East-West J. of Mathematics: Vol. 11, No 2 (2025) pp. 142-158

A NO-CODE SOLUTION FOR TEACHING STATISTICAL HYPOTHESIS TESTING FOR STUDENTS IN THE SOCIAL SCIENCES AND ECONOMICS

Nguyen Trang Thao[1,2], **Le Thi Nhan**[3]

[1] *Laboratory for Artificial Intelligence*
Institute for Computational Science and Artificial Intelligence
Van Lang University - Ho Chi Minh City, Viet Nam.
E-mail: thao.nguyentrang@vlu.edu.vn

[2] *Faculty of Information Technology*
School of Technology, Van Lang University - Ho Chi Minh City, Viet Nam.
E-mail: thao.nguyentrang@vlu.edu.vn

[3] *Faculty of Information Technology*
Nguyen Tat Thanh University - Ho Chi Minh City, Viet Nam.
E-mail: lthinhan@ntt.edu.vn

Abstract

Teaching statistical hypothesis testing to students in the social sciences and economics presents many challenges, as students in these fields often lack programming skills, while advanced platforms such as Altair AI Studio (RapidMiner) do not natively support common statistical testing tools like t-tests, proportion tests, and ANOVA analysis. Our solution is to build testing processes in Altair AI Studio (RapidMiner) that do not require users to program. These processes primarily rely on using macro settings to provide parameters and leverage internal Python execution capabilities to handle the back-end. The developed processes include one-sample t-tests, independent two-sample t-tests, paired two-sample t-tests, one- and two-proportion tests, and ANOVA with post-hoc analysis. Students only need to set input values via macros, eliminating the need

Key words: RapidMiner, hypothesis testing, t-test, proportion test, ANOVA, no-code platform, Python macro integration.

for programming and allowing them to focus on understanding core statistical concepts. A dataset, namely, Amazon service reviews, was used to demonstrate the flexibility and applicability of the proposed solution.

1 Introduction

RapidMiner is well known as a powerful and flexible platform for machine learning and artificial intelligence that allows users to build advanced analytical models without needing programming skills. Its simple drag-and-drop interface makes it easy for people from many fields—such as social sciences and economics—to design and run complex data analysis workflows visually and efficiently [7]. With more than a thousand built-in machine learning algorithms and comprehensive data-preprocessing methods, RapidMiner supports the entire data science pipeline—from data collection and data cleaning to model training, model evaluation, and result visualization—in one unified environment. This no-code framework lowers technical barriers in education, helping students focus on key concepts in statistics and machine learning—such as hypothesis testing, regression analysis, and clustering—without worrying about programming syntax. For example, common inferential statistical techniques such as t-tests and ANOVA are easy to use through built-in operators that guide users in formulating hypotheses, checking assumptions, and testing statistical significance, which leads to a more active and application-oriented learning experience. In addition, RapidMiner's emphasis on reproducible workflows and visual process design helps students better understand statistical reasoning and model evaluation by encouraging exploration through clear, interactive, and transparent visual models [2]. Its widespread use in academic research and industry demonstrates its effectiveness both as a pedagogical tool and as a practical platform for real-world data science and machine learning tasks [7].

Statistics courses require students to not only master statistical concepts but also effectively apply hypothesis testing techniques to real-world data analysis. However, students in economics and social sciences often lack strong programming skills [8]. Therefore, implementing statistical tests such as t-tests, proportion tests, or ANOVA using modern platforms such as RapidMiner can be challenging [3]. On the other hand, traditional tools such as SPSS, while user-friendly for basic statistical procedures, are limited in their ability to integrate with machine learning models—a growing expectation in modern data science education.

While RapidMiner is a powerful platform for machine learning tasks, it lacks built-in operators for basic statistical hypothesis testing, such as t-tests, proportion tests, and ANOVAs [1]. This limitation poses a significant challenge in academic settings where hypothesis testing is a core requirement. Furthermore, students in economics and social sciences often have little to no programming experience, making the integration of external tools, such as Python or R, a significant barrier to teaching and learning [4], [6]. As a result, statistics educators face the difficult task of maintaining a hands-on approach without overwhelming students with complex technical knowledge [5].

To address these challenges, we developed a hypothesis testing solution entirely within RapidMiner's drag-and-drop interface. This method leverages macro parameters to provide user input and embeds Python executable code for backend statistical computation. The main goals of the solution are as follows.

- To perform all hypothesis tests commonly used in economics and social science education.
- To eliminate programming requirements for students.
- Ensure statistical flexibility and accuracy through Python code that runs in the background.

This design allows students to focus on understanding concepts and running and interpreting statistical results visually, rather than spending too much time on learning programming techniques.

2 Methodology

Fig. 1 illustrates the general structure of the solution, a hypothesis testing workflow in Altair AI Studio (RapidMiner) that uses a no-code approach for users. Users start with a test selection process (t-test, proportion test, ANOVA, etc.), and then they provide the input dataset via the context panel and declare the parameters for the test via the Set Macros operator. This allows the user to avoid programming. All the complex calculations are handled by PythonExec, without the user having to understand it. The result is a test table that can help them draw conclusions.

This solution is very useful in education. Specifically, students can practice hypothesis testing similar to R and Python, but with a simpler process. Although SPSS provides users with a graphical user interface (GUI) for statistical testing, it still has some limitations. For example, connecting to machine learning models is not easy. With the proposed design process on RapidMiner, users can perform both classical statistical tests and combine them with advanced machine learning processes. This makes the proposed solution more flexible and more useful for teaching socio-economic students, where many students are not good at programming but still need to use statistical methods in real-world situations. This no-programming design helps students focus on learning the meaning of the tests instead of writing code. The detailed processes and operators we have built are presented below.

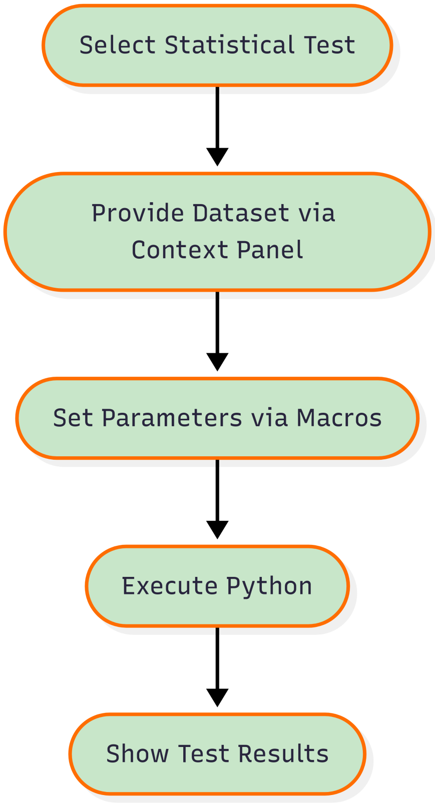


Figure 1: General flowchart of the solution

2.1 Statistical tests implemented

Each hypothesis testing problem is fully integrated into the intuitive workflows of Altair AI Studio (RapidMiner). Logic and computation are handled via embedded Python scripts, while all user-configurable parameters are exposed as macros - ensuring that end users, including students, do not need to write a single line of code. Details of the validation problem types are presented in Table 1.

Table 1: *Statistical tests in our program*

Test Type	**Objective of the Test**	**Integrated Modules**
One-sample t-test	Test the mean of a single variable against a hypothesized value	Python + Macro
One-proportion test	Test the proportion of a category within a population	Python + Macro
Independent two-sample t-test	Compare means between two independent groups	Python + Macro
Paired-sample t-test	Compare means before and after (paired observations)	Python + Macro
Two-proportion test	Compare proportions between two groups	Python + Macro
ANOVA with post-hoc Tukey	Compare means across multiple groups with post-hoc analysis	Python + Macro

2.2 Definition of Macro Names

All hypothesis testing procedures in this solution use the same macro name. For example, macros such as TEST_VARIABLE, TAIL, or GROUP_VALUES are used in all the hypothesis testing. This makes it easy for users to switch between validation types. They only need to update the values without changing the structure. Details of the macro names are described in Table 2.

Table 2: *The macros' names are used*

Macro Name	**Description**
TEST_VARIABLE	The name of the variable to be tested
HYPOTHESIZED_VALUE	The numerical value under the null hypothesis
TAIL	The type of test: ”two” (two-tailed), ”left” (left-tailed), or ”right” (right-tailed)
GROUP_VARIABLE	The name of the grouping variable that contains the categories to be tested
GROUP_VALUES	The specific group values to be tested, separated by double pipe ——

2.3 Data used

To illustrate our solution, a text review dataset for Amazon's shopping service, called Amazon comments, is stored in the local repository of Altair AI Studio.

The preprocessing steps are illustrated in Fig. 2. First, all reviews are converted to lowercase in the “Process Documents from Data” subroutine. Second, Latent Dirichlet Allocation (LDA) is employed by the “Extract Topics from Data (LDA)” operator to categorize the reviews into three distinct topics. All topics are manually identified using the “Map” function:

- Topic 0: General Service Experience,
- Topic 1: Problem Solving and Support Time,
- Topic 2: Refund Request or Escalation

Third, sentiment analysis is performed using the “Extract Sentiment (VADER)” operator. This process creates a new variable called ”Score,” which represents the positive sentiment of each review. The current set of examples is then passed to a subroutine that generates automatic responses to comments and is subsequently stored as “Amazon_reviews_test”, the dataset on which we conduct our hypothesis testing.

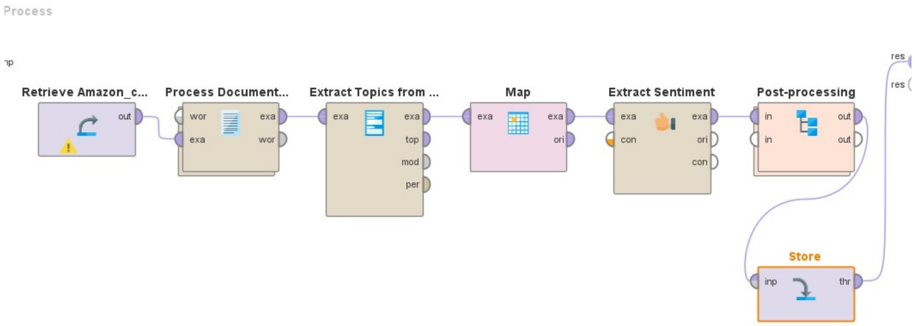


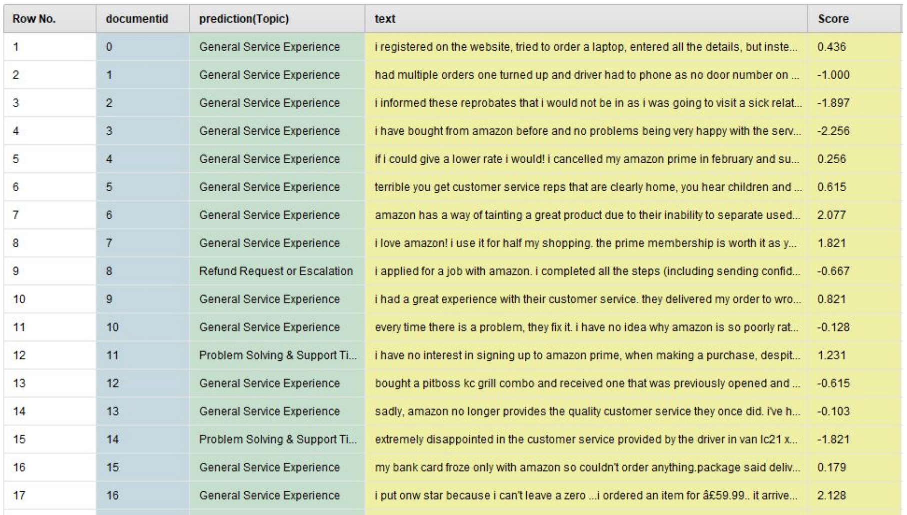
Figure 2: Pre-processing step for Amazon data set

After completing topic clustering and assigning sentiment scores to the reviews, we can perform key statistical tests, such as

- One-sample t-test (to test whether the average satisfaction level differs from a specific expected value),
- Two-sample t-test (to compare satisfaction levels between two groups),
- Proportion tests for one and two groups,
- ANOVA + Post-hoc tests (to compare multiple groups).

3 Illustrated examples

First, let us consider the Amazon review dataset, with some samples illustrated in Fig. 3. It can be observed that the dataset includes three major variables including "text", "prediction (Topic)", and "Score". The variable "text" represents the consumer's review. The variable "prediction (Topic)" indicates the topic of reviews generated by an AI model. Meanwhile, the variable "Score" represents the sentiment. Some descriptive analyses are shown in Fig. 4 and Fig. 5. The descriptive statistics reveal differences in topic counts or proportions, as well as sentiment scores among topics. Notably, consumers tend to have a positive attitude toward "problem solving and support time" related to Amazon, although they comment on this topic less frequently than on others. These results are based on the samples; thus, we need to conduct hypothesis testing to draw conclusions for the population. With a significance level in mind, we aim to determine whether the proportion of "problem solving and support time" is less than $\frac{1}{3}$, which is the mean of proportions. Additionally, we need to ascertain whether there is a difference between the topic scores. To this end, we will conduct the following tests.



Row No.	documentid	prediction(Topic)	text	Score
1	0	General Service Experience	i registered on the website, tried to order a laptop, entered all the details, but inste...	0.436
2	1	General Service Experience	had multiple orders one turned up and driver had to phone as no door number on ...	-1.000
3	2	General Service Experience	i informed these reprobates that i would not be in as i was going to visit a sick relat...	-1.897
4	3	General Service Experience	i have bought from amazon before and no problems being very happy with the serv...	-2.256
5	4	General Service Experience	if i could give a lower rate i would! i cancelled my amazon prime in february and su...	0.256
6	5	General Service Experience	terrible you get customer service reps that are clearly home, you hear children and ...	0.615
7	6	General Service Experience	amazon has a way of tainting a great product due to their inability to separate used...	2.077
8	7	General Service Experience	i love amazon! i use it for half my shopping. the prime membership is worth it as y...	1.821
9	8	Refund Request or Escalation	i applied for a job with amazon. i completed all the steps (including sending confid...	-0.667
10	9	General Service Experience	i had a great experience with their customer service. they delivered my order to wro...	0.821
11	10	General Service Experience	every time there is a problem, they fix it. i have no idea why amazon is so poorly rat...	-0.128
12	11	Problem Solving & Support Ti...	i have no interest in signing up to amazon prime, when making a purchase, despit...	1.231
13	12	General Service Experience	bought a pitboss kc grill combo and received one that was previously opened and ...	-0.615
14	13	General Service Experience	sadly, amazon no longer provides the quality customer service they once did. i've h...	-0.103
15	14	Problem Solving & Support Ti...	extremely disappointed in the customer service provided by the driver in van lc21 x...	-1.821
16	15	General Service Experience	my bank card froze only with amazon so couldn't order anything.package said deliv...	0.179
17	16	General Service Experience	i put onw star because i can't leave a zero ...i ordered an item for â£59.99.. it arrive...	2.128

Figure 3: An illustration for data samples

3.1 One-Proportion test

Our goal is to test whether the proportion of reviews related to "problem solving and support time" is less than 0.33 or not.

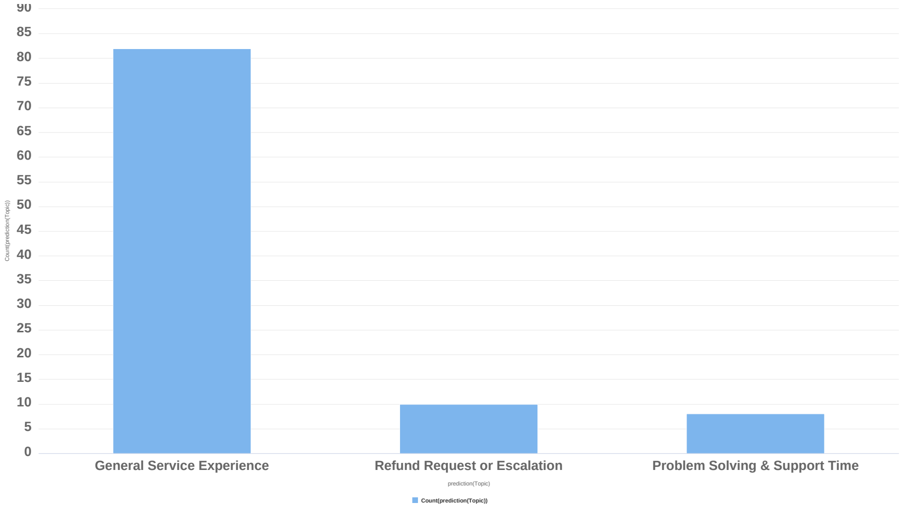


Figure 4: A bar chart for topic's count

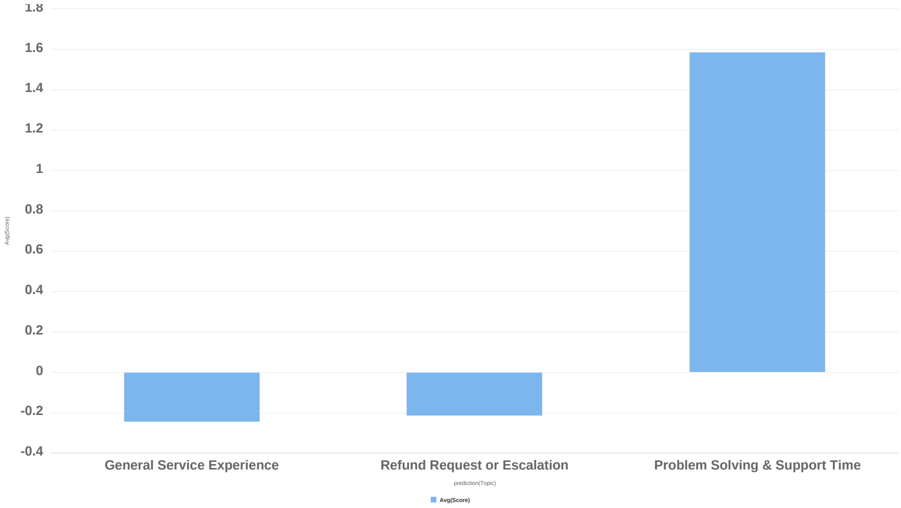


Figure 5: A bar chart for sentiment score according to topic

- H_0: $p = 0.33$.
- H_1: $p < 0.33$.

The detailed procedure is as follows.

1. The learner opens the “One-Proportion Test” process, as illustrated in Fig. 6.
2. The learner does not need to rebuild the workflow or modify the Execute Python operator. The learner only needs to specify in the context table that the input example set is Amazon_reviews_test (Fig. 7).
3. Next, in the Set Macros parameter table, the learner needs to assign values for the existing macros (Fig. 8).
4. After running the workflow, the learner will receive a results table that includes the p-value, z-statistic, and the observed and expected proportions (Fig. 9).

The obtained p-value is approximately zero, allowing us to reject H_0 at nearly any significance level, that is, we can conclude that the proportion of “problem solving and support time” is less than 0.33.

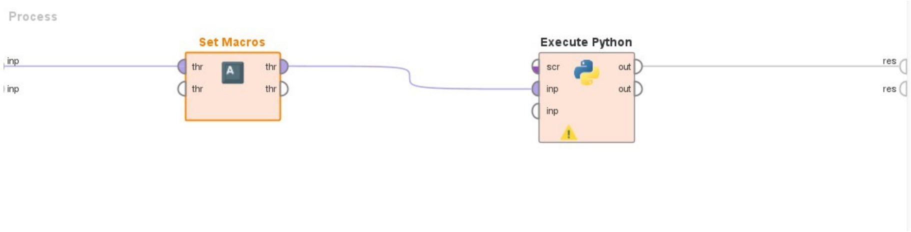


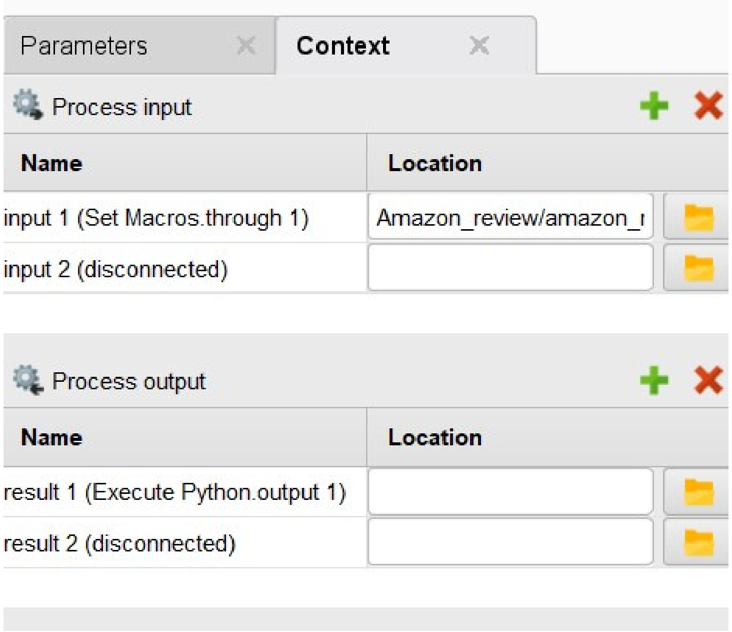
Figure 6: The process “One-sample proportion test”

3.2 ANOVA Test

Consider the Amazon_reviews_test dataset. Our goal is to test whether there is a significant difference in the mean sentiment scores among different review topics.

- H_0: There is no difference in the mean scores across the different topics.
- H_1: There is at least one topic with a different mean score.

The step-by-step workflow is as follows.



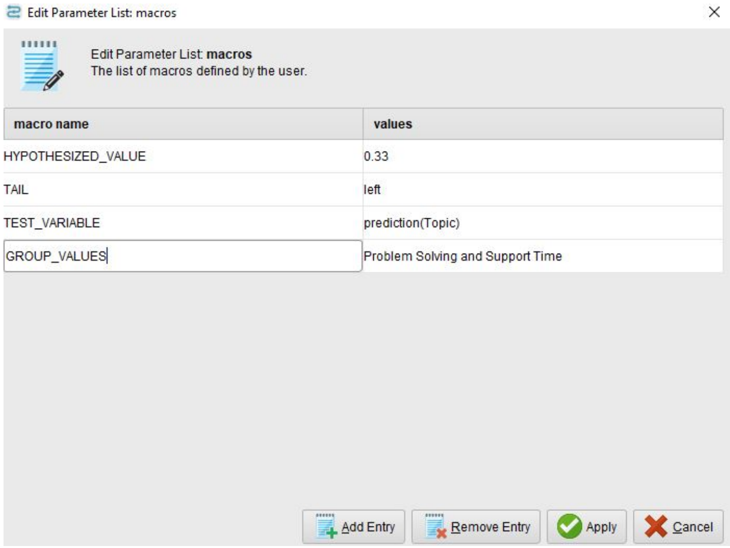
Process input

Name	Location
input 1 (Set Macros.through 1)	Amazon_review/amazon_r
input 2 (disconnected)	

Process output

Name	Location
result 1 (Execute Python.output 1)	
result 2 (disconnected)	

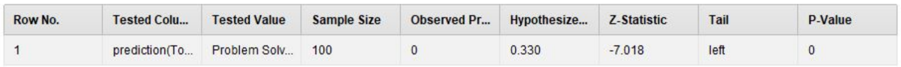
Figure 7: Declare the input (example set) in the Context Panel



Edit Parameter List: **macros**
The list of macros defined by the user.

macro name	values
HYPOTHESIZED_VALUE	0.33
TAIL	left
TEST_VARIABLE	prediction(Topic)
GROUP_VALUES	Problem Solving and Support Time

Figure 8: Declare the value of macros



Row No.	Tested Colu...	Tested Value	Sample Size	Observed Pr...	Hypothesize...	Z-Statistic	Tail	P-Value
1	prediction(To...	Problem Solv...	100	0	0.330	-7.018	left	0

Figure 9: The obtained result

1. The learner opens the "ANOVA Test" process, as illustrated in Fig. 10.
2. The learner does not need to rebuild the workflow or modify the Execute Python operator. The learner only needs to specify in the context table that the input example set is Amazon_reviews_test (Fig. 11).
3. Next, in the parameter table of Set Macros, the learner needs to assign values to the existing macros (Fig. 12).
4. After running the procedure, the student will receive a results table that includes the F statistic, p-value, and, if available, the results of the post-hoc Tukey test (Fig. 13).

Based on this result, we can reject H_0 at a significance level of 0.05. In other words, there is a difference in scores between topics. The post-hoc Tukey test also indicates a significant difference in scores between "problem solving and support time", the most positive topic, and "general service experience", the most negative topic.

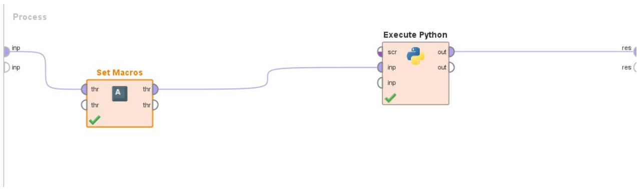
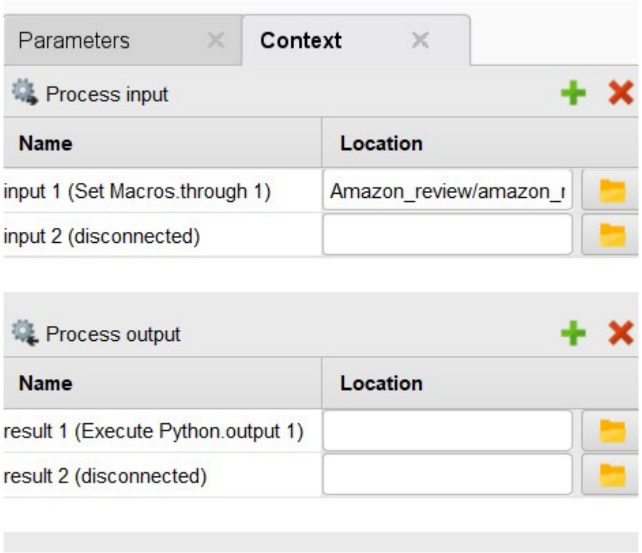


Figure 10: The process for the ANOVA test

4 Results of the evaluation of users' satisfaction and feedback

To evaluate the effectiveness of the no-code solution for hypothesis testing on the Altair AI Studio (RapidMiner) platform, we conducted a pilot implementation with a group of seven postgraduate students in business administration, specifically within the business analytics course. The Altair Group had previously recognized this solution as a "Goldmine Solution." We summarize the results and feedback from the practical implementation process as follows.



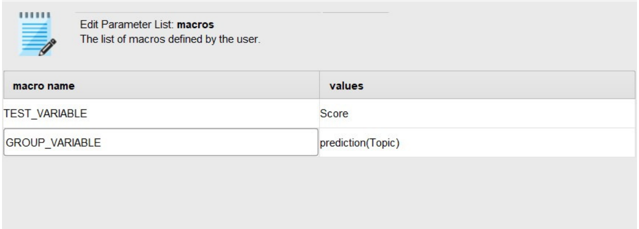
Process input

Name	Location
input 1 (Set Macros.through 1)	Amazon_review/amazon_r
input 2 (disconnected)	

Process output

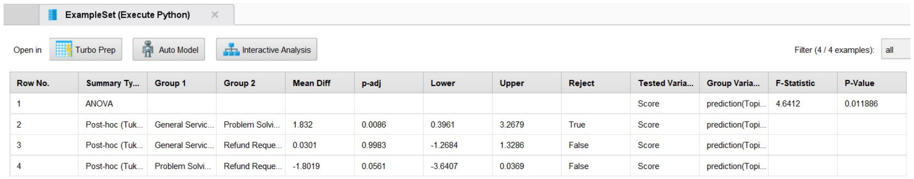
Name	Location
result 1 (Execute Python.output 1)	
result 2 (disconnected)	

Figure 11: Declare the input for the ANOVA test


Edit Parameter List: **macros**
The list of macros defined by the user.

macro name	values
TEST_VARIABLE	Score
GROUP_VARIABLE	prediction(Topic)

Figure 12: Declare the macro values for the ANOVA test



Filter (4 / 4 examples): all

Row No.	Summary Ty...	Group 1	Group 2	Mean Diff	p-adj	Lower	Upper	Reject	Tested Varia...	Group Varia...	F-Statistic	P-Value
1	ANOVA								Score	prediction(Topi...	4.6412	0.011886
2	Post-hoc (Tuk...	General Servic...	Problem Solvi...	1.832	0.0086	0.3961	3.2679	True	Score	prediction(Topi...		
3	Post-hoc (Tuk...	General Servic...	Refund Reque...	0.0301	0.9983	-1.2684	1.3286	False	Score	prediction(Topi...		
4	Post-hoc (Tuk...	Problem Solvi...	Refund Reque...	-1.8019	0.0561	-3.6407	0.0369	False	Score	prediction(Topi...		

Figure 13: The obtained ANOVA test result

4.1 Quantitative satisfaction levels and implementation feasibility

The level of satisfaction and the solution acceptance rate were assessed through the pilot implementation results involving a group of 20 postgraduate students in the Business Analytics course. This no-code hypothesis testing solution was designed to help learners focus on analysis and interpretation of results rather than technical operations and programming. The collected results show that the overall learner satisfaction level with the solution reached 4.57/5. Notably, 100% of the students participating in the trial successfully implemented the provided hypothesis testing processes, confirming the solution's feasibility and ease of use in an educational setting. Fig. 14 clearly illustrates this level of satisfaction. Accordingly, 100% of the students reported a satisfaction level of 'satisfied' or higher, with 'very satisfied' accounting for the higher percentage at 57.1%.

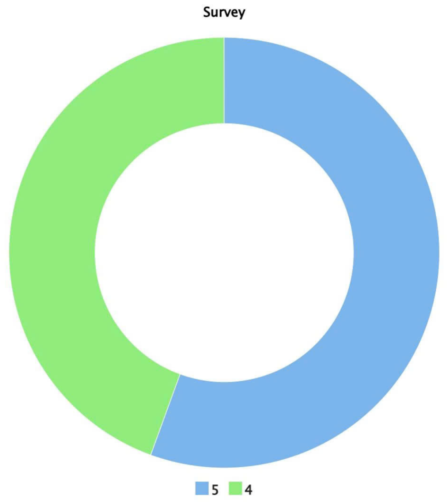


Figure 14: Distribution of learner satisfaction levels with the solution

4.2 Aspects that users most appreciated and were most impressed by

A qualitative analysis of the postgraduate students' experience highlighted the aspects that users most appreciated and were most impressed by when using the no-code hypothesis testing solution. Learners highly valued the intuitive and user-friendly interface of the RapidMiner platform. The drag-and-drop mechanism for components, which replaces programming and coding, was identified as a key factor that helped users to significantly save time and eliminate the pressure associated with syntax, a common barrier. The core benefit of this solution is that it allows users to easily focus on analysis and interpretation of results rather than coding. This helped users intuitively understand the hypothesis testing procedure, thereby enabling them to apply their knowledge to real-world data analysis, moving beyond the limitation of merely memorizing formulas. This feedback is visually summarized through a keyword chart Fig. 15, highlighting the frequency of terms such as “easy”, “RapidMiner”, “intuitive”, “programming”, “drag and drop”, etc.

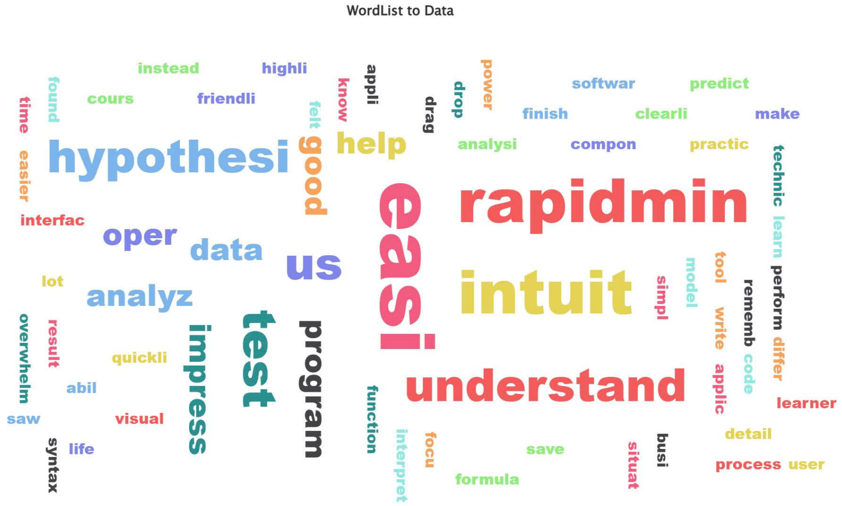


Figure 15: Distribution of keywords expressing learner impressions of the solution

4.3 Suggestions and proposed improvements

To continuously improve the quality of teaching and the solution’s effectiveness, the research team collected open suggestions and proposals from learners. A qualitative analysis of this feedback showed a major need to enhance the applicability and the depth of understanding regarding the operational principles of the analytical process. Specifically, learners suggested adding more practical examples linked to corporate data or academic research to clearly demonstrate the solution’s applicability. They also recommended the need to present in more detail the step-by-step operation within RapidMiner and simultaneously clearly explain the operational principles of the algorithms and tools used. This ensures that learners not only become proficient in operation but also understand the essence of the analytical process.

Furthermore, learners emphasized the necessity of linking theory and practice more closely, as well as providing a general overview of the software (purpose, utility) right from the start of the course, so that students understand the objectives and practical applications of the subject in their work. Another important suggestion was to add a comparison between manual methods (e.g., Excel/SPSS) and the process on RapidMiner to clearly show the convenience and limitations of each approach. The main keywords extracted from the learners' suggestions are presented in the keyword chart Fig. 16.

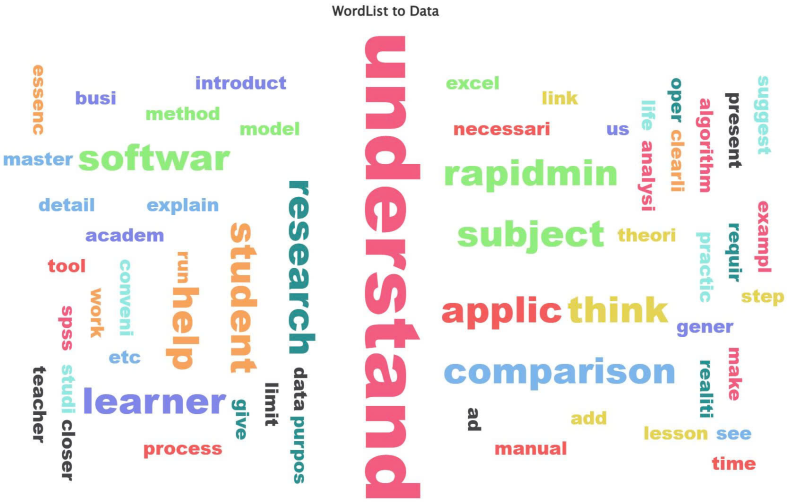


Figure 16: Distribution of keywords expressing learner suggestions and proposed improvements

5 Conclusion

The no-code hypothesis testing solution using Altair AI Studio (RapidMiner) successfully integrated essential statistical testing tools (such as t-tests, proportion tests, and ANOVA) into the no-code platform, addressing the challenge of programming skill deficits among students in the economics and social sciences fields. Practical implementation demonstrated high feasibility and effectiveness, with 100% of learners successfully implementing the processes and overall satisfaction reaching 4.57/5. The solution was highly valued by learners for its intuitive interface and its ability to help them focus on analysis and result interpretation instead of technical operations. Furthermore, using

Altair AI Studio offers a strategic advantage in flexibility and the ability to easily integrate with future machine learning models, surpassing traditional statistical tools like SPSS. However, to optimize learning, qualitative feedback indicates the need to add more practical examples and deeper explanations of algorithmic principles to help learners not only master the operations but also understand the essence of the analytical process. In the future, the solution can be expanded for integration into Altair AI Hub to provide a cloud-based data analytics service.

Acknowledgments

We would like to thank Nguyen Tat Thanh University for their support.

References

[1] S. B. Amin and S. Rahman, *Biogas generation from household level farming*, Energy Resources in Bangladesh: Trends and Contemporary Issues, Springer International Publishing, Cham, 2019, pp. 113–116.

[2] M. Bjaoui, H. Sakly, M. Said, N. Kraiem, and M. S. Bouhlel, *Depth insight for data scientist with RapidMiner - an innovative tool for AI and big data towards medical applications*, Proceedings of the 2nd International Conference on Digital Tools & Uses Congress (DTUC'20), ACM, New York, 2020, pp. 1–6.

[3] M. F. Mustapha, A. N. I. Zulkifli, O. Kairan, N. N. S. Mat Zizi, N. N. Yahya, and N. M. Mohamad, *The prediction of student's academic performance using RapidMiner*, Indonesian Journal of Electrical Engineering and Computer Science, **32**(1) (2023), 363.

[4] N. Hashimzade, O. Kirsanov, and T. Kirsanova, *Integrating programming into the modern undergraduate economics curriculum*, International Review of Economics Education, **49** (2025), Article 100310.

[5] R. Kadar, N. A. Wahab, J. Othman, M. Shamsuddin, and S. B. Mahlan, *A study of difficulties in teaching and learning programming: A systematic literature review*,International Journal of Academic Research in Progressive Education and Development, **10**(3) (2021), 591–605.

[6] M. Li, *Teaching beginner-level computational social science: Interactive open education resources with learnr and shiny apps*, Frontiers in Education, **8** (2023).

[7] J. N. Van Rijn, V. Umaashankar, S. Fischer, B. Bischl, L. Torgo, B. Gao, P. Winter, B. Wiswedel, M. R. Berthold, and J. Vanschoren, *A RapidMiner extension for open machine learning*, RapidMiner Community Meeting and Conference, 2013, 59–70.

[8] A. van Witteloostuijn and J. van Hugten, *The state of the art of hypothesis testing in the social sciences*, Social Sciences & Humanities Open, **6**(1) (2022), Article 100314.