important in educational contexts, where excessive technical complexity can discourage engagement and hinder learning progression.

The strong internal consistency of the SUS responses further supports the reliability of the usability evaluation. Together, these findings indicate that MLapi provides stable and coherent user experience, aligning with its design objectives of accessibility and transparency.

6.2 Demographic neutrality and inclusivity

None of the four hypotheses examining demographic differences in usability perceptions were supported. Specifically, perceived usability did not differ significantly across gender, age, educational level, or professional experience. This demographic neutrality is a notable finding, as it suggests that MLapi can be effectively used by individuals with diverse backgrounds and levels of expertise.

In the context of machine learning education, such neutrality is particularly valuable. Educational technologies that disproportionately favor users with advanced technical backgrounds risk reinforcing existing skill gaps. The absence of demographic effects in this study indicates that MLapi lowers entry barriers and supports equitable access to machine learning tools, aligning with broader goals of inclusivity in data analytics education.

6.3 Usability dimensions and system design

The Principal Component Analysis revealed three underlying dimensions of usability perception: complexity, agility, and learnability. These dimensions correspond closely with the core design principles of MLapi. Perceived complexity reflects users' sensitivity to unnecessary cognitive effort and technical friction. The fact that complexity emerged as a distinct dimension suggests that users were attentive to how effectively MLapi reduced operational burden while executing sophisticated analytical tasks. System agility captures perceptions related to smooth integration, responsiveness, and workflow coherence, which are critical for both professional analytics and educational experimentation. Learnability, the third dimension, is particularly relevant for educational applications, as it reflects users' confidence in their ability to quickly become proficient with the system. The emergence of these dimensions supports the notion that usability in ML tools is multidimensional and extends beyond simple ease of use.

6.4 Positioning MLapi within education tools

Existing ML educational tools often adopt one of two contrasting approaches. Low-code or no-code platforms emphasize accessibility but hide implementation details, limiting opportunities for learners to develop transferable programming skills. Conversely, professional ML frameworks expose full technical complexity but can overwhelm novice users.

MLapi occupies an intermediate position between these extremes. By enabling users to initiate analysis through a familiar spreadsheet interface while simultaneously exposing the underlying Python code and results in a Jupyter Notebook format, MLapi supports progressive skill development. Users can engage with machine learning concepts at an accessible level while gradually developing an understanding of Python-based workflows.

The high usability scores and demographic neutrality observed in this study suggest that this hybrid design approach is effective. MLapi demonstrates that it is possible to support both ease of use and technical transparency, addressing a key gap identified in the literature and in existing educational practice.

6.5 Summary

Overall, the discussion of results indicates that MLapi achieves its primary design objectives and addresses a critical gap in ML education tools. The combination of high usability, demographic neutrality, and multidimensional usability supports MLapi's potential as both an educational technology and a practical data analytics tool. These findings provide a foundation for considering broader implications, limitations, and future research directions, which are discussed in the following section.

7. Implications, Limitations, and Future Research

7.1 Implications for educational practice

The findings of this study suggest that MLapi has the potential to support data analytics and machine learning education by reducing technical barriers while preserving analytical transparency. The high usability scores and the absence of demographic differences indicate that MLapi can be adopted by learners and professionals with diverse backgrounds, levels of experience, and educational attainment. This inclusivity is particularly important in educational settings, where learners often enter with heterogeneous skill profiles.

By integrating a familiar spreadsheet-based interface with Python-based machine learning execution, MLapi enables learners to engage with analytical workflows without requiring immediate mastery of programming syntax. At the same time, the exposure of underlying Python code through Jupyter Notebook outputs supports gradual skill development and promotes reproducibility. This dual design allows educators to introduce machine learning concepts in a progressive manner, aligning conceptual understanding with practical implementation.

7.2 Implications for educational technology design

From a system design perspective, the results underscore the importance of balancing usability and technical depth in educational machine learning tools. The identification of distinct usability dimensions (perceived complexity, system agility, and learnability) highlights the multifaceted nature of user experience in analytical systems. Educational technologies that address only surface-level ease of use may fail to support deeper learning, while technical systems risk alienating novice users.

MLapi demonstrates that architectural choices, such as the separation of client interaction from analytical processing and the use of standardized output formats, can enhance both usability and educational value. These design principles may inform the development of future learning tools that aim to support data analytics and machine learning outcomes without sacrificing accessibility.

7.3 Implications for professional data analytics practice

Beyond educational contexts, the findings suggest that MLapi may also be useful in professional environments where analysts rely heavily on spreadsheet tools but require access to advanced machine learning capabilities. The ability to execute robust analytical methods while maintaining transparency and reproducibility can support exploratory analysis, model validation, and collaborative work. The system's usability across varying levels of professional experience indicates its potential applicability in mixedskill teams.

7.4 Limitations of the study

Several limitations should be considered when interpreting the results of this study. First, the evaluation focused exclusively on perceived usability rather than on learning outcomes or performance measures. While usability is a critical prerequisite for adoption, it does not directly capture the extent to which MLapi enhances conceptual understanding or skill acquisition.

Second, the study employed a demonstration-based evaluation rather than prolonged hands-on usage. Although this approach ensured consistent system exposure, it may not fully reflect usability perceptions that emerge through extended interaction or real-world application.

Third, the sample consisted of professionals from a specific sector and geographical context, which may limit the generalizability of the findings to academic or novice learner populations.

7.5 Directions for future research

Future research could extend the present study in several ways. Longitudinal studies involving hands-on use of MLapi in classroom settings could examine its impact on learning outcomes, skill development, and learner confidence. Comparative studies could evaluate MLapi against other machine learning educational tools to assess relative effectiveness and learning value.

Further research may also explore adaptations of MLapi for different educational levels or domains, as well as enhancements to its functionality, such as expanded algorithm support or adaptive learning features. Investigating how learners transition from interface-driven interaction to independent Python coding when using MLapi would provide valuable insight into its role in supporting progressive learning pathways.

8. Conclusion

This study introduced MLapi, an API-based machine learning tool designed to support data analytics education by balancing accessibility, usability, and technical transparency. Addressing a persistent challenge in machine learning education, MLapi seeks to reduce entry barriers while enabling users to engage with authentic Python-based analytical workflows. The system's architecture integrates a familiar spreadsheet interface with a Python-driven analytics engine, producing transparent and reproducible outputs in a Jupyter Notebook format.

An empirical usability evaluation using the System Usability Scale demonstrated that MLapi achieved high perceived usability, strong internal reliability, and consistent evaluations across demographic groups. The absence of statistically significant differences related to gender, age, educational level, and professional experience indicates that MLapi supports inclusive use across diverse user profiles. Exploratory analysis further revealed that usability perceptions were structured around key dimensions related to complexity, agility, and learnability, reflecting the system's design objectives.

The findings contribute to the field of educational technology by providing empirical evidence that machine learning tools can be both user-friendly and technically transparent. MLapi illustrates how architectural and interface design choices can support progressive skill development without oversimplifying analytical processes. As such, the system offers a viable approach for integrating machine learning into data analytics education and practice.

While the study focused on usability evaluation, the results establish a foundation for future research examining learning outcomes, long-term engagement, and comparative effectiveness. Overall, MLapi represents a step toward more accessible and inclusive machine learning education, supporting broader participation in data-driven analysis and decision-making.

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’, *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16).* pp.265-283.

Alzahrani, N., Vahid, F., Edgcomb, A., Nguyen, K. and Lysecky, R. (2018) ‘Python versus C++ an analysis of student struggle on small coding exercises in introductory programming courses’, *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.* pp. 86-91, https://doi.org/10.1145/3159450.3160586

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

Balaji, S. N. and Silic, M. (2022) ‘Using AI And Machine Learning Efficiently to Decide on Voyage Fixture of Tanker Ships to Increase Turnarounds and Profitability’, *Global Journal of Business and Integral Security*, 5(1). Available at: https://gbis.ch/index.php/gbis/article/view/63

Bangor, A., Kortum, P. and Miller, J. (2008) ‘An empirical evaluation of the System Usability Scale’, *International Journal of Human–Computer Interaction, 24*(6), 574–594.

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

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’, *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/9783-540-78246-9_38

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

Brooke, J. (2013) ‘SUS: A retrospective’, *Journal of Usability Studies.* 8(2), pp.29–40.

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

Brunton, S. L., Noack, B. R. and Koumoutsakos, P. 2019 ‘Machine learning for fluid mechanics’, *Annual Review of Fluid Mechanics.* 52, 477–508.

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

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

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

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). ‘Unsupervised learning-based anomaly detection in data analytics’, *IEEE Access, 6*, pp.35360–35372.

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), https://doi.org/10.1002/cae.22729

Ford, N., Parsons, R., Kua, P. and Sadalage, P. (2022) ‘Building Evolutionary Architectures’, *O'Reilly Media, Inc*.

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

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

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/10.1007/978-1-4842-87545_18

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

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

Konomos, A. (2026). ‘Mind the gap: A review of machine learning tools in educational settings’, *Global journal of Business and Integral Security*. 9(1), https://www.gbis.ch/index.php/gbis/article/view/984

Lehmann, E.L. and Romano, J.P. (2005) ‘Testing statistical hypotheses’, *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

Lindéus, G. and Shetty, S. (2023) ‘Exploring the Efficiency and Accuracy of AI-Powered Predictive Analytics: A Six-Country Case Study of the Logistic Performance Index’, *Global journal of Business and Integral Security*. 6(3). https://www.gbis.ch/index.php/gbis/article/view/268

Liu, Y., Chen, P. and Ding, Y. (2018). ‘Deep learning applications in education’, *IEEE Access*, pp.60674–60683.

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

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. H. (2006). ‘The Dartmouth College artificial intelligence conference’, *AI Magazine*. 27(4), 87–91.

Nagpal, R. and Gabrani, G. (2019). ‘Python for data analytics’, *International Journal of Computer Sciences and Engineering*. 7(3), 78–83.

Nasteski, V. (2017) ‘An overview of the supervised machine learning methods’, *Horizons*.

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

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

Panch, T., Szolovits, P. and Atun, R. (2018) ‘Artificial intelligence, machine learning and health systems’, *Journal of global health*. 8(2), https://doi.org/10.7189/jogh.08.020303

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’, *Journal of machine Learning research*. 12, pp.2825-2830.

Rao, D.D. and Provodnikova, A. (2021) ‘Analyzing the role of business analytics adoption on effective entrepreneurship, *Global journal of Business and Integral Security*. 4(6). https://gbis.ch/index.php/gbis/article/view/35

Reddy, B. and Fields, R. (2022) ‘From past to present: a comprehensive technical review of rule-based expert systems from 1980-2021’. *Proceedings of the 2022 ACM Southeast Conference*. pp.167-172, https://doi.org/10.1145/3476883.3520211

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

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

Sundberg, J. and Holmström, J. (2023). 'Barriers to machine learning adoption among novice analysts', *Information Systems Frontiers*. 25(1), 215–229.

Turing, A. M. (1950). 'Computing machinery and intelligence', *Mind.* 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

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

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*. 12(3), pp.776-785, https://doi.org/10.22214/ijraset.2024.58903

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

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