HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

DEPARTMENT OF MECHANICAL AND MEDICAL ENGINEERING

MASTERS IN SMART SYSTEMS

MASTER THESIS REPORT

IMPLEMENTING ADAPTIVE LEARNING AND DATA ANALYTICS FOR INTERACTIVE E-LEARNING

By

SAGAR MANJUNATH

Matriculation number: 273096

Under the supervision of

Dr. Edgar Seemann

Dr. Phoebe Perlwitz

DECLARATION OF ORIGINALITY

I hereby solemnly affirm that I have completed this master thesis entirely on my own with no unauthorized assistance from others. All the sources, references, materials, images that I have used here and any passages that were quoted directly or indirectly have been properly acknowledged and fully cited in the bibliography section at the end. I confirm that every aspect of this research is a result of my independent research and effort and all the intellectual contributions of the other authors have been clearly and accurately acknowledged.

Wolfsburg, 31.10.2024 Sagar

Place, Date Signature

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Dr. Edgar Seemann from the faculty of Mechanical and Medical engineering and Dr. Phoebe Perlwitz who are responsible for the design of this project, for their individual guidance, patience and expertise without which the completion of this project would not have been possible. Dr. Phoebe Perlwitz, despite her busy schedule, has invested a lot of time and energy in providing me with proper guidance and helped me propel in the right direction with this project through the online meetings every week. The valuable guidelines from both my supervisors have definitely deepened my understanding not only on data analytics and machine learning but also on handling a project and presenting it in the right way.

I would also like extend my heartfelt thanks to IT department led by Dr. Seemann and Dr. Perlwitz for designing the KISS ME webpage as this was one the key factors of the project for data gathering and achieving the desired results.

ABSTRACT

The main focus of this master thesis is to build an interactive E-learning platform through data analytics and implementation of adaptive learning.

Despite the many advances in the field of technology and science, it has been evident that many students struggle to perform well in their studies which may be because of outdated learning techniques, amount of time spent on the course or sparse prior knowledge in the related field of study. With the old traditional teaching methods which is a traditional classroom teaching, it becomes very difficult for the teacher to individually handpick each student and address the problem. In order to circumvent this problem, online learning platforms also called as Learning Management System (LMS) have emerged in the past few years which stores all the information that is relevant to a student's performance in the course such as past grades, number of times a course material was accessed, period of engagement with a course and so on which allows the faculty to prudently review each student progress in the course. Although it makes the work of an educator easier, it is still time consuming to scrutinize every individual when the population is too large.

This is where the learning analytics comes in which performs descriptive analytics to identify trends and relationships in their performance and can also help in the diagnosis of the solution as to why it is happening. By further using regression and classification algorithms, predictions can be made on students as to how will they perform in the future by establishing relationships between certain parameters.

The next step would be to perform a prescriptive analysis in an attempt to obtain the best outcome and figure out the best methods in learning to improve students' performance iteratively and improve the result through each iteration to find the optimal solution.

With the help of this learning platform, each student's strengths and weaknesses can be recognized based on numerous parameters which will be taken into consideration during analysis and utilize the result of this analysis to build an adaptive learning model which will lay out the path for each student to strengthen their weak points and incorporate intervention strategies like providing them with additional resources and even offering one-one tutoring if possible. A re-assessment can be conducted once the necessary intervention has been reinforced and verify if the algorithm has really helped the student in improving their grades.

TABLE OF CONTENTS

Contents

TABLE OF FIGURES

INTRODUCTION

CONTEXTUAL BACKGROUND

It is a well-known fact that Artificial learning and Machine learning are implemented in almost every technological centre in the modern world as they help in automating most of the processes which are monotonous in nature and thereby help in saving resources and time. Over past few decades, there has been a massive noticeable development in the field of education as well and many new innovative technologies such as smart classes, digital learning, online tutoring have been introduced into the world of education which has made Distance Education (DE) possible and learning more effective at the same time. [1]

A Program for International Students Assessment (PISA) by OCED evaluates the ability of 15 years olds' in the field of mathematics, reading and science every 3 years. The 2022 research showed that the students had hit the all-time low with 30% missing on the basic fundamentals in math and 25% in reading in comparison to 2018 studies. In New York Times, Sarah Mervosh has stated that the mathematics scores among the 15-year olds' in US have also significantly dropped when compared to 20 years back.[2], [3].

Going back to the most recent PISA examination that took place in Germany in the year 2022, where 6,100 15-year olds' participated, the reports indicated that there was a difference of 11 points in mathematics between the male and female which was 7 back in 2018. The following graph depicts the points scored in mathematics in each PISA examination that took place between 2006 and 2022 [3].

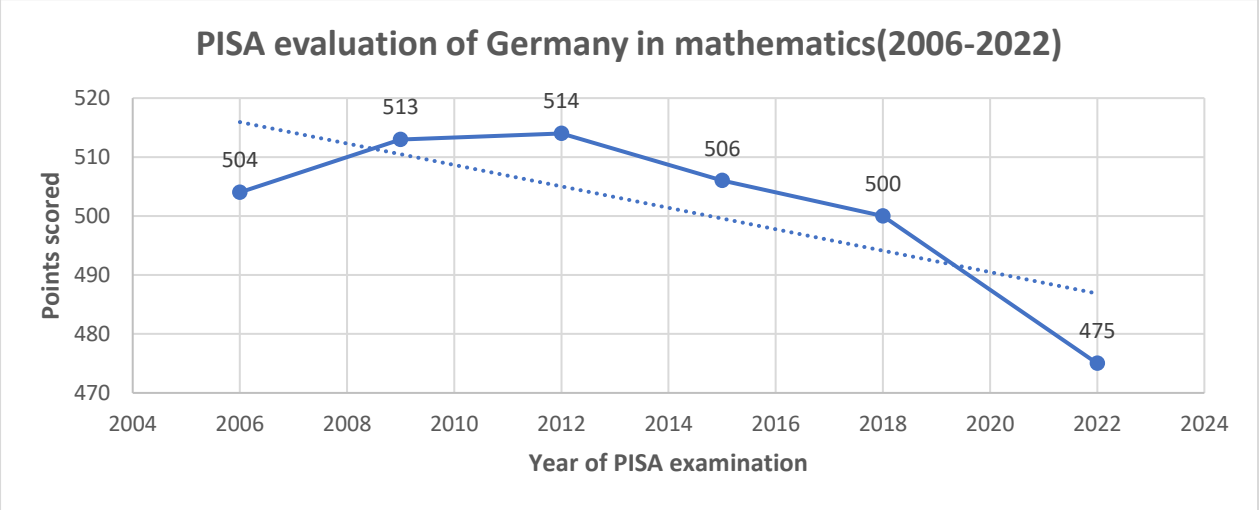


Figure 1: PISA findings for 2006-2022

From the above plot and the data that has been presented, it can be clearly seen from the trendline that there is a steady plunge in the points scored by the students over the years which is either an indication of lack of means to reach out to students or the increasing inefficiency of traditional means which in turn emphasizes the need for targeted intervention to enhance the students' performance.

With the implementation of Digital education, the other growing concern in the field of education is keeping the students engaged and making sure they are performing well [4]. But it is nearly impossible for a teacher to tailor the learning methods that are coherent with individual students learning needs and comfort [4].

The most common approach that's is being implemented in most pf the educational institutions is to maintain a pace with which the average graders are most comfortable as it strikes a balance between all class of students [4].

This method of generalisation has a good impact over low to mid-size class population but in order to deal with a bigger population size, one needs a more robust approach which focuses on individuals' strength and weaknesses. This is where the implementation of data analytics plays a vital role in the field of education [4], [5]

To take the digital education one step further, data analytics has entered the world of education a decade ago in order to evaluate the performance of student based on various factors such as their age, prior knowledge on the subject, course of study and cluster them into groups based on their learning rates and past grades [5], [6]. The first step in the field of data analytics is to identify the group of individuals of similar learning styles, strengths and weaknesses and based on these parameters, a tailored learning method which is in accordance with each groups assessment is laid out [5]. In the preceding step, the effectiveness of the model will be continuously monitored through further assessment of students' performance and also taking into consideration, the feedback of the students [5].

This method of collecting and analysing data about the students is called "Learning Analytics" [7]. It helps the institution faculty to make predictions of who can drop and exit a course and which students can thrive and come out with flying colours. The biggest contribution of learning analytics in the betterment of student's performance is the offering of personalized learning [7]. The data that is gathered by a Learning Management System (LMS) gives the course instructor a rough idea of how a student might proceed through the rest of the course [7]. For instance, through research conducted by Smith, Lange and Hudson (2012), they were able to glean information like the frequency of logging into the LMS, time duration of engagement with the learning content and assignment grades and utilize this information to forecast an individual's performance in the course [7].

The extent to which learning analytics can be scaled has no limits ranging from a small course to a national level and its application at each level creates a different unique value as the LMS serves as a pedagogical platform by incorporating a digital environment [7], [8]. Speaking

about its application at a university level, it helps in equipping faculty with the information about student to know where they are struggling and at what particular point the faculty has to make an intervention and simultaneously helps the students in retaining the content which they have learned and help them focus on topics where they need the most help [7], [9]. For instance, a student might not have accessed a certain learning material for a long time which when notified to the faculty by the LMS, the faculty might be able to give the student the motivation required [7]. In another scenario, the grades of someone who had a good track record in the past might drop. An intervention from the faculty at this point can help the student to make up for what he/she was lacking and the usage of LMS among the students have an impact across numerous subjects and languages such as mathematics, science and technology [7], [8]. In this way, when a faculty continuously keeps track of the students' progress and their engagement duration with the course, they will be able to create a link between students' performance and their grades which helps them to tailor the course in an even better way and several studies have shown a statistical increase in mathematics scores in comparison to those who went through the traditional classroom approach[7], [8], [9].

Despite many successes in the implementation of learning analytics in the field of education, there are still many concerns which are very difficult to overcome such as privacy concerns, biases and the possibility of giving rise to educational inequalities [7], [10]. For instance, some students might get very uncomfortable knowing that they are constantly being observed and monitored by their faculty and this feeling of angst cannot be simply eradicated by the LMS system [7]. There are also scenarios where some of the subjects require hands-on training which is not very effective through online E-learning [11]. In addition, interpersonal issues of students which cannot be factored into the analytics because of which some of the "why's" remain unanswered renders the learning analytics tool futile in some scenarios [7]. Furthermore, there is constant need for faculty involvement if the system has to make a big stride towards success [7].

Another big threat in this field is the concern for data privacy. One has to follow certain guidelines regarding legal and ethical issues, like Family Educational Rights and Privacy Acts (FERPA) in the United States which imparts rights to the students and their respective family to review their educational records maintained by the institution or to disclose such records to only certain third parties like government officials and accrediting organizations [12]. Similarly in Germany, one has to follow the guidelines set by German Datenschutz to protect the privacy of students being ethical about the data being collected from the students and most of the data protection policies in Germany is handled by General Data Protection Regulation (GDPR) [13]. Within the scope of this project, HFU data guidelines are taken into consideration which in turn adheres to GDPR or DatenSchutz-GrundVerordnung (DSGVO) for the purpose of collecting, storing and processing of the personal data of the students without the possibility of any data leaks which can lead to misuse of students' personal information [14].

LITERATURE REVIEW

Over the past decades, as already mentioned in the introduction section, the implementation of data analytics and adaptive learning in the field of education has been one of the most useful ways of utilising AI and data analytics as it has helped many students make bigger strides in accomplishing academic heights which without them would have been an arduous task. In order to ensure the success of AI implementation in digital education, large volume of data from the students such as their performance in their previous exams, engagement patterns and the way each student prefers to cover a topic have been gathered and fed to the AI and Learning Analytics tool in order to gain valuable insights about the students so that the AI tools can help the students in enhancing their performance in the best way possible. As our research has its focus set mainly on the improving the performance of students in mathematics, three simple factors have been taken into consideration, namely age, gender and field of study to determine which of these three factors has the most influence on their performance in mathematics and how best can that information be utilised to lift their grades up. Furthermore, many side factors such as time consumed by each individual to solve a task correctly and number of times help was accessed to solve the task were factored in as even they are crucial in designing the necessary targeted intervention as per the needs of each individual. Taking these factors into account, the following Literature Survey conducted gives an overview of the previous researches conducted in this field, gaps in those researches and possibilities for future implementation.

IMPACT OF AGE, GENDER AND FIELD OF STUDY ON MATH PERFORMANCE

An attempt to investigate the correlation between age, gender and academic achievements among math and science students comprising of 332 students (223 Females and 109 males) exhibited a linear relationship between age and their grades in mathematics with age having a variance of 0.1% while gender with a relatively higher variance of 1.2% while both maintaining a positive correlation with the math grades proving that gender was a better predictor among the two [15]. Gender difference in STEM performance arise mainly from the early educational experiences [16]. A study utilizing a longitudinal data from Italian universities by Piruallа and his team with its focus set on freshmen and their academic trajectories over the four years showed that the males scored relatively higher grades in mathematics than the females while females outperformed males in care-oriented courses like biology [16].

A study conducted on 132 students enrolled in IT courses to determine whether or not the programming skills acquired by students had an impact on their math skills showed that students enrolled in technical courses on programming were equipped with strong mathematical skills than those enrolled in non-technical courses [17]. It has also been

revealed that students enrolled in STEM courses tend to perform better in solving mathematical tasks than those enrolled in non-STEM courses [18].

Many other studies also underscore the importances of age and gender in educational strategies to enhance outcomes in diverse university settings [19], [20]. A study on student approach to learning by Warren Lake and William Boyd comprising of 560 students with majority of the demographic over 70% identifying as mature aged revealed a significant age effect on age approaches with mature aged students scoring higher for complex problems that require integration of knowledge from multiple concepts and lower for the tasks that require rote memorization and superficial understanding [19]. This information tells us that aged students require more attention when it comes to memorising formulas and concepts which is relatively easier for the younger pupils. Few studies in which the male pupils have scored slightly better than the female folks in mathematics have been attributed to differences in cognitive and math reasoning abilities [20], [21].

RESEARCH GAPS AND SUMMARY

Although the above study provides a comprehensive understanding of correlation between age, gender and field of study and mathematics performance among university students, many studies do not dive into the underlying reason for the correlations. For instance, the strong correlation between gender and higher grades has not been attributed to anything in few studies which makes it hard to provide a proper guideline in improving their performance. Moreover, many of the studies are cross-sectional rather than longitudinal which does not factor in the changes in grades over time leading to significant loss of data. It can also be seen that most of the studies makes an attempt to answer the question which gender or age group shows better performance in mathematics but provides no feasible solution as to enhance the performance of a particular age group or gender that requires timely intervention.

It can be briefly said from the above survey that an attempt has to be made to neutralise the differences in math grades between the two genders regardless of whether it is due to the early educational experiences or due to the existing stereotypes by laying more emphasis on female students through targeted individual intervention which can be made possible through the implementation of data analytics in E-learning.

ROLE OF DATA ANALYTICS AND MACHINE LEARNING IN DIGITAL EDUCATION

Now that some of the factors that influence the performance of university students have been introduced in the previous section, this section of the literature review looks at the implementation of learning analytics and machine learning in digital education in order to optimise the performance of the students.

In order to measure the performance of math students, two quizzes covering different topics with consistent difficulty levels were designed for 5 fifth year students [22]. The study revealed that multiclass decision tree algorithm achieved a highest accuracy of 95.16% in predicting their performance in mathematics among multiclass decision forest, multiclass logistic regression and multiclass Neural Network [22]. Furthermore, findings also revealed that students could have the same overall score yet differ significantly in their understanding of different subtopics [22]. Nevertheless, the data collected was limited to one session which may not represent a students' true capabilities and also discusses about the potential for improvement by factoring in more significant variables like the total time taken by student to complete the quiz which gives a deeper understanding of a students' status about a particular subtopic [22].

Implementing ML models in digital education enables simultaneous analysis of all strategies which is in contrast with the traditional regression models with limited number of variables [23]. A study incorporated 154 learning strategies and cooperative behaviours of German mathematic students to explore the factors influencing final exam success. The analysis revealed that rote memorization negatively impacts the performance while having a thorough contemplation of the subject and explaining it in one's own words came out as the strongest predictor from the ML algorithm implemented [23]. However, the author also talks about some limitations one of which is the challenges with High-dimensional data as multi-collinearity can complicate the interpretation of the result [23]. Another challenge to the implementation of ML in education is its heavy reliance on quality and completeness of the data as the results might lead to inaccurate predictions in case of biased or incomplete data [23].

THEORITICAL FRAMEWORK

STATISTICS FOR DATA ANALYSIS

Every process that takes place in the world results in generation of some information and this collection of information is termed as data. With the help of this data, it is possible to define the process being executed and in turn predict the future outcomes of it with a definite probability. Numerous definitions of statistics highlight its dual nature as both art and science which has its focus set on reasoning, data collection and decision making under certain uncertainty. In general, the process of acquiring data, making it more uniform and compatible for interpretations and pulling insights out of it is called data analysis which is made possible through a branch of mathematics called statistics [24], [25].

The basic text book definition of statistics is that it is a branch of mathematics that deals with the collection, analysis, interpretation and presentation of data. It is well-known fact that in

today's world, every industry regardless of its scale of operation ranging from leading financial institutions to a small grocery shop produces data of some sort which when goes through an analysis generates some useful insights that can be implemented for improving the efficiency of the organization [25].

METHODS

DESCRIPTIVE STATISTICS

It is the most fundamental building block of statistical analysis that are used to quantify the data and represent a set of numerical data through graphical visualisations. The important measures of descriptive statistics include:

1. ***Measure of Central tendency*** – Which tells us how centred the distribution of the data is based on parameters called *'mean', 'median'* and *'mode'* which are nothing but 'average of the distribution', 'centre value of the distribution' and 'value with the highest frequency'.
2. ***Measure of Dispersion*** – This shows the extent to which the data is spread and includes parameters such as *'standard deviation', 'variance'* and *'skewness'*
3. ***Frequency distribution and Graphical representation*** – This tool helps us in obtaining a visual representation of the frequency at which different values occur and also visualise the distribution of data across a spectrum of categories. Some of the common graphical representations include
    - **Histogram:** Gives the distribution of numeric values with a specific number of bins/intervals.
    - **Box plots:** A graphical representation of mean, median, the spread of the dataset given and also the outliers.
    - **Pie charts:** A visual representation of percentage of each category occupied within a dataset

[26], [27]

INFERENTIAL STATISTICS

This is the part of statistics where conclusions or inferences are drawn based on the data obtained by performing descriptive statistics on particular small sample taken out of a larger population and these inferences are later applied across the entire population. Care should be taken to ensure that the sample size selected strikes a balance between being adequately large enough but not so large to make the analysis too computationally expensive and at the same time yield statistically meaningful outcomes. Some of the inferential statistics techniques are: [26], [27]

1. ***Hypothesis test***: Aims at verifying significance of specific outcomes rather than attributing them to chances alone
2. ***Corelation analysis***: Assists in analysing the corelation between different variables
3. ***Logistic or Linear regression***: they facilitate inference and prediction of causality between variables.
4. ***Confidence Interval:*** Helps in determining the likelihood of occurrence of a predicted outcome.

[26], [27]

STATISTICAL POWER

Statistical power also known as sensitivity can be defined as the likelihood of some significance test outputting an effect in case there is any. In this context, an effect can be a real, non-zero relationship between the variables in a given population. The probability that a significance test will correctly reject a false null hypothesis, is often overlooked in research planning. Ignoring power can lead to erroneous conclusions and missed opportunities to identify significant effects in data [28], [29].

In terms of statistics, statistical power is the probability of avoiding type-II error. Statistical power helps in determining the parameters required to attain a targeted power level for the study under consideration [29].

Some of the advantages of statistical power are:

1. By understanding the role of sample sizes which is a key contributing factor to statistical power, one can confidently design a study that possess satisfactory probability of detecting the effects that can generate the desired output. [28], [29]
2. Just as low statistical power is incapable of detecting the effects of a significance test, an increased sample size almost always has high costs associated with it as the relationship between sample size and statistical power is not linear. [28], [29]

Hence, it is necessary for a researcher to estimate the sample size required by his analysis to generate the desired effect before gathering the data as the slightest of miscalculation in this step could make the whole analysis unreliable and the following actions would be erratic [29].

DATA GATHERING

Quantitative research mainly focusses on structured methods like surveys and experiments to gather tangible data which can be analysed statistically to draw conclusions. Collection can take place through numerous steps five of which are explained here.

1. One of the infamous and easiest technique is to survey a group of people with several open-ended questions which results in the accumulation of qualitative data.

2. The next most common technique which a data analyst resort to is a direct one-one Interview with a set of specific questions and the responses are often times recorded or transcribed in parallel to the interview. Although this is a time-consuming process, it results in a fairly accurate data as people tend to be honest in a face-to-face interview compared to an online survey.
3. Third method is the utilisation of focus groups where people respond to spontaneous questions and conversational prompts. Through this the moderator can capture the reactions of the participants over others perspective and gather data more precisely. The ideal group size is considered to be around 8 to 10.
4. The fourth method of data extraction is a traditional one which includes observation using the senses.
5. The last data extraction method would be through analysis of textual content when one has to scrutinize the changes undertaken in official or organizational views in a specific context.

All these techniques can be carried out either on digital platforms or offline. But ever since the recent pandemic, there has been several challenges in the traditional offline methods such as social distancing which has caused a shift towards online alternatives to keep the research alive. online data collection methods allow researcher to maintain the integrity of their studies while simultaneously adhering to safety protocols. Online surveys enable participants to respond at their convenience facilitating easier data gathering than the traditional approaches which is rather rigid due to their fixed schedule and place of gathering. The automated collection of responses enhances the research process by streamlining the analysis. Moreover, online platforms provide tools for randomization which is essential for smooth collection of data.

While this was a brief overview of most common data gathering techniques, each of them has its own pros and cons and the selection of the technique is purely left to the analyst depending on the type of data required for the analysis. Nevertheless, regardless of the technique opted for, it has to be particular enough to generate coherence between the data collected and simultaneously be broad enough to bring forth a wide spectrum of insights.

[30], [31]

For the purpose of our study, a web-based questionnaire has been implemented to gather data from a group of students who are the target audience of this research.

MACHINE LEARNING ALGORITHMS

By taking statistics a step further, one can build algorithms that make predictions and forecast data from a given set of data or even create a regression analysis by instigating a relationship between two or more variables, recognize hidden patterns in data which cannot be detected by human eyes, classify data points in a set based on their relevance in some aspect. These

algorithms are termed as machine learning algorithms which are now explicitly computational techniques, which enables computers to learn from data and thereby make predictions, classify data and take the appropriate decisions. By leveraging these statistical techniques and computational power, it is possible to extract valuable insights from a given datasets and implement it in applications spanning from image recognition and Natural Language processing (NLP) to recommendation system and predictive analytics. Furthermore, it is also possible to incorporate them in business analytics where they go by the name predictive analysis and are used to find solutions to business-oriented problems and improving the efficiency of an existing business model or creating a new one. Similarly, efforts have been made to implement the same for educational purposes through which the performance of the students can be enhanced through prediction, classification and proper diagnostic measures.

[32], [33]

Machine learning mimics human learning through computational modelling while deep learning makes use of multiple layers of feature extraction. ML approaches are categorized into supervised, unsupervised, and reinforcement learning, with supervised learning being the most prevalent in educational applications. The most commonly used ones are Neural networks and random Forest for the purpose of predicting students' performance and drop out chances. Feature selection has been shown to enhance predictive accuracy for student performance in terms of online education when some of the significant factors are included. Several institutions are including ML as a tool for grading and testing students and also enabling the faculty to provide instant feedback on students' performance. Many instances have shown that the predictions that the algorithms make improve over time with the increase in availability of the relevant data.

[33], [34]

Machine learning algorithm can be broadly classified into three categories. But Supervised ML has been used extensively for this project due to the data size presented to us.

SUPERVISED LEARNING

It is a type of algorithm based on a model which makes use of a data set having both a set of input values and the corresponding output data set for each input. This enables the algorithm to learn the relationship between the two or more variables and by exposing the algorithm iterative to such data sets, it can generalise patterns and predict the output variables for new unseen data. The data that is supplied to the algorithm for the purpose of learning is called the 'training data' set. Once the algorithm gets a grasp of the training data set, it generates an optimal function that enables it to predict the outcomes of inputs that were not a part of the training data set which is also known as 'Test data'. With every passing prediction, if the accuracy of prediction for new unknown dataset increases, one can say that the model is capable of performing the task in the right way.

[35], [36]

The application of supervised learning spans from predicting customer churn in business to diagnosing diseases in healthcare institution where many of the tasks which were once human-driven are now being automated. [35]

The supervised machine learning can be further classified into 'Classification' and 'Regression' which are discussed further in detail.

I. REGRESSION ALGORITHMS

Regression analysis is one of the simplest and powerful statistical method used to contemplate and quantify the relationship between a set of independent variables and dependent variables. The most widely used purpose of regression analysis is to determine the value of dependent variable also known as response variable based on the values of independent variables known as regressors. By fitting a suitable mathematical model to analyse the data, one can make predictions about the future outcomes based on the existing relationship between the variables and even assess the strength and direction of the established relationship between the variables. [37]

To be precise, the two main application areas of regression analysis are as follows:

→Prediction and forecasting in which prediction of new outcomes will be made based on the relationship established between two or more variables and this functionality can be extensively used in various fields of machine learning [37]

→Secondly, regression analysis can be used to simply determine the relationship between the dependent and independent variables. [37]

Some of the common regression analysis algorithms are as follows:

i. LINEAR REGRESSION

The simplest form of supervised regression algorithm is linear regression which as the name itself indicates, tries to establish a linear relationship between the dependent or response variables and independent variables or regressors. If the regression is performed explicitly on one variable, then it's simple linear regression and in case of multiple variables, it becomes multiple linear regression.

In order to perform linear regression, one has to employ the linear predictor function which is of the following form:

$$f(i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip},$$

Where x_{ik} is the k-th independent variable for i-th datapoint. β_0 is the intercept and β_1 to β_p are the weights for each dependent variable of each data point respectively.

The dependent variables are usually continuous and unbounded while the independent ones can be continuous, discrete or even categorical. By implementing regression, one can answer questions like to what extent one of the phenomena have over the others and how intricate the relationship between each phenomenon.

The predicted values should be as close as possible to the corresponding actual response. Let's say 'Yi' is the actual response and f(xi) is the predicted value. Then yi – f(xi) is called the 'residual'. Smaller the value of residual is, more accurate is our model.

Once the predictor function is established between the variables, the next step is to optimise the weights or coefficients and the intercept resulted from the predictor function. One way to obtain the best weight is to minimize the 'Sum of Squared Residuals' or SSR given by the expression:

$$SSR = \Sigma_i (y_i - f(\mathbf{x}_i))^2.$$

This technique is called the method of Ordinary Least Squares (OLS) and is performed iteratively till the SSR value approaches a global minimum.

PERFORMANCE OF LINEAR REGRESSION MODEL:

The parameter for determining the performance or accuracy of the implemented linear regression model is "Co-efficient of determination" denoted as R^2. Basically, R^2 indicates the amount of variation of the dependent variable y, that can be explained. Larger the value of coefficient of determination, better will be the fit obtained.

A value of $R^2 = 1$ is a perfect fit with SSR = 0. Generally, a R^2 of 0.9 to 1 is preferred in a regression model to capture every aspect of a problem and produce precise outcomes. [38], [39], [40]

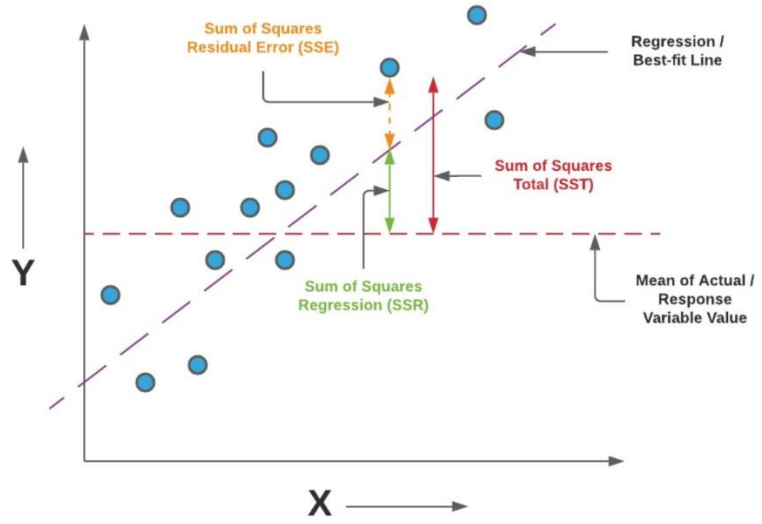


Figure 2: Simple Linear regression [41]

Having a look at the diagram above, most of the things are self-explanatory. The Sum of Squares of Regression (SSR) is the factor that explains the variance in the model while the Sum of Square of Error (SSE) is the amount of variance that is not explained by the model. [40], [41]

F-STATISTICS:

Another parameter that indicates the explainability of variance offered by the regression model built is "F-statistics". BY taking F-statistics into account, one can estimate the overall significance of the model which helps in decision making as to whether the model requires further improvement or not. Also, f-statistics makes the comparison between the performances of different models possible which further helps in adding or removing the required variables.

F-statistics is given by the following expression:

$$\mathbf{f = (SSR/DF_{SSR})/(SSE/DF_{SSE})}$$

Where,

DF_{SSR} – Degree of freedom for SSR and is equal to number of co-efficient, p (β_1 to β_p)

$$DF_{SSR} = p$$

DF_{SSE} – Degree of freedom for SSE and is equal to the difference between number of records, N and number of co-efficient, p

$$DF_{SSR} = N\text{-}p\text{-}1$$

Larger the value of F-stats, greater will be the amount of variance explained by the model.

[40], [42]

LIMITATIONS

1. **Limited flexibility**: The linear relationship assumption between the model variables limits its usage in most of the real-world applications such as time series problems which are non-linear in nature

2. **Susceptible to outliers**: As the regression uses the OLS model to determine the coefficients which finds the best fit line by minimising the SSE, any outlier will inflict a squared effect on the models' prediction which derails the accuracy of the model severely. [43]

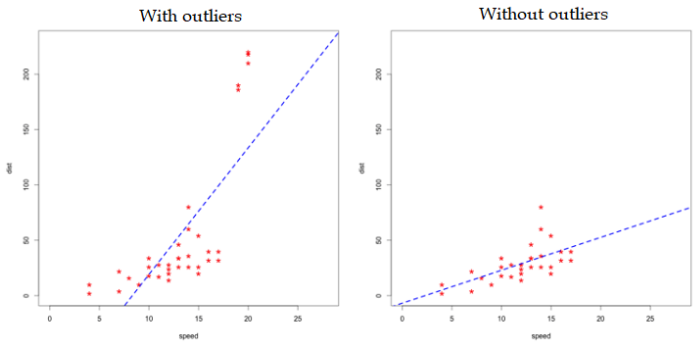


Figure 3: Effect of Outliers on Linear Regression [43]

II. CLASSIFICATION ALGORITHMS

The algorithms help in segregating the given dataset into multiple classes based on certain level of closeness and similarities are known as classification algorithms. By analysing the training dataset, classification algorithm assigns target classes to a given data set and labels them into different classes. Classification algorithm falls primarily under supervised learning methods which implies that the predictions require a training data set. This powerful algorithm plays a vital role in numerous field such as spam mail detection, image recognition, medical diagnosis and various other everyday applications. Once the dataset is categorised into several classes based on their similarities, the behaviour of different elements in dataset can be easily predicted based on the prevalence in their characteristics. For instance, the classification algorithm segments the customers who visit an E-

commerce website more often as someone who is more likely to make a purchase than those who never return. This helps the company in identifying potential customers.

The steps taken by classification algorithm in handling the dataset can be broken down as follows:

➔ Initially a dataset consisting of labelled elements is gathered where each element is labelled into a particular class based on its feature. This is then split into training dataset and test dataset

➔ A training dataset is fed to the algorithm which acts as the fundamental block of the classification algorithm and the test dataset evaluates the models' efficiency by utilising the dataset whose classes are unknown to the algorithm.

[44], [45]

Some of the most widely used classification algorithms are listed as follows:

i. LOGISTIC REGRESSION

While the previously discussed regression algorithms help in establishing a relationship between the regressor and the response variables which are continuous variables, logistic regression is a supervised classification algorithm that has discrete variables and is an important tool for binary classification problems. An example can be making prediction as to whether or not a cancer can be malignant in nature or not.

Logistic regression is primarily based on sigmoid function or logistic function to map out the predicted probability values of output for a given input value.

The sigmoid function is defined as follows:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Where σ: R → (0,1) represents the probability of dependent variable belonging to a either of the two cases with 't' being the output variable itself. The output variable 't' is a linear function of a single explanatory variable which can be expressed as:

$$t = \beta_0 + \beta_1 x$$

With the above expression, it is clear that the output of logistic regression lies between 0 and 1.

Once the model outputs a probability, a threshold value (p) is commonly used to assign the class labels. If the predicted probability is greater than 'p', then the instance is classified as class 1/positive and class 0/Negative otherwise.

- If P (Y = y_i | x) > p, classify as class 1.
- If P (Y = y_i | x) ≤ p, classify as class 0.

Since the logistic regression is essentially the sigmoid function of linear regression, the proper model equation can be given out as:

$$\sigma\,(t) = \frac{1}{1+e\text{^}[-(\beta o + \beta 1 * x1 + \cdots + \beta m * xm)]}$$

where,

σ is the predicted probability output of a given variable

β1, β2 …. βm are the model weights

x1, x2 …. Xm are the input features [46], [47]

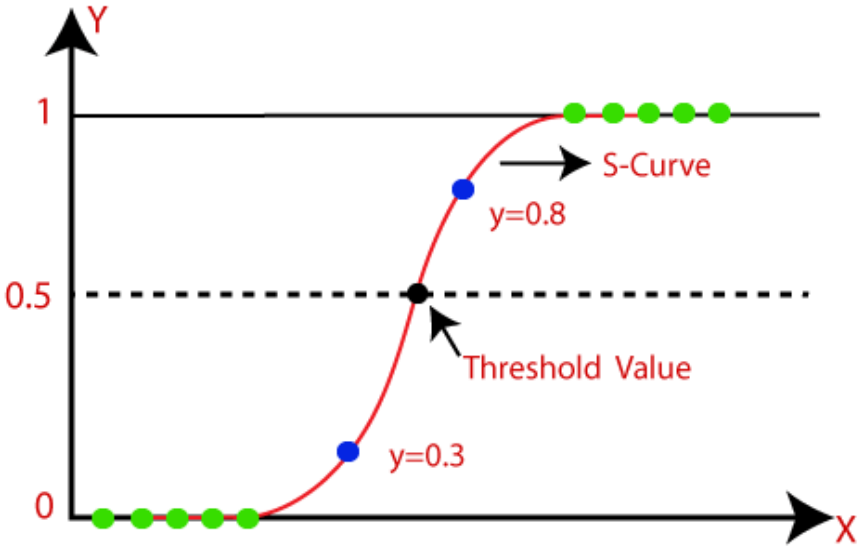


Figure 4: Logistic regression [47]

PERFORMANCE METRICS

Once the model is trained, several performance metrics are evaluated which are appropriate for classification problems. They are discussed as follows:

1. **Accuracy**: The percentage of correct predictions out of all predictions

ACCURACY = (TP + TN)/ (TP + TN + FP + FN)

2. **Precision**: Proportion of correct predictions that are actually positive.

   PRECISION = TP/ (TP + FP)

3. **Recall**: Proportion of actual positives that are correctly predicted as positives.

   RECALL = TP/ (TP + FN)

   From the above expressions it is clear that fewer FP's are needed in order to increase the precision and FN is irrelevant whereas fewer FN's are needed in case of higher recall and number of FP's doesn't matter.

   Raising the classification threshold diminishes the FP's thereby increasing precision. But this raise in classification threshold simultaneously reduces the TP's or keeps them constant while increasing the FN's or keeps them constant – either of which will bring the recall down.

   This tells us that it is impossible to have high precision and recall at the same time as they are inversely related to each other and is called recall/precision trade-off

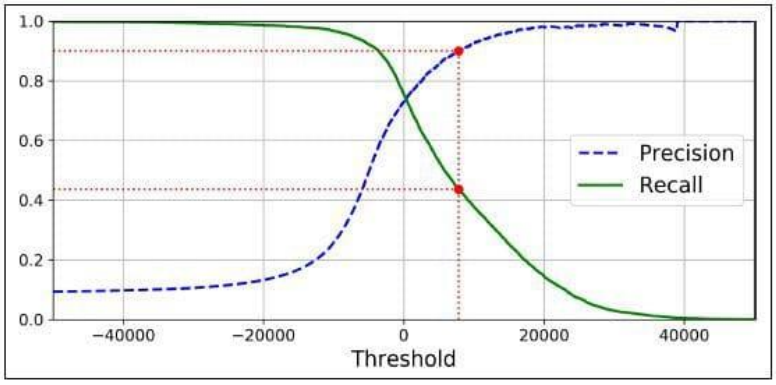


Figure 5: Precision Recall trade-off [48]

4. **ROC curve and AUC**: Also known as Receiver operating Characteristics (ROC) curve is an evaluation metric that can be used to lower both FP and FN. It is a plot of False positive rate (x-axis) against True positive rate (y-axis):

- True Positive Rate = TP / (TP + FN) (a.k.a sensitivity)
- False Positive Rate = FP / (FP + TN) (a.k.a inverted specificity rate)
- Specificity = TN / (TN + FP)

The objective is to keep the False Positive rate as low as possible while the True Positive rate higher to achieve a higher accuracy. This way the ROC curve evaluates how well the model distinguishes between two classes across different threshold values.

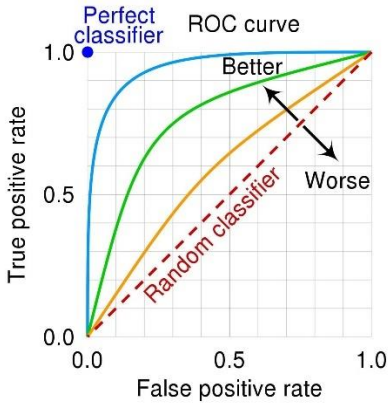


Figure 6: Different modes of ROC [48]

The AUC or Area Under Curve estimates the area under the ROC curve for all possible classification thresholds. The more the AUC, better the performance of the model.

[46], [48]

LIMITATIONS

1. The assumption of linearity between the features and the log-odds sometimes leads to under-performance of the model.
2. The linear decision boundaries made by the logistic regression may not be applicable to non-linear models which is mostly the case in real world scenarios. [49]

APPLICATIONS

1. It is widely used in predicting mortality in injured patients.

2. When used as a binary classification algorithm, it helps in identifying spam emails. [46]

ii. DECISION TREE

Decision tree is a non-parametric supervised ML algorithm which can handle both classification and regression problems. It helps in creating a tree like model of decisions based on features in the dataset fed to the model. They help in decision making in numerous fields like business, engineering, medical, law and many more. The tree structure of a decision tree is hierarchical and it consists of root node, branches, internal nodes and leaf nodes. [50], [51]

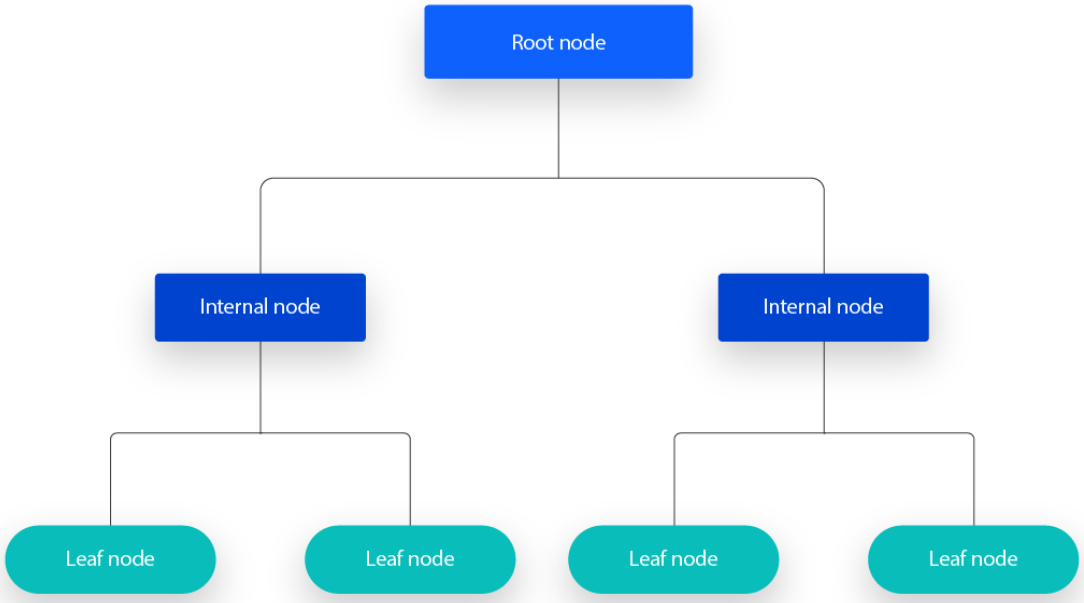


Figure 7: Building blocks of a decision tree [52]

The process of decision making in a decision tree starts from the root node which then branches out into the internal nodes which are also called the decision nodes. The features available in the dataset given are evaluated by these internal nodes based on which homogeneous subsets are formed which are also denoted as leaf nodes or terminal nodes. These terminal nodes represent all possible outcomes within the dataset fed to the model. This structure repeats itself until all or most of the features are segmented into different classes. The method by which the internal nodes split into leaf nodes is through simple true or false questions and the decision-making process is refined through information gain which dictates whether or not a feature should be used to split a node. [50], [51], [52]

It is to be noted that it is relatively easier to attain pure leaf nodes with smaller trees and the complexity increases as the size of the tree grows and the purity level of the pure leaf nodes goes down with bigger tree models. This phenomenon of reduced purity in pure leaf nodes due to large size of the tree is known as fragmentation and often lead to overfitting. One of the simple remedies for this is 'Pruning' which necessarily eliminates the branches that uses the features with low information gain. [50], [52]

Some of the popular decision tree algorithms are as follows:

- **Iterative Dichotomiser 3 (ID3)**: Makes use of Entropy and information gain as metrics to perform feature splits.
- **C4.5**: Makes use of information gain and gain ratios as metrics to evaluate split points.
- **Classification And Regression Trees (CART)**: It makes use of Gini impurity to identify the ideal attribute for feature split. Lower the Gini index, more ideal will be the split. [52]

IDEAL ATTRIBUTE SELECTION METHODS:

- **ENTROPY AND INFORMATION GAIN:**
  Entropy mainly calculates how impure the information is and ranges from the value 0 to 1. It is expressed as follows:

  Entropy (S) = - Σ p(C) * $\log_2$p(C)

  Where,

  S – The dataset whose entropy is calculated.

  C – classes in the given data set

  P(C) – Fraction of data points that belong to the dataset C.

  In order to select the ideal feature for splitting at the internal nodes, one with the smallest value of entropy has to be selected.

  The information gain based on entropy is the difference in entropy before and after a split on given attribute. Higher the information gain, more efficient will be the split. It is expressed as follows:

  Gain = Entropy(p) – (Σ p(i) * Entropy(i))

  [50], [52]

- **GINI IMPURITY**

  Gini impurity is defined as the measure of probability of incorrectly labelling a data point if it were labelled randomly and independently. In simple terms, it tells us how impure a dataset is with 1 being fully impure and 0 being fully pure. It is expressed as follows:

  GINI = 1- (Σ_i p(i))

[50], [52]

PERFORMANCE METRICS

There are several techniques to evaluate how efficient a decision tree is and some of the important ones are as follows:

1. Accuracy: The percentage of correct predictions out of all predictions

   ACCURACY = (TP + TN)/ (TP + TN + FP + FN)

2. Precision: Proportion of correct predictions that are actually positive.

   PRECISION = TP/ (TP + FP)

3. Specificity: Also known as True Negative Rate is as follows:

   SPECIFICITY = TN/ (TN + FP)

4. Sensitivity: Also called True Positive Rate is expressed as:

   SENSITIVITY = TP/ (TP + FN)

[50]

LIMITATION:

1. **Susceptible to overfitting**: As the tree complexity increases, its flexibility towards the generalization of new data goes down resulting in overfitting. This can be avoided with the help of pruning process.

2. **High variance estimators**: It is possible that a decision tree becomes very different even for a slight variation in the data. This scenario can be

circumvented through the process of bagging which reduces the variance of decision trees.

[51], [52]

FEATURE IMPORTANCE

As the name goes, feature importance gives us an idea of how relevant a feature is in predicting the output for a given dataset. This feature of decision tree helps in reducing the dimensionality by eliminating the irrelevant feature and retain only those that are most pertinent to the output of the dataset. Gini index is primarily used in the calculation of feature importance. In general, feature importance helps in cutting computational cost, avoiding overfitting and enhancing interpretability. [53], [54]

iii. SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is another supervised ML algorithm used for classification and regression tasks. The classification of the dataset by this algorithm takes place through an optimal line called hyperplane that has a definite margin from each of the classes on either side. SVM's can handle both linear as well as non-linear classifications with the help of kernel trick which maps the input into N-dimensional feature spaces after which the non-linear classification can be handled as linear classification task. [55], [56]

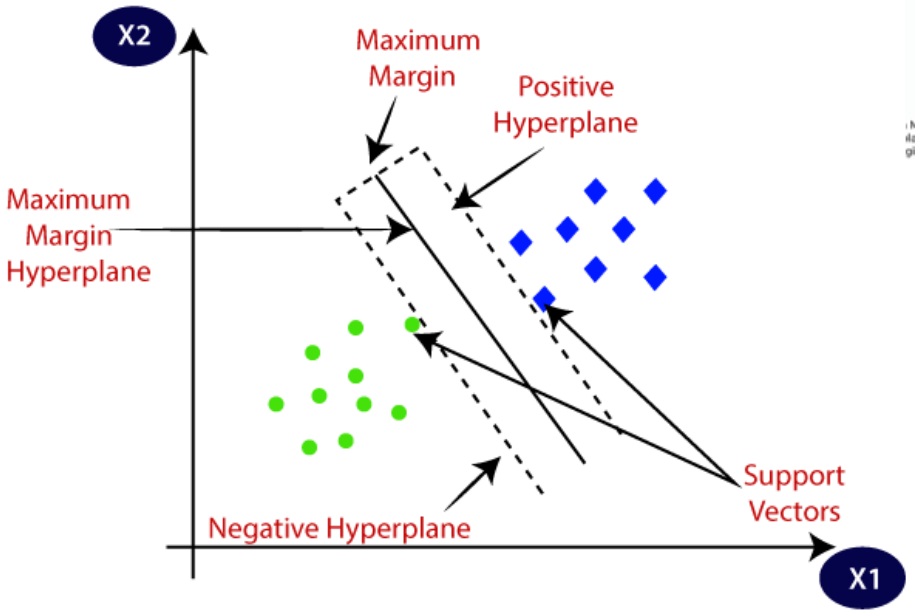


Figure 8: Components of SVM [57]

As it can be seen from the diagram above, the main goal of the SVM algorithm is to find the maximum margin called the hyperplane between the two segmented datasets with which the distance between the hyperplane and the data points on either side is maximum. The data points on either side which are closest to the hyperplane are called support vectors and they influence the position and orientation of the hyperplane. The gap between these support vectors and the hyperplane is termed as margin which is to be maximised by the SVM algorithm. As there is a possibility to determine multiple hyperplanes to differentiate the datasets, it enables the algorithm to generalize well to the new data quickly and make accurate predictions. [55], [56], [57]

SVM classifiers can be classified into two types:

- **Linear SVM**: In the use case of linear SVM's the dataset need not undergo any transformations for the purpose of getting classified. The separating hyperplane can be mathematically represented as follows:

  W*x + b = 0

  Where,
  W → Weight vector,
  X → input vector and
  b → Bias term.

  The two methods by which the margin can be calculated are hard-margin classification and soft-margin classification.

  In case of hard-margin classification, the elements in the dataset will be clearly outside of the support vectors and is represented as follows:

  (W*x + b)*y >= a where a is the margin projected onto weight W.

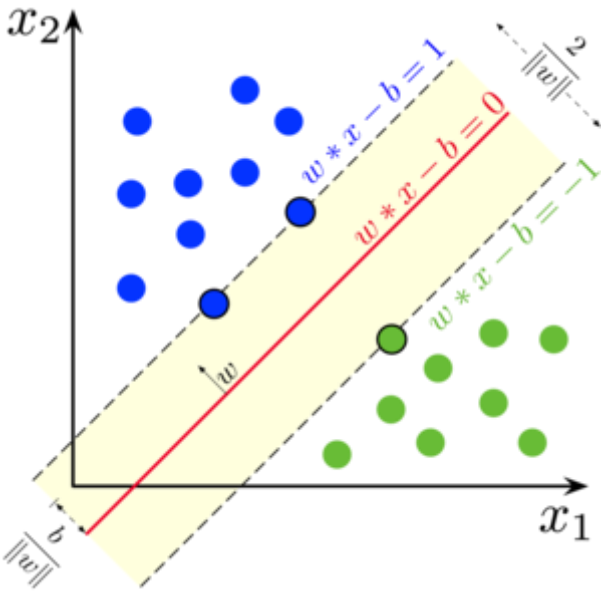


Figure 9: Hyperplanes and margin [56]

On the other hand, soft-margin classification allows misclassifications up to certain extent when the dataset is not linearly separable.

The goal of the soft margin classification is to minimise the following function:

$W^2 + C*[(1/n) * \Sigma max(0,1 - y_i*(W*x_i - b))]$

Where y is the i-th target and the parameter C makes sure that the margin is appropriate and x_i lies on correct side of margin. A larger C value narrows the margin down thereby minimising the misclassification while a smaller C value widens it up allowing more room for misclassification. [55], [56], [57]

- **Non-linear SVM**: There are certain real-world scenarios where the data is not linearly arranged due to which they cannot be separated into a straight line. In such cases, non-linear SVM's are made use of. Firstly, the training data is transformed into N-dimensional feature space which employs dot product method for vector multiplications. But this in turn increases the computational complexity which is why kernel trick is implemented along with the SVM algorithm which replaces these dot product calculations with kernel functions. Some of the famous kernel functions are polynomial kernel, Gaussian or Radial Basis Function (RBF) kernel and sigmoid kernel. [55], [56], [57]

The working of a non-linear SVM is pictorially represented as below:

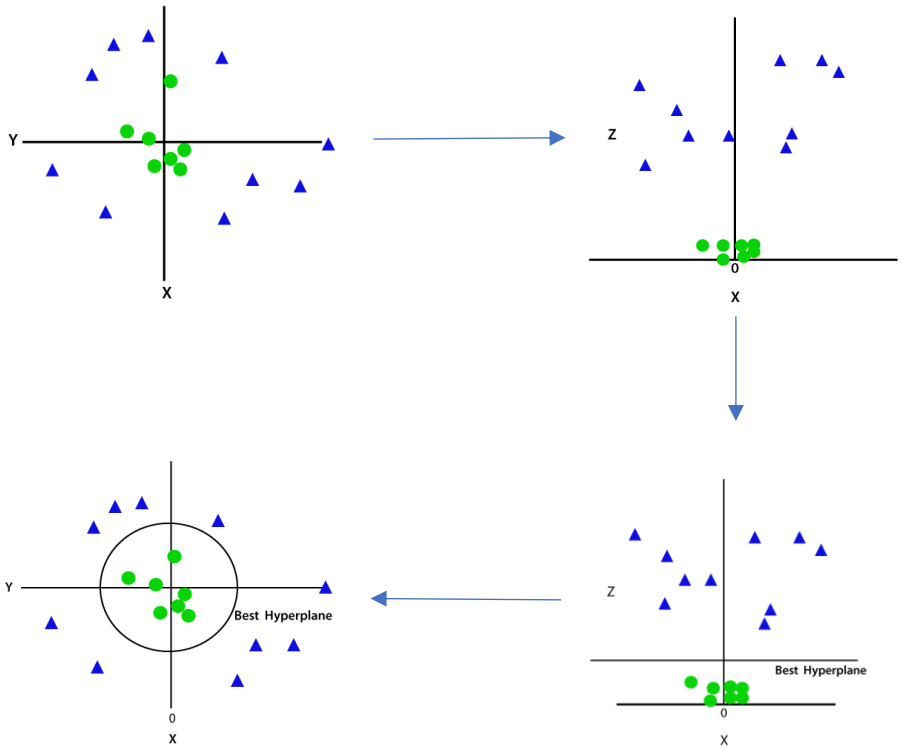


Figure 10: Implementation of Non-linear SVM for Non-linear data [57]

EVALUATION METRICS

The evaluation metrics used in the case of SVM are almost the same as that of decision tree and logistic regression which are precision, recall and accuracy. [56]

LIMITATIONS

1. **Requirement for full labelling:** SVM requires full labelling of input data which makes it computationally more expensive than the other algorithms.
2. **Difficult to interpret:** Interpretability of SVM is in comparison more difficult. [56]

APPLICATIONS

1. The most common use case of SVM's is Natural language processing (NLP) for the purpose of text classification and sentimental analysis as they are compatible with higher dimensional data.
2. It is also further used in image classification tasks like object detection which proves to be useful in security tasks.
3. It also helps in retrieving Geographic information in analysing the layered geophysical structures underground and predicting seismic liquefication potential of soil. [55]

LEARNING ANALYTICS

As discussed earlier, learning analytics is field that primarily focuses on collecting, analysing and reporting data about learners and implement the necessary steps to optimize the learner's performance. According to the consolidated model of the learning analytics, learning analytics is essentially a combination of data science, design and theory. The theory aspect of learning analytics is responsible for the choice of questions being directed towards the target demographic, hypothesis testing, analysis of the data and explaining gaps and anomalies in the data. The next aspect is the design which comprises of interaction and visualisation of the online platforms and also the design of the concepts being learned by the participants. Lastly, the data science component of the learning analytics takes care of collection, analysis and reporting of the data which helps in optimising the learner's performance. The composition of data analytics takes the following form: [58], [59], [60]

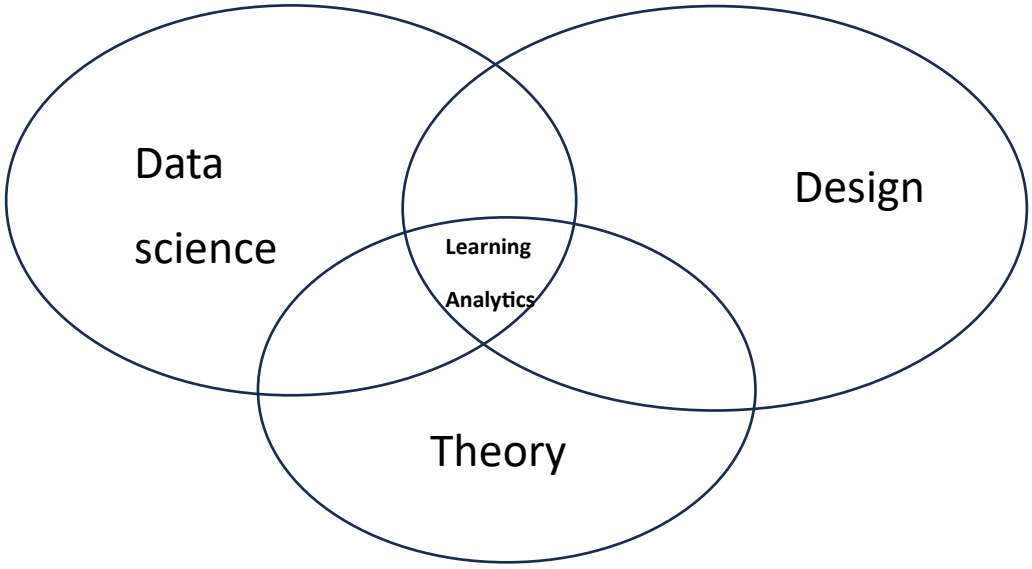


Figure 11: Components of Learning Analytics [59]

The first and foremost step in learning analytics is to gather data from various sources such as existing databases and through online platforms which is then subjected to statistical and

computational methods to identify hidden patterns and trends from the collected data. This analysed data is later on presented in a easily understandable formats such as dashboards and charts to make the process of decision making easier. In the next step, ML models are implemented to predict the students' performance which helps in identifying those who are at the imminent risk of dropping out and based on the individual student's needs, learning experiences are tailored for each individual learner. With the help of adaptive learning, the course contents and difficulty levels are adjusted based on learners' progress. Finally, there has to be a medium for the communication to happen between the faculty and the learner as some cases might require timely intervention to see where the student is lacking and implement the required actions. [58], [61], [62]

Basically, the process of learning analytics can be broken down into four main phases and they are as follows:

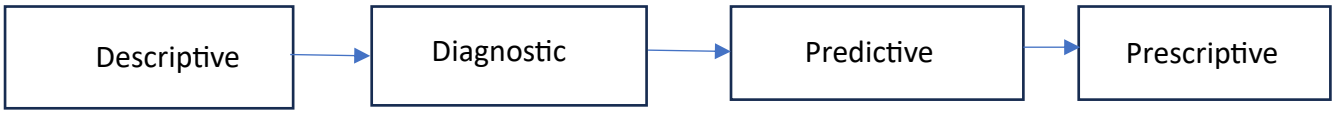


Figure 12: Phases of Learning analytics [61]

1. Descriptive phase answers the question what is taking place and helps in building a report on the previous data and identifying trends and pattern. This helps in obtaining a comparison between the performance of current learning system and the previous learning system.
2. Diagnostic analysis speaks about the reason behind the phenomenon that took place in the descriptive phase and chalks out the relationship between the entities responsible for certain output. This phase also helps in detecting abnormalities in the learning management system where further inspection is required.
3. The third phase is the predictive phase which identifies the future performance of the students based on which adjustments can be made by the faculty in the learners' content.
4. The prescriptive phase is responsible for generating the suggestions based on the learners' performance by tailoring the content as per the individual's needs.

[61]

PROSPECTS AND CHALLENGES OF LEARNING ANALYTICS

Learning analytics which can be defined as an automated system for analysing the data from Learning Management System (LMS) is slowly becoming indispensable part of higher education in today's world. It is mainly an application of big data which is aimed at improving learning and teaching outcomes and driven largely by the ease of data gathering required for the analysis and personalizing an individuals' learning experiences. The outcome of the analysis relies heavily on the machine learning techniques that requires careful selection of

dataset and the ML algorithms in order to obtain valuable findings. It is to be noted that the effective implementation of learning analytics takes place only when timely interventions by the faculty is incorporated based on the predictions made by the LA. Despite its many benefits, research indicates that the expected outcomes have not yet been achieved through LA because of some of the challenges being thrown at it. Some of them include:

1. Inconclusive empirical studies
2. Oversimplified interpretations of data
3. Making the right decision about data and ML algorithms
4. Ethical implications on the usage of data acquired

Along with relying on the quantity of data, equal importance has to be given to qualitative insights in order to present a comprehensive understanding of the student engagement and this requires careful selection of data and the right choice of algorithm for analysis. A formal approach to LA implementation where the institutions focus on gathering useful data and minimising the collection of sensitive information and simultaneously concentrating on a specific problem which students are facing rather than enormous amount of technological implementation can lead to high probability of success of LA in educational institutions. It is also advisable to lay more emphasis over predictive analytics rather than the descriptive analysis as the former offers more actionable insights through which students' performance can be enhanced. Furthermore, ethical considerations have to be at the forefront of LA as the students' data privacy and integrity is of utmost importance. [63], [64]

OBJECTIVES

In this study of implementing Adaptive learning and data analytics for interactive E-learning, an attempt has been made to build an adaptive learning model which performs a descriptive statistical analysis on the student's performance data obtained through a questionnaire which comprises of diverse math topics such as quadratic equation, percentage reckoning, vector calculations and many other fundamental concepts in math with varying difficulty levels and strategically implements various regression and classification algorithms to cluster students based on their performance and provide them with adaptive suggestions so as to improve their performance and help them improve their grades.

A website with the name "KISSME" with a set of math questionnaire of various topics and difficulty level has been set up by a set of IT professionals using typescript which generates a log file and stores it in the intended database. The log file will be consisting of sensitive information about the students like their demographic information namely age group, gender and their study course and the variables utilised while solving the questionnaire like time taken to solve the problems and number of times a student required help and whether the student solved the question correctly or not.

RESEARCH QUESTIONS

1. How do factors such as gender, age and field of study (STEM/non-STEM) of university students influence their performance in mathematics?

   H0: Age and gender, both have a positive correlation with the students' performance in mathematics

2. What conclusion can be drawn from analysing usage parameters (time taken and help count) to solve the exercises and student information?
   H1: Students from non-STEM background require more time in solving the problems and also tend to access more help in comparison to STEM background students

METHODOLOGY

As with any other project, the current thesis also begins with the defining the end goal of the project. As this study is related to data analysis in the field of education, the next steps would be to gather the data and analyse them. Once the results of the analysis are obtained, suggestions will be made to the students in order to improve their performance and finally each student's progress will be closely monitored to determine the efficiency of the model implemented and necessary changes will be incorporated through the process of iteration to reach optimisation. A detailed explanation of each step has been chalked out below.

IDENTIFYING THE OBJECTIVES

After a thorough literature survey, some of the feasible research questions in context of "implementation of data analysis and adaptive learning in E-learning" whose answers can be utilised towards the betterment of student grades in math are as stated above in the objective section.

The objectives are defined in accordance with the course curriculum and aims at identifying the weak points of the students and help them to improve their performance through personalized suggestions and timely intervention.

DATA COLLECTION AND CLEANING

Once the objectives are defined, the next step would be to glean the data required for the analysis. The method of data collection implemented in our study is through a set of math questionnaire of various topics with varying difficulty levels set up on a website called "KISSME" developed by IT specialists through Typescript.

Every student taking part in the course will be then requested to take part in the quiz via distribution of Flyers to their college mails IDs. After the completion of the questionnaire, each student's response will be stored in a JSON log file into our database in a raw form.

The data collected from the student will not contain any sensitive information about the student and will be in accordance with the college data protection guidelines. Any data collected will be treated as confidential and will be solely used for analysis and helping the students perform better.

The target data that are going to be collected from the students are as follows:

1. Age group
2. Field of study (STEM/ non-STEM)
3. Gender

Apart from this, the resources utilised by the students during the questionnaire such as time taken to complete each problem and number of times help was accessed in solving each problem will also be logged.

The math topics selected for the analysis of students' performance are as follows:

1. Quadratic equation
2. Percentage calculation
3. Interest calculation
4. Logarithmic rules
5. Vector manipulation
6. Power rules
7. Polynomial division
8. Linear equation

Once the data has been garnered, the next step would be to categorize the collected data. Python in VS code will be used for this purpose. The data collected will be in a raw unstructured form with many key-value pairs (dictionaries). So, it would be necessary to segregate these dictionary items in each column to extract the data we need and place them in a separate column so that it can be further used for analysis. Python modules such as 'Numpy', 'Pandas' have been used for restructuring the data.

DATA ANALYSIS

After the data has been cleaned and segregated according to the needs of the analysis, the next appropriate step would be to build a descriptive-statistics based on the data collected to

get a general idea of how the data is distributed and how the data can be further classified for the benefit of the students. Python modules such as 'Scikitlearn', 'Numpy' has been used to build a general descriptive statistic table as follows:

Once a numeric representation of how the data looks like has been obtained, it would be necessary to visualize the data using various plots. The python module 'Matplotlib' has been extensively used for this purpose.

ADAPTION STRATEGIES

After the analysis, the next step would be to start defining the solution as to how the performance of each student can be improvised. This includes the following steps:

1. Suggesting individualised Learning paths based on the weak points identified for each student through the conducted survey. This can be in the form of additional learning materials, extra classes, one-to-one online short tutoring that has been tailored to address the specific needs of each student.
2. Utilising online adaptive learning platforms that adjusts the difficulty level of the content based on each student's performance.
3. Receiving personalized feedback about how the new system that has been implemented is helping them improve their performance so that measures can be taken to strengthen the areas where the students particularly lack.
4. In case the struggle of a student with a particular topic persists, the faculty will be notified about the same and an intervention has to take place at this point to help the student
5. Lastly, it's necessary to track the progress of each student and see whether the adopted strategy is effective or not so that outcome can be optimized.

EVALUATION AND ITERATION

Once the whole model is set in motion, it is time to continuously monitor the efficiency the system through student feedbacks and analysis of learning outcomes. The evaluation and iteration phase plays a crucial role in ensuring the effectiveness of the data-driven adaptive learning system. After the initial implementation, continuous monitoring and assessment are required to improve student outcomes and refine the system further. This phase includes assessing the impact of the strategies, gathering feedback from various students, and making necessary adjustments based on real-world results. The steps involved in this phase can be broken down as follows:

1. ***Initial performance assessment:*** This step gives an idea about the effectiveness of the strategies implemented based on the students' performance by comparing students' performance on math quizzes and tests before and after the implementation of the adaptive learning techniques and creating a progress report based on their

performance metrics such as number of tasks solved correctly, time take per question and number of helps requests that tracks their progress on selected math topics.

2. ***Student and Teacher feedback:*** Through surveys and questionnaires, students' perspective on their learning experience can be gathered, including the difficulty level, learning materials available, usefulness of suggestions and overall satisfaction with the learning strategies.
3. ***Analysing the learning outcomes:*** With the help of different statistical tools trends in students' performance is compared between different age groups and genders which identifies the areas for further intervention.
4. ***Iterative adjustments:*** Based on the student's feedback and the analysis conducted in the previous step, the system is continuously refined towards its optimal state. This can be done by modifying the adaptive learning strategies like adjusting the difficulty levels for certain group of students, adding more relevant materials and adjusting the intervention timing of the faculty.

Once the above steps are implemented, this phase will operate as a perpetual improvement loop where each cycle of initial assessment, feedback, analysis and adjustments lead to better outcomes.

EXECUTION

OVERVIEW

In this section, an explanation will be given on how the above steps have been implemented along with the tools which have utilised for the purpose. To begin with, a website was designed by the IT department led by Dr. Edgar Seemann and Dr. Phoebe Perlwitz where the students enter their information about their gender, age and course required for the further analysis. After the completion of this step, they will be taken to a page where they will get to choose between variety of math topics with different difficulty levels. All the answers entered by the students along with the parameters such as time take, number of helps accessed will be logged as a JavaScript file which will be utilised for analysis with the help of python and its relevant libraries. Based on the analysis results, the hypotheses have been answered and personalised suggestions to each combination of age, gender and course group will be made through which students can improve their performance.

DESIGN OF WEBSITE AND DATA GATHERING

The website "KISSME" which stands for KISS Math Exercises can be divided into two sections and each of which is explained below:

1. DATA GATHERING PAGE

This is the first page of the website where the students enter their demographic information particularly their age group, gender and their field of study. The page looks as follows:

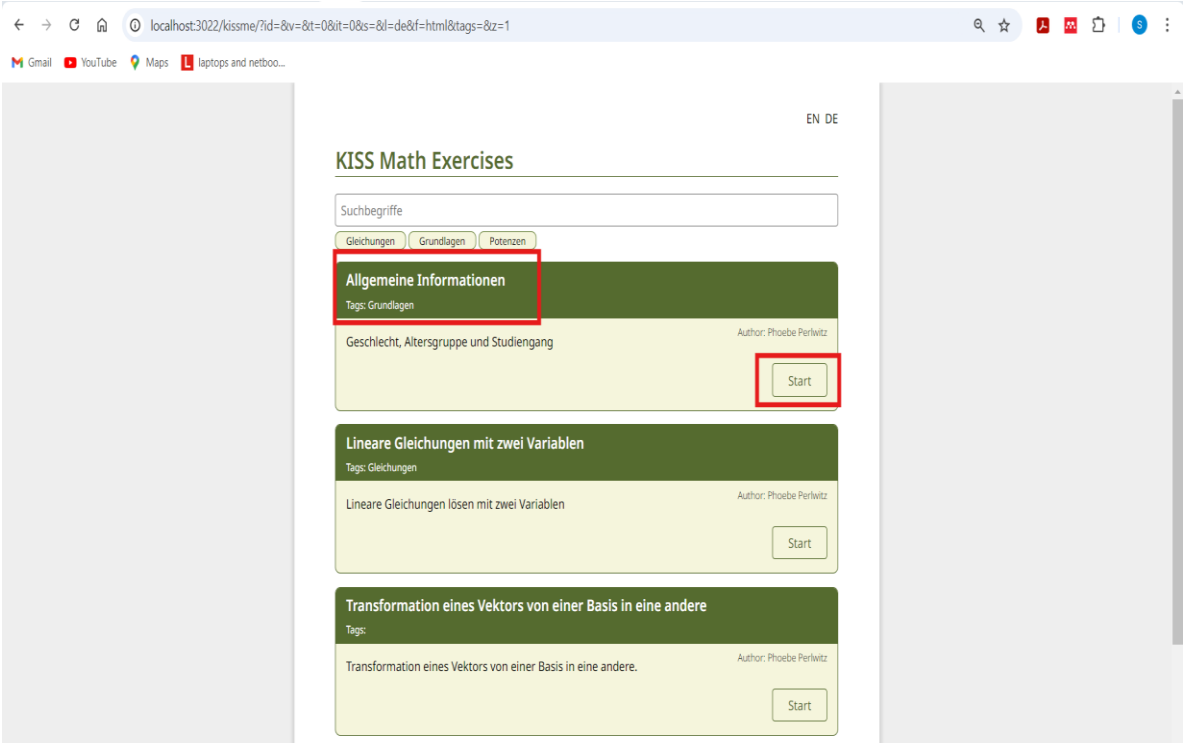


Figure 13:KISS ME starting page

The section with the heading “Allgemeine informationen” which translated to general information will take a student to the page below where one can enter their information upon clicking “start”.

Figure 14: Page requesting general information of students

Here, it can be seen that the "Altersgruppe" (translated as Age group) has been presented with three options namely, 18-20 which mainly consists of fresher men and sophomores, 21-28 which involves third year students, final year students, those with arrears and also those who have opted for a semester breaks and have deferred their graduation and finally 28+ who have presumably started their course late and also those who have not been able to complete their course due to reasons of their own.

Next question is the" studiengang" (translated as study course) which is also provided with three choices and they are as follows:

- Mechanical and Medical engineering
- Informatic science
- Andere/others (which are particularly pointed at non-technical courses such as management and business related)

The reason for these choices is that it is a known fact that the level of mathematics varies from one course to another and students from different domain have to be tested out to see the difference in level of understanding in math student from each domain has based on which students can be grouped into different classes where they will be presented with the appropriate suggestions and materials that corresponds to their pace of learning. For instance, students from Mechanical and Medical engineering background might have a strong hold in mathematics and may probably require advanced level coaching and materials as most of the engineering concepts are based on complex math. On the other hand, for someone who is from a non-STEM (Science Technological, Engineering and

Mathematics) background and are into management related courses, they might not possess a strong background in mathematics because of which beginner and intermediary level topics relevant to their field has to be laid more emphasis.

The final piece of information required form the students is about their gender which can be found under the section “Geschlecht” and is also given with three choices namely: “Mannlich” (Male),” Weiblich” (Female) and “Divers” (others).

2. DESIGN OF THE MATH EXERCISES

Some of the math topics that has been selected for analysing the student’s performance include interest calculation, percent calculation, quadratic equations, linear equations, logarithms, polynomial divisions, vector manipulation and many more and are a part of the semester curriculum. The topics are also presented with various difficulty levels ranging from easy, intermediate and hard. Once the students are done entering their general information, they will be redirected to a page where they can select a math topic and the appropriate difficulty level of their interest. The topic selection page looks as flows:

Figure 15: Page for selection of math topic by student

As it can be seen, The page also provides the user the option to choose between “Gleichungen” (Equations), “Grundlagen” (Basics) and “Potenzen”(Powers) at the top of the page and also each topic shows which category it falls into under the tags section and the author of the content is also displayed on the right.

Now that an overview of the KISS ME webpage design has been given, we will now examine how students interact with the platform while attempting to solve the assigned mathematical tasks.

After a student decides for a particular topic, they should hit the START button which takes them to math questionnaire of that particular topic as below:

Figure 16: Starting an exercise

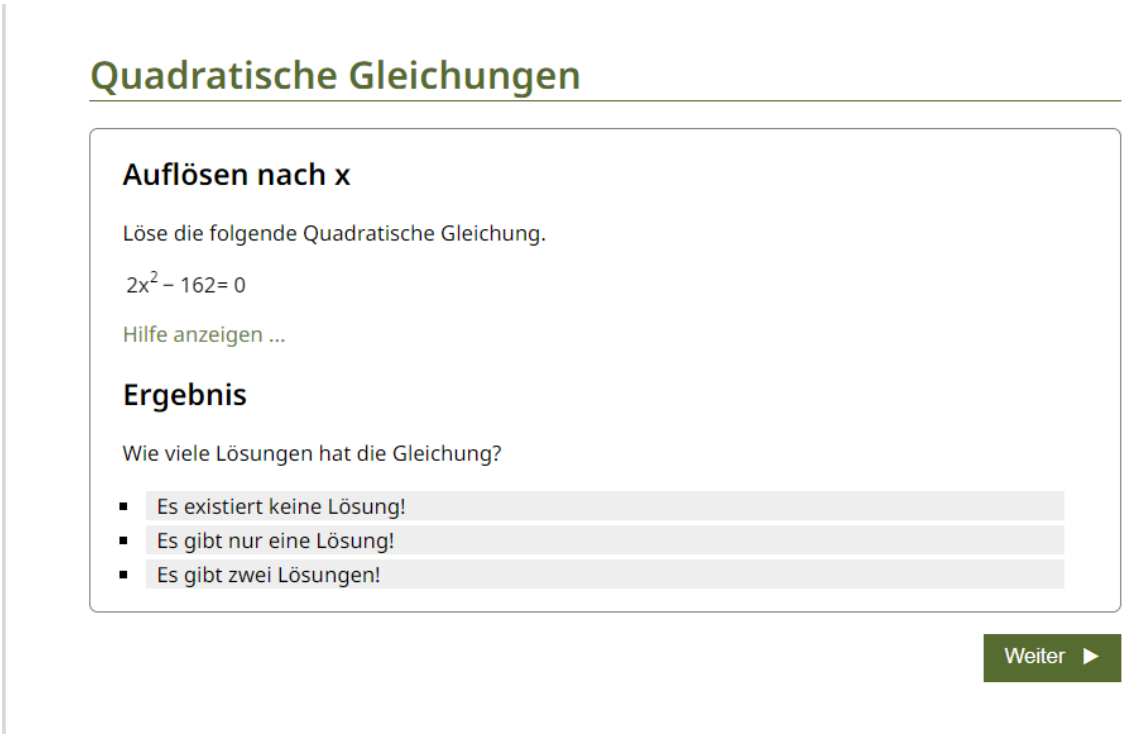


Figure 17: Task page

For the purpose of explanation, the topic Quadratic Gleichungen (Quadratic Equations) has been selected. After hitting the START button, it can be seen that the first problem of the topic has showed up and one can see many possibilities here which are explained below.

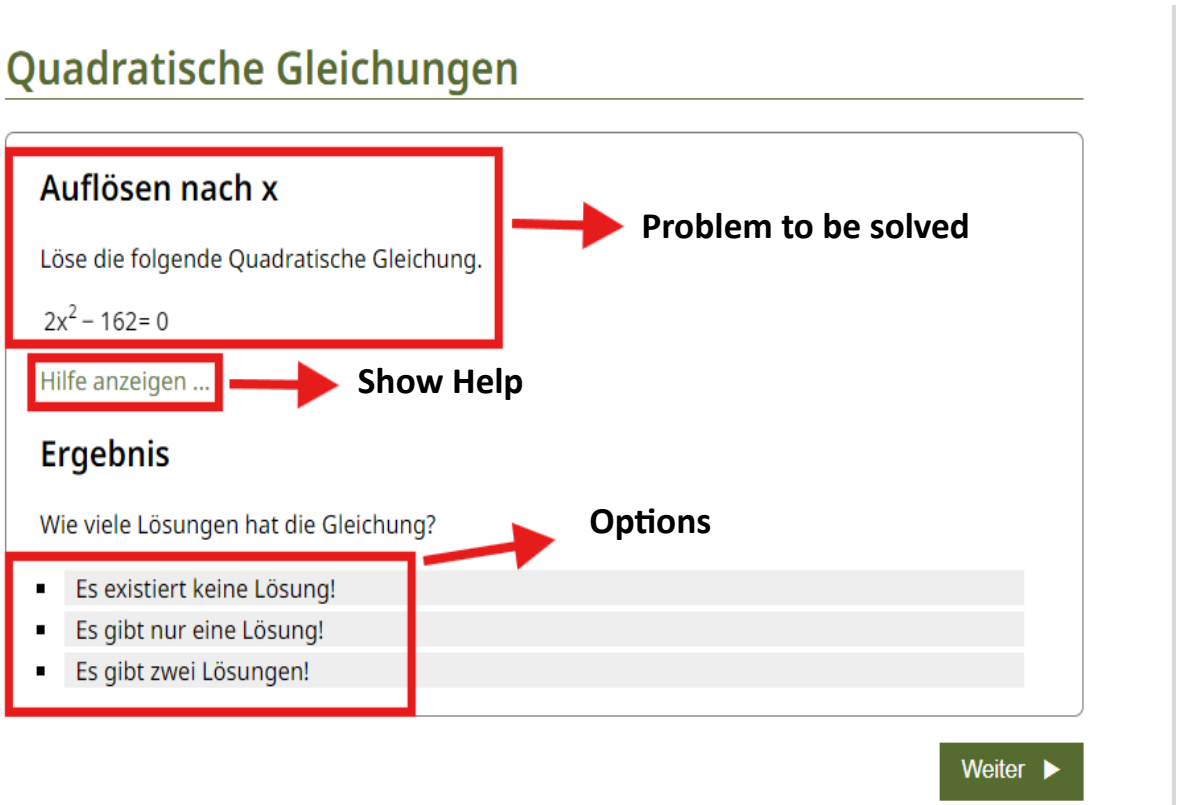
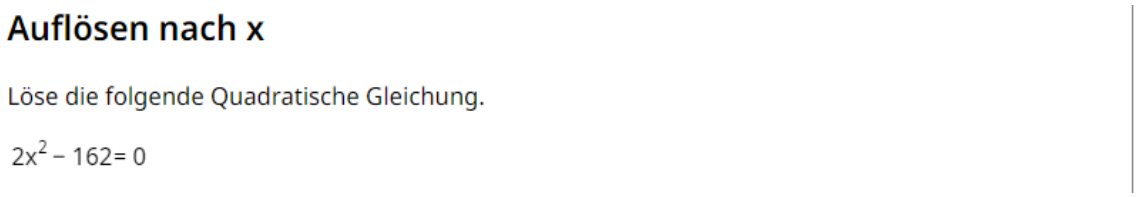


Figure 18: Explanation of the task page

The Task page is generally divided up into three sections, namely "Auflösen" translated as solve, "Hilfe anzeigen" translated as show help and the last section where the students can select their choice of answer. Each of this section has been explained in detail below.

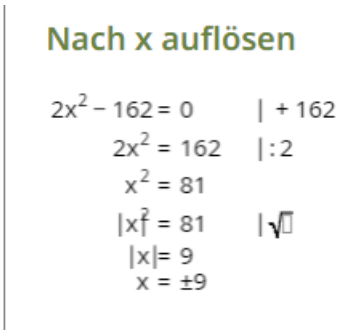
- **TASK SECTION**



This section simply presents the user with the task that needs to be solved and nothing more. Here it is apparent that the task at hand is **$2*x^2 - 162 = 0$**.

- **HELP SECTION**

This section offers help to the users in case the user feels stuck at a question and feel like they need help to move further. The maximum number of help a user can get varies from 1 to 3 depending on the level of hardness of the task to be solved and in the scenario where multiple help requests are made, they ae displayed one after another as shown below.



As the example problem taken was an easy one, the number of helps available is just one. Let's have a look at another example which is slightly more complicated than this one.

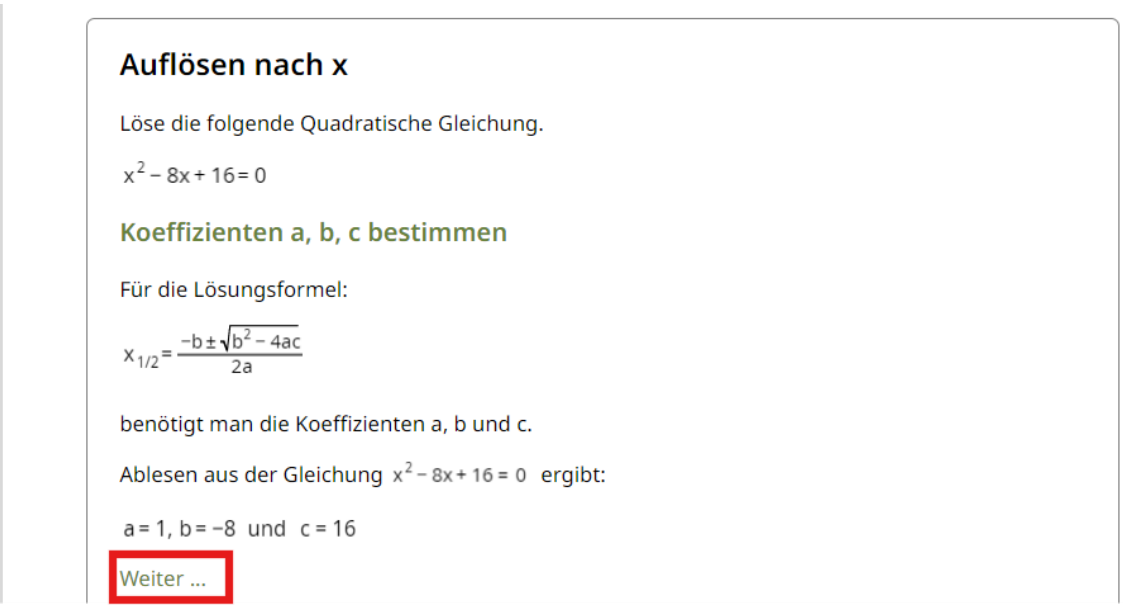


Figure 19: Help No.1

Here it can be seen that first help gives out the coefficients of the equation a, b and c and expects the user to solve the task further. In the event the user is still unable to

solve the task, they can simply click on the “weiter” (further) button which displays the second help which is shown below.

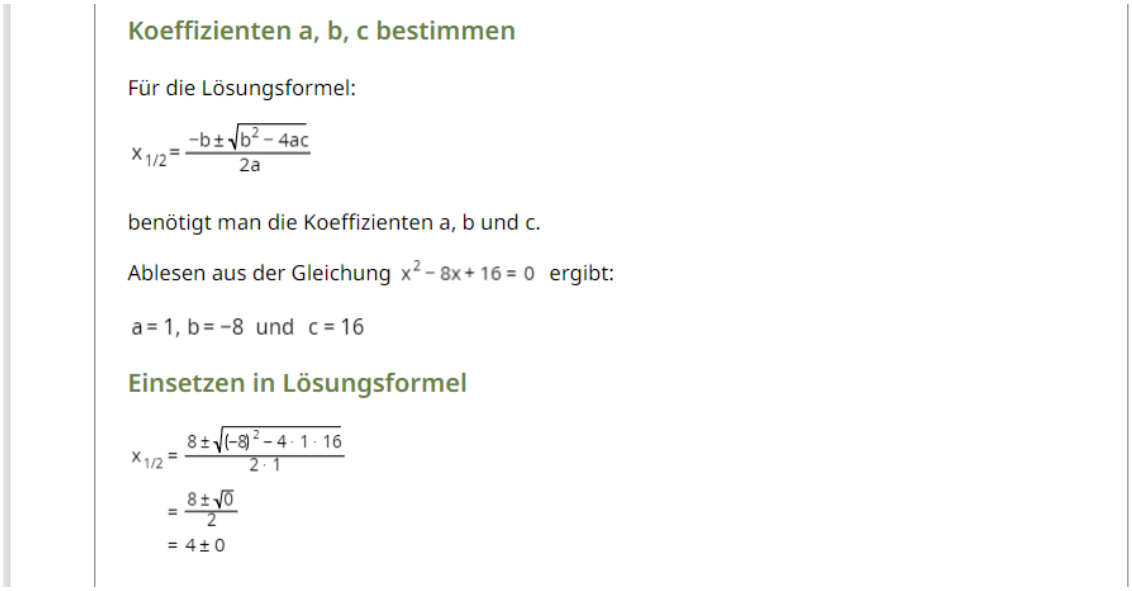
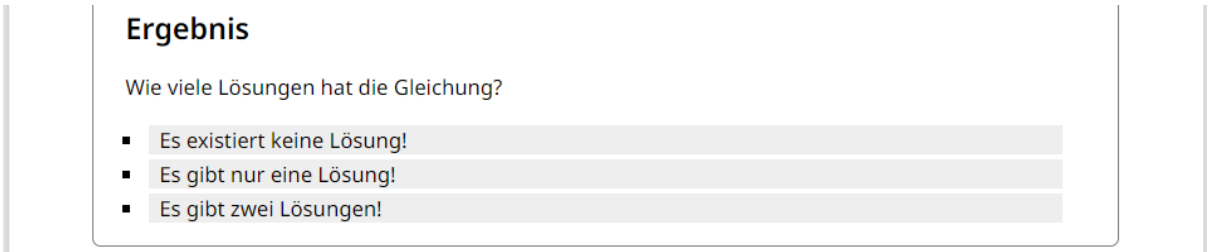


Figure 20: Help No.2

Here, the solution for x is simply given out which is the final answer. One can also notice that each help provides the user with the steps necessary to solve the task. For instance, the first help with the heading “Koeffizienten a, b, c bestimmen” (determine the coefficients) shows the user, how the coefficients of the equation can be figured out. In the next help section “Einsetzen in Lösungsformel” (Insert into the formula) tells the users how the equation can be solved for x using these coefficients. This way, a user also gets a chance to learn while trying to solve the tasks simultaneously.

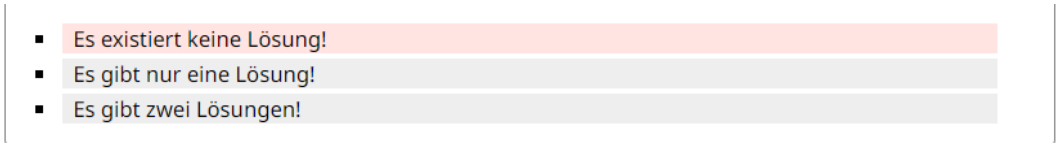
- **OPTIONS**

Finally, this is the part of the webpage where the students can choose their choice of answer and proceed forward.

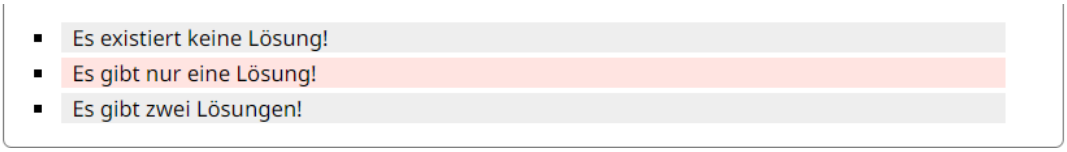


The heading which reads as “Ergebnis” (Results) presents itself with three options:

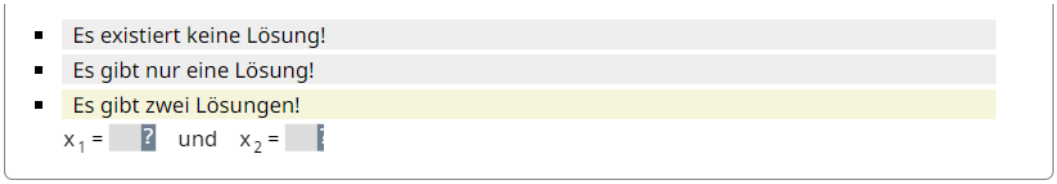
The first one being “There is no solution” which upon clicking gets highlighted.



The second one reads “There is only one solution”.



The third option which reads “There are two solutions” which when clicked presents the user with two empty boxes.



These boxes are the place where they enter their answers. It can also be seen that there exists a question mark symbol next to the boxes which directly gives the user with the answers but will be registered as not solved in the log file.

Once the user has entered their answer in the boxes or are done selecting the right answer they can simply click in the “weiter” button which takes them to the next question.

Figure 21: Entering the answers in the box

3. DATA LOGGING

All the general information and the answers entered by the students will be saved in a log file in the background. An instance where a user enters his/her general information and the data getting registered in the log file is shown below:

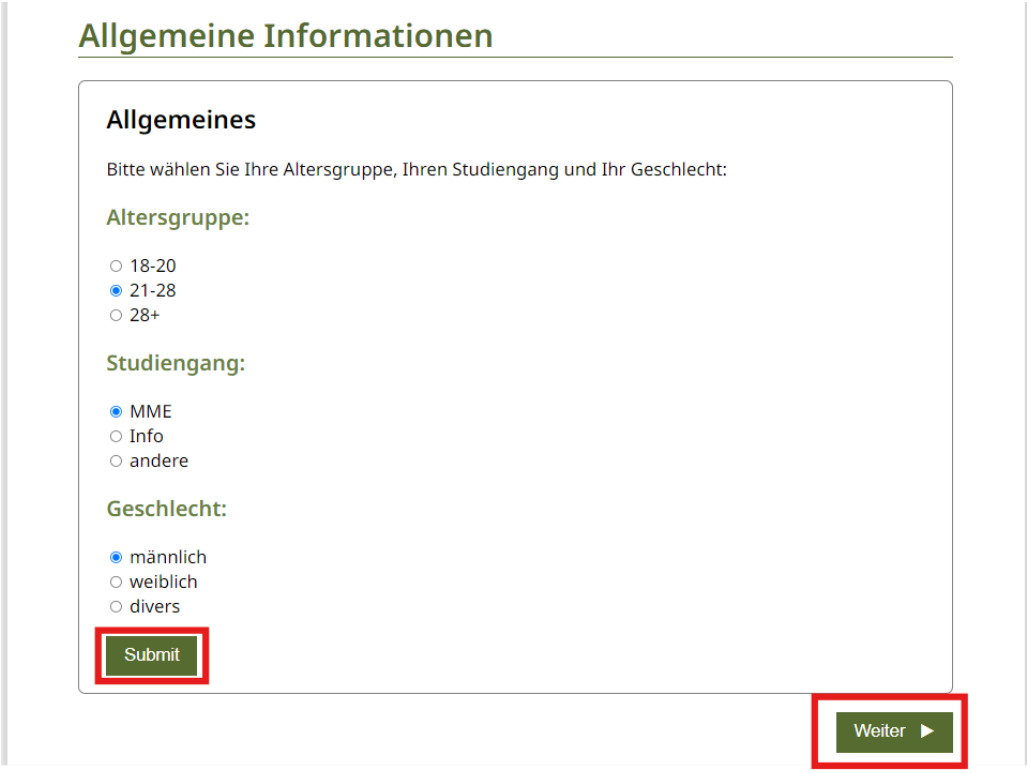


Figure 22: Submitting the general information by students

Here, it can be seen that the user has selected the following parameters:

- Altesgruppe (Age group): 21-28
- Studiengang (Course of study): MME
- Geschlecht (Sex): Männlich (Male)

All this information will be stored in a log file in the form of numbers as shown below:

Figure 23: data logging in the background

It is being displayed in the fourth and the seventh line with the following information which are explained below:

1. **“Client”: “K35quga2ker”**
   This is the information generated through the combination of client and server addresses and is unique for each client.

2. **“id”: “courseSelectionExercise_de”**
   This is the identity of the questionnaire being selected and it’s the same every time a student has to enter this information.

3. **“Seed”: “0”**
   This is the random seed number that is used to initialize the process

4. **“type”: “START” (fourth line) and “type”: “FINISH” (seventh line)**
   This information tells us the type of process that is in progress. START indicates that the questionnaire has been initialised and FINISH indicates that the questionnaire has been submitted. Also, there is another type INPUT which gives the information about each value being inputted by the user.

5. **“Events”:**
   [{"time":1729415146198,"event":"INPUT","value":{"ageGroup":"1"}},{"time":1729415147927,"event":"INPUT","value":{"course":"0"}},{"time":1729415149204,"event":"INPUT","value":{"gender":"0"}}]

This line (7th line) gives the information about all the values inputted by the client along with corresponding time that particular input took place in Unix epoch format. The information of our interest are as follows:

- "ageGroup": "1"
- "Course": "0"
- "gender": "0"

It is to be noted that the optional values showed on the general information questionnaire page are coded into 0, 1 and 2 in the log file for storage purposes.

6. **"_meta": {"time":"2024-10-20T09:05:40.162Z"} (START of event) and "_meta":{"time":"2024-10-20T09:06:02.966Z"} (FINISH of event)**
   Finally, the time and date stamp of each event is given at the end of the corresponding line.

DATA EXTRACTION AND CLEANING

After the completion of the students' participation in the questionnaire, the log file stored in the system storage will be fed to the analysis algorithm where the data which is in raw form will be subjected to extraction and thorough cleaning and be converted into a data frame that is suitable for carrying out further analysis. The programming software used for this purpose is python and the method implemented will be discussed below:

1. Reading the log file obtained

As the log file being read will be easier to read in JSON format, 'JSON' library in python has been used for opening and reading the log file in its raw format:

```
1. import json
2. with open('log0', 'r', encoding = 'utf-8') as f:
3.     data = [json.loads(line) for line in f]
```

The above code snippet opens the log file 'log0' with the UTF-8 encoding and 'r' specifies that the file opened will be read. The third line iterates over each line in the file 'f' and the new result will be appended to the list. The 'json.loads()' function from the JSON module helps in the conversion of a JSON formatted string into a corresponding python object like a dictionary [65].

The above code snippet results in the following output:

```
OUTPUT:

 1. [{'userid': 'test',
```

```
 2.   '_meta': {'time': '2024-06-04T08:28:18.752Z'},
 3.   'data': [{'id': 'ex1-1', 'val': 0, 'sol': 2, 'pts': 5, 'Pts': 5},
 4.    {'id': 'ex1-2', 'val': 1, 'sol': 3, 'pts': 0, 'Pts': 5}]},
 5.  {'userid': 'test',
 6.   '_meta': {'time': '2024-06-04T08:28:20.757Z'},
 7.   'data': [{'id': 'ex1-1', 'val': 1, 'sol': 2, 'pts': 5, 'Pts': 5},
 8.    {'id': 'ex1-2', 'val': 1, 'sol': 3, 'pts': 0, 'Pts': 5}]},
 9.  {'userid': 'test',
10.   '_meta': {'time': '2024-06-04T08:28:22.760Z'},
11.   'data': [{'id': 'ex1-1', 'val': 2, 'sol': 2, 'pts': 5, 'Pts': 5},
12.    {'id': 'ex1-2', 'val': 1, 'sol': 3, 'pts': 0, 'Pts': 5}]},
13.  {'client': '7xp9g5fo78v',
14.   'id': 'surveystats',
15.   'data': {'ageGroup': '3'},
16.   '_meta': {'time': '2024-06-04T08:28:32.077Z'}},
17.  …
```

2. Converting the above JSON data into a usable data frame

Now that the log file has been opened and read, it needs to be converted into a form where the data is more readable and is also extractable for the analysis purpose. The 'pandas' library in python is a useful tool in building data frames which can be easily manipulated either by extracting a part of the framer, filling Nan values or removing the undesired values. [66]

```
1. import pandas as pd
2. df = pd.DataFrame(data)
3. df = df.sort_values(by=['client', 'userid'])
4. df = df.reset_index()
```

Initially 'pandas' is imported as 'pd' in short and the JSON data which was obtained in the previous step is fed to the 'Dataframe' function of the pandas which converts it into data frame table as shown below and the rows are sorted by 'client' first and 'user_id' next as this makes the extraction of simple information easier. The following data frame will be created as a result of the above snippet.

userid	_meta	data	client	id	variant	seed	type	events	result
NaN	{'time': '2024-06-04T12:09:26.070Z'}	NaN	21ntq5ug7jf	powerrules_de		0	START	NaN	Nal
NaN	{'time': '2024-06-04T12:09:36.809Z'}	NaN	21ntq5ug7jf	powerrules_de		0	INPUT	NaN	Nal
NaN	{'time': '2024-06-04T12:09:39.175Z'}	NaN	21ntq5ug7jf	powerrules_de		4080997195132541 3	START	NaN	Nal
NaN	{'time': '2024-06-04T12:09:39.175Z'}	NaN	21ntq5ug7jf	powerrules_de		0	FINISH	[{'time': 1717502977259, 'event': 'INPUT', 'va...	{'data': {'resB1 {'value': '7', 'parent 'm..
NaN	{'time': '2024-06-04T12:09:45.545Z'}	NaN	21ntq5ug7jf	powerrules_de		4080997195132541 3	INPUT	NaN	Nal
NaN	{'time': '2024-06-04T12:10:19.234Z'}	NaN	21ntq5ug7jf	powerrules_de		4080997195132541 3	HELP	NaN	Nal
NaN	{'time':	NaN	21ntq5ug7jf	powerrules_de		1371028824182114 6	START	NaN	Nal

Figure 24: Data frame Created from the logged data

3. ADDING HELP COUNT AND TIME TAKEN COLUMNS

From the output above, it can be seen that the column 'events' contain most of the useful information of which 'time taken (in seconds)' and 'help_count' piques our interest the most as these two parameters helps in interpreting the strength and weakness of the participants.

- **HELP COUNT COLUMN**

Whenever a student requests for help, it gets registered in the 'event' key of the 'events' column. An example where a user has accessed for help twice for the same task has been shown below:

```
1. 'events': [{'time': 1717503053602,
2. 'event': 'HELP', 'value': {'resB1': '#fill'}},
3. {'time': 1717503057905, 'event': 'HELP', 'value': {'resN1': '#fill'}}],
```

By knowing this information, a separate column has been created particularly to keep a record of number of helps count for each task attempted.

```
 1. def extract_help_count(events):
 2.     help_count = 0
 3.     for event in events:
 4.         if event['event'] == 'HELP':
 5.             value = event.get('value', 0)
 6.             if isinstance(value, (int, float)):
 7.                 help_count = int(value) + 1
 8.             elif isinstance(value, str) and value.isdigit
 9.                 help_count = int(value) + 1
10.     return help_count
11.
12. df['help_count'] = df['events'].apply(lambda x: extract_help_count(x) if isinstance(x, list)
else 0)
```

The function 'extract_help_count' above code snippet looks for the word 'HELP' in the 'event' key of the 'events' column row by row and returns the value if it is in int, float or digit form and gets incremented if multiple instances of 'HELP' is found in the same 'event' dictionary. The last line of the snippet adds a new column 'help_count' which incorporates the function defined above and inserts the value returned by the function to the corresponding row.

variant	seed	type	events	results	help_count	time_taken_seconds	ageGroup_mapped	course_mapped	gender_mapped	is_Solved
	0	START	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN	None
	40809971951325413	START	{}	{}	0	NaN	NaN	NaN	NaN	None

4179301007734286	START	{}	{}	0	NaN	NaN	NaN	NaN	None
4179301007734286	HELP	{}	{}	0	NaN	NaN	NaN	NaN	None
0	START	{}	{}	0	NaN	NaN	NaN	NaN	None
0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN	None
0	HELP	{}	{}	0	NaN	NaN	NaN	NaN	None
0	FINISH	[{'time': 1717503278039, 'event': 'INPUT', 'va...	{'data': {'x': {'value': '0,5', 'parent': 'me-...	2	104.054	NaN	NaN	NaN	False

Figure 25: Help count column added

- **TIME TAKEN IN SECONDS**

This is followed by 'time taken in seconds' column which tells us how much time a particular user has taken in solving a particular task. Once again, the 'events' column will be made use of to extract the time information.

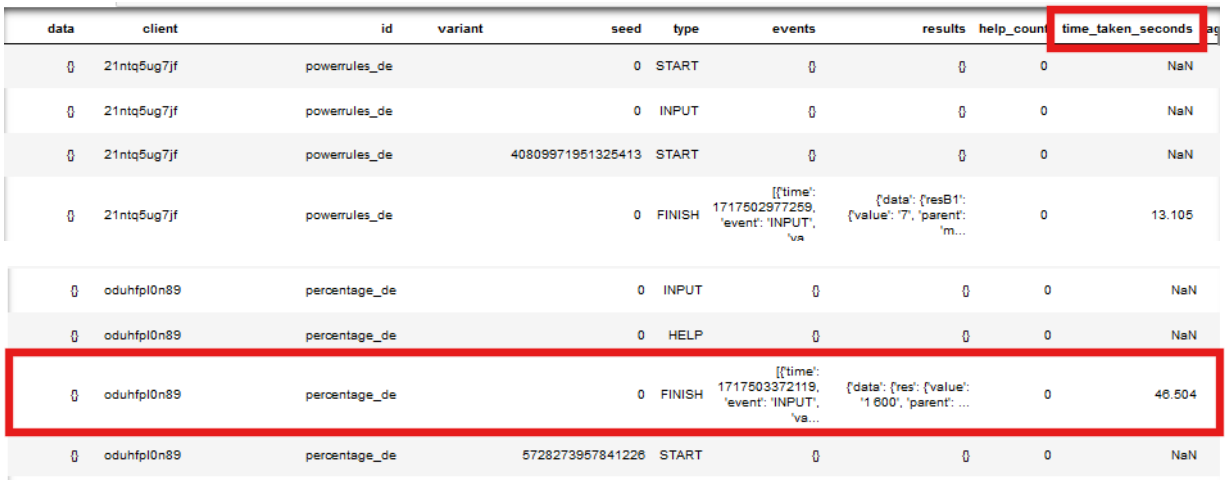
```
{'client': 'oduhfpl0n89',
  'id': 'percentage_de',
  'variant': '',
  'seed': '0',
  'type': 'START',
  '_meta': {'time': '2024-06-04T12:15:29.762Z'}},
……
……
{'client': 'oduhfpl0n89',
  'id': 'percentage_de',
  'variant': '',
  'seed': '0',
  'type': 'FINISH',
  'events': [{'time': 1717503372119, 'event': 'INPUT', 'value': {'res': '1'}},
   {   {'time': 1717503374758, 'event': 'HELP', 'value': {'res': '#fill'}}],
  'results': {'data': {'res': {'value': '1\u2006600',}},
   'isSolved': False},
  '_meta': {'time': '2024-06-04T12:16:16.266Z'}}

```

Here, the example of the client 'oduhfpl0n89' has been taken where he/she has attempted to solve a percentage task. It can be noted that there is a time stamp for both START and FINISH events in another column called '_meta'. The difference in time between these two events gives us the time consumed in solving the task by that user. As there are multiple events getting logged simultaneously between the START and FINISH of this event under consideration and all those events too have time stamps, it is imperative to make sure that the time stamp of a FINISH event is subtracted from the correct corresponding START event. For this purpose, the SEED value in each column has been utilised which will be unique for every new task attempted by the user. In this case we have a seed value of 0 which is the same in both the events. The following code snippet is implemented in extracting the time difference between the FINISH and the START events of the same SEED value.

```
from datetime import datetime

#Calculating the time difference
def calculate_time_difference(df):
    start_times = {}
    time_differences = []

    for index, row in df.iterrows():
        if row['type'] == 'START':
            start_times[row['seed']] = datetime.strptime(row['_meta']['time'], '%Y-%m-%dT%H:%M:%S.%fZ')
        elif row['type'] == 'FINISH':
            start_time = start_times.get(row['seed'])
            if start_time is not None:
                finish_time = datetime.strptime(row['_meta']['time'], '%Y-%m-%dT%H:%M:%S.%fZ')
                time_difference = (finish_time - start_time).total_seconds()
                time_differences.append(time_difference)
            else:

                time_differences.append(None)


    num_pad = len(df) - len(time_differences)
    time_differences.extend([None] * num_pad)

    return time_differences

time_diffs = calculate_time_difference(df)

#Assign the time difference to correct rows
finish_counter = 0

for index, time_diff in enumerate(time_diffs):
    if df.loc[index, 'type'] == 'FINISH':
        df.loc[index, 'time_taken_seconds'] = time_diffs[finish_counter]
        finish_counter += 1

```

The snippet can be divided into two parts. One for calculating the time difference and the second one for assigning the calculated difference to the correct rows where the event type corresponds to FINISH. ‘datetime’ library has been utilised for the purpose of manipulating and extracting time.

The output with another column ‘time_taken_seconds’ will be added right next to help_count and will be having the values of time difference between START and FINISH at locations where event type is FINISH. In the instance considered previously of the client oduhfpl0n89 for the math topic percentage, the start time was 12:15:29:762 and the FINSIH time was 12:16:16:266 and the difference comes out to be 46.504 seconds which can be seen being inserted into the new column created for that corresponding user when the event type is FINSIH.



data	client	id	variant	seed	type	events	results	help_count	time_taken_seconds	ag
{}	21ntq5ug7jf	powerrules_de		0	START	{}	{}	0	NaN	
{}	21ntq5ug7jf	powerrules_de		0	INPUT	{}	{}	0	NaN	
{}	21ntq5ug7jf	powerrules_de		40809971951325413	START	{}	{}	0	NaN	
{}	21ntq5ug7jf	powerrules_de		0	FINISH	[{'time': 1717502977259, 'event': 'INPUT', 'va	{'data': {'resB1': {'value': '7', 'parent': 'm...	0	13.105	
{}	oduhfpl0n89	percentage_de		0	INPUT	{}	{}	0	NaN	
{}	oduhfpl0n89	percentage_de		0	HELP	{}	{}	0	NaN	
{}	oduhfpl0n89	percentage_de		0	FINISH	[{'time': 1717503372119, 'event': 'INPUT', 'va...	{'data': {'res': {'value': '1 600', 'parent': ...	0	46.504	
{}	oduhfpl0n89	percentage_de		5728273957841226	START	{}	{}	0	NaN	

Figure 26: Time taken in seconds column added

4. MAPPING THE VALUES OF AGE GROUP, GENDER AND COURSE OF STUDY

As discussed earlier, the age group, gender and course of study information entered by the students will be stored as 0, 1 or 2 depending on the users input in the log files. These values need to be made readable by mapping them into their original values say, '18-20', 'Weiblich' and 'MME' if the log entry is 0, 1 and 0.

```
#Extraction of age_group, course and gender values
1. df['ageGroup'] = [row[0]['value']['ageGroup'] if isinstance(row, list) and len(row)>=3 and
row and isinstance(row[0], dict) and 'value' in row[0] and isinstance(row[0]['value'], dict) and
'ageGroup' in row[0]['value'] else None for row in df['events']]

2. df['course'] = [row[1]['value']['course'] if isinstance(row, list) and row and len(row)>=3
and isinstance(row[1], dict) and 'value' in row[1] and isinstance(row[1]['value'], dict) and
'course' in row[1]['value'] else None for row in df['events']]

3. df['gender'] = [row[2]['value']['gender'] if isinstance(row, list) and row and len(row)>=3
and isinstance(row[2], dict) and 'value' in row[2] and isinstance(row[2]['value'], dict) and
'gender' in row[2]['value'] else None for row in df['events']]

#Mapping of 0,1 and 2 into actual values
4. df['ageGroup_mapped'] = df['ageGroup'].map({"0": "18-20", "1": "21-28", "2":"28+"})

5. df['course_mapped'] = df['course'].map({"0": "MME", "1": "info", "2":"andere"})

6. df['gender_mapped'] = df['gender'].map({"0": "manlich", "1": "weiblich", "2":"divers"})

7. df_mapped = df.drop(["ageGroup", "course", "gender"], axis = 1)
```

In the first section, three new columns are created with the names 'ageGroup', 'course' and 'gender' which extracts the value that corresponds to the column name from the 'events' column in the data frame if certain conditions such as:

- **isinstance(row, list)** → the row being searched for a value is a list,

- **len(row)>3** → row has atleast three elements (as there are three parameters agegroup, gender and course), row is not empty,
- **row** → row is not empty
- **isinstance(row[0], dict)** →ensures that first element in a row is dictionary
- **'value'in row[0]** → ensures that the value in first dictionary is not empty
- **isinstance(row[0]['value'], dict)** → makes sure that the value of the first element is also a dictionary
- **'ageGroup' in row[0]['value'** → finally, making sure that the age group exists in nested value dictionary.

This process is done for the other two parameters as well (gender and course of study) and lastly, these values are mapped according to the information available on the website asking the general information of user in three separate columns named ' ageGroup_mapped', 'gender_mapped' and 'course_mapped' in the second section of the code.

Once the values (0, 1 2) are mapped into their actual values, the column created for the extraction of 0, 1 and 2 ('ageGroup', 'course' and 'gender') will be dropped using the pandas drop function. The output obtained from the above snippet is as follows:

id	variant	seed	type	events	results	help_count	time_taken_seconds	ageGroup_mapped	course_mapped	gender_mapped
rules_de		0	START	{}	{}	0	NaN	NaN	NaN	NaN
rules_de		0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN
rules_de		40809971951325413	START	{}	{}	0	NaN	NaN	NaN	NaN
rules_de		0	START	{}	{}	0	NaN	NaN	NaN	NaN
cEqs_de	intermediate	0	START	{}	{}	0	NaN	NaN	NaN	NaN
rcise_de		0	START	{}	{}	0	NaN	NaN	NaN	NaN
rcise_de		0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN
rcise_de		0	FINISH	[{'time': 1717502899648, 'event': 'INPUT', 'va...	{'data': {}, 'isSolved': True}	0	16.259	18-20	andere	weiblich

Figure 27: Age group, gender and course columns added

We can see the information an user being printed as '18-20' in the 'ageGroup_mapped' column, 'andere' in the 'course_mapped' column and 'weiblich' in the 'gender_mapped' column.

5. DETERMINING WHETHER THE STUDENT HAS SOLVED THE TASK OR NOT

The final piece of information required for the analysis is knowing whether or not the student was able to solve the task correctly and this is again embedded in the 'event' dictionary of 'events' column under the key-value pair 'isSolved': True/False.

Taking the example of the client ‘oduhfpl0n89’ for the math topic percentage, it can be seen that the user was not able to solve the problem correctly (indicated by “FALSE” in ‘isSolved’ value). This piece of information will be extracted using the code snippet shown below.

```
  'results': {'data': {'res': {'value': '1\u2006600',
     'parent': 'me-solution',
     'color': 'w'}},
   'isSolved': False},
```

```
def is_solved_extracted(rows):
        if isinstance(rows, dict) and 'isSolved' in rows:
            return rows['isSolved']
        else:
            None

df['is_Solved'] = df['results'].apply(is_solved_extracted)
```

The function ‘is_solved_extracted(rows)’ checks whether the row is an instance of dictionary and also there exists a key in that dictionary called ‘isSolved’ which when true, the value of ‘isSolved’ will be returned. This function is applied on the ‘results’ column of the whole data frame using the ‘apply’ method which creates a new column called ‘is_solved’. The resulting data frame is shown below.

variant	seed	type	events	results	help_count	time_taken_seconds	ageGroup_mapped	course_mapped	gender_mapped	is_Solved
	0	START	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN	None
	40809971951325413	START	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	FINISH	[{'time': 1717502977259, 'event': 'INPUT', 'va...	{'data': {'resB1': {'value': '7', 'parent': 'm...	0	13.105	NaN	NaN	NaN	True
	0	START	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	INPUT	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	HELP	{}	{}	0	NaN	NaN	NaN	NaN	None
	0	FINISH	[{'time': 1717503372119, 'event': 'INPUT', 'va...	{'data': {'res': {'value': '1 600', 'parent': ...	0	46.504	NaN	NaN	NaN	False

Figure 28: "Is_solved" column added

With all these pieces of information, we can now proceed to the next stage which is the general overview of the data that is obtained and the interpretation of the results as well.

RESULTS AND DISCUSSION

FIRST LOOK OF THE DATA

After putting the log file through thorough data cleaning and data extraction steps which are mentioned in the previous section, it was discovered that eighteen students had taken part in the math questionnaire conducted out of which eight of them were females, two of them were males and the remaining eight gave no information about their gender. Regarding the age group, five of them belonged to 18-20 group, another five were from 21-28 group and once again, the remaining eight made no information available about their age-group as well. With respect to the field of study, all of the students had opted for 'andere' (others) which rendered this parameter useless for the purpose of analysis.

It can be noted that the females have outnumbered the males in terms of participation leaving out of the consideration a significant portion of the participants who haven't disclosed their genders. In regards to the age group, the division of data has been balanced but again the age group of majorities of the population still remains a mystery. The third parameter, field of study, where all of the participants have opted for 'andere' limits the ability to get an understanding about the participants academic background which otherwise could have proved to be a valuable piece of information in understanding the differences in responses from participants of different background (STEM or non-STEM).

A pictorial representation of the student's participation is shown below in the form of pie charts:

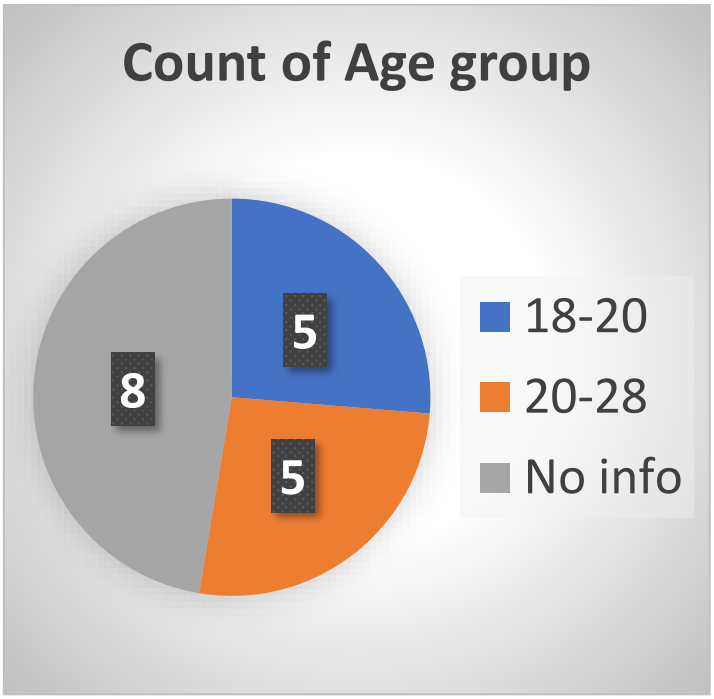


Figure 29: Count of participants from different age groups

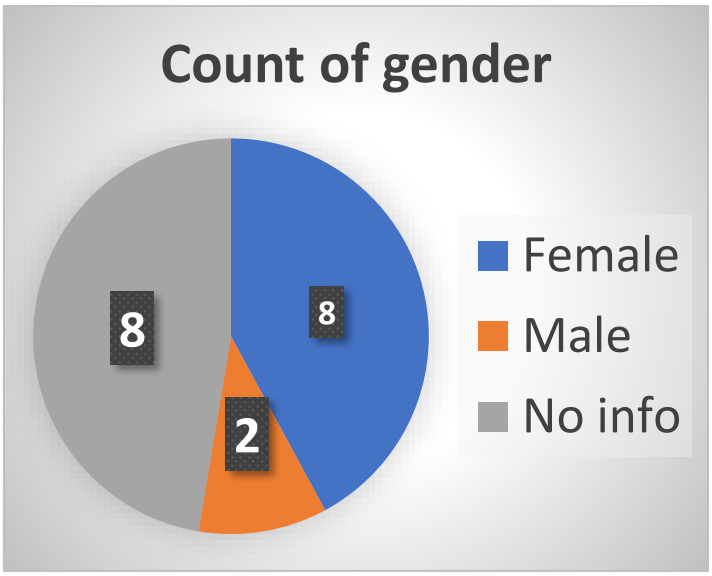


Figure 30: Count of participants from different genders

The clients whose information were not disclosed were discovered with the following code:

```
1. df_mapped['client'] = df_mapped['client'].astype(str)
2.
3. missing_clients = df_mapped.groupby('client').filter(lambda x: x[['ageGroup_mapped',
'course_mapped', 'gender_mapped']].isnull().all().any() or (x[['ageGroup_mapped',
'course_mapped', 'gender_mapped']] == '').all().all())
```

```
4. unique_missing_clients = missing_clients['client'].unique()
```

This gave the following output of eight clients whose information were not revealed:

Clients with missing information:

```
1. ['21ntq5ug7jf' '45ep209qpu9' '7xp9g5fo78v' '999yyaei36a' 'fjbwxozqb9o'
2.  'me53k42emh9' 'obajjfnax0r' 'xmvrm4dje6']
```

From the data available for analysis, it can be said that while this data provides insights about the participants characteristics to certain extent, the clients with missing information and the singular response regarding the field of study (‘andere’) poses a limitation on the conclusions that can be drawn from the analysis and in answering the hypotheses. Moreover, the limited data size truncates the possibility of trying out many of the algorithms that require reasonable size of data to be tested on such as unsupervised algorithms.

OVERVIEW OF THE PARTICIPATION WITH DESCRIPTIVE STATS

In this section, a few basic questions such as how many questions were answered by each age group and gender, average time and help taken by each age group and gender will be answered to get a general idea about the standpoint of both the genders and age group which later serves as a fundamental block in making the topic suggestions for each group.

'Matplotlib' has been extensively used for the purpose of plotting in this section.

WHO ATTEMPTED THE MOST NUMBER OF TASKS?

GENDER

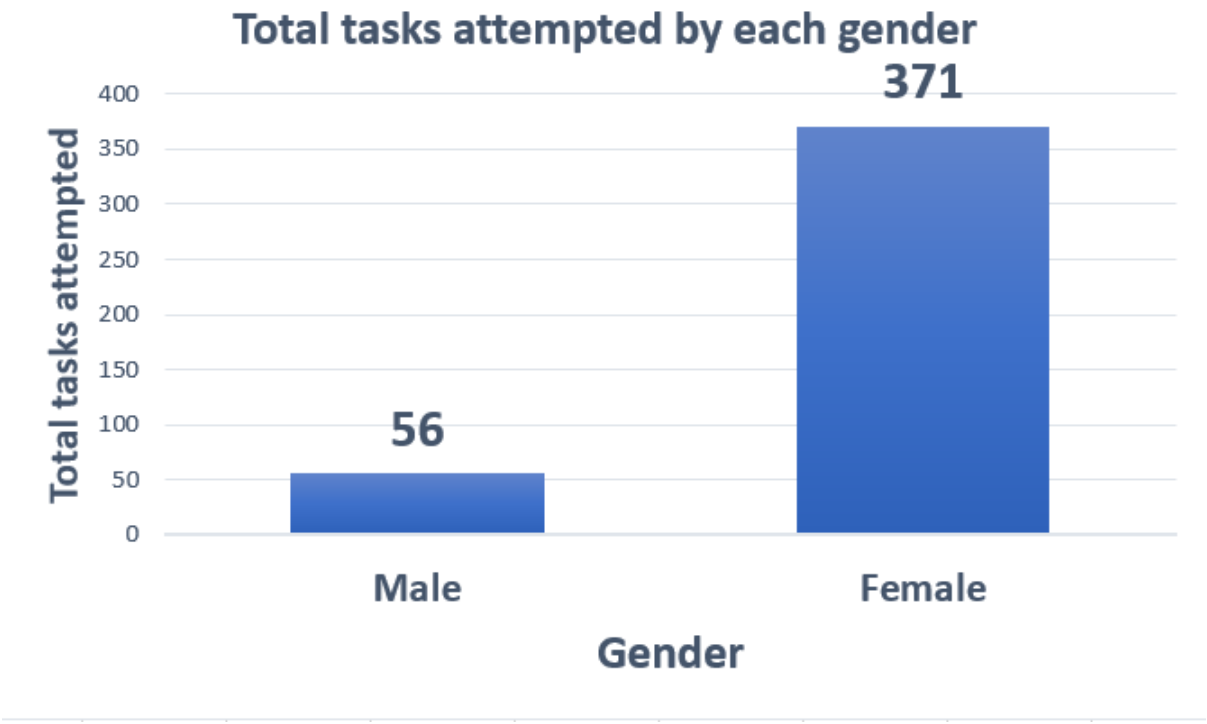


Figure 31: Total number of tasks attempted by both the genders

A total of 56 tasks were attempted by the two male pupils while a grand total of 371 tasks were attempted by the eight female folks. Despite the difference in number of participants in both genders, the average number of tasks attempted per participant reveals an interesting trend. On average, each male participant completed 23 tasks, while each female participant completed an average of 46.3 tasks.

This disparity in task attempts indicates that, on average, female participants engaged more actively with the tasks than their male counterparts, nearly doubling the average number of

attempts per individual. Even though the higher total task count by the female folks can be explained by the larger number of female participants, the average task count points out a higher level of engagement and effort from the female group.

As a conclusion, there is a clear difference in the level of task engagement between the two genders. Although this result provides a useful insight into the participation patterns of both the genders, there can be several factors that needs to be further analysed as to understand what might be the motivation behind their engagement levels. Factors such as interest in subject (on a scale of 1 to 10), external influences and their pre-knowledge in the subject can be taken into consideration in future analysis.

AGE GROUP

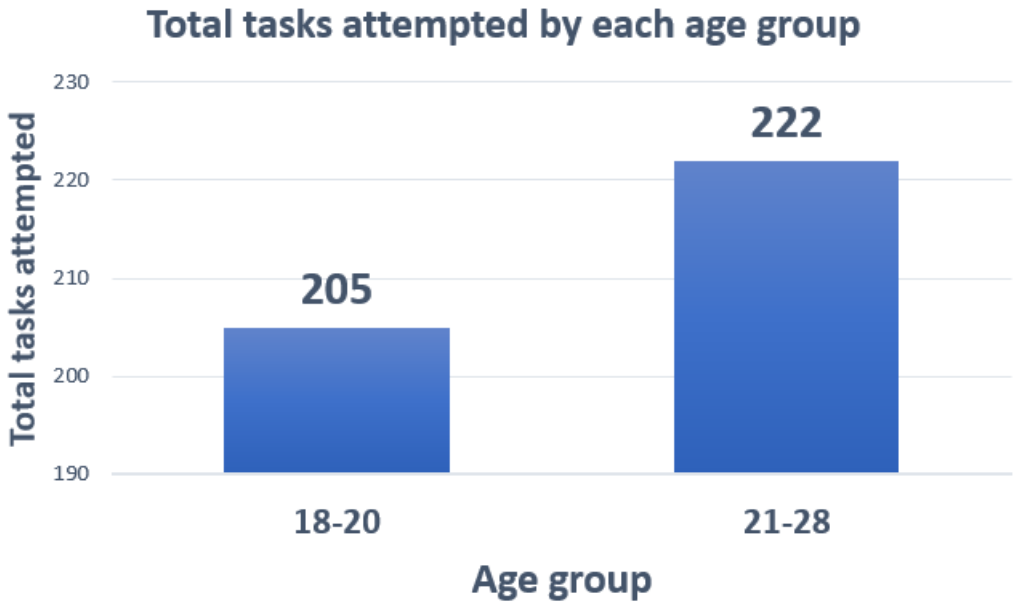


Figure 32: Total tasks attempted by both the age groups

In terms of age group, the difference in number of tasks attempted by both the age groups was trivial in comparison to that of gender. The five students of 18-20 age group attempted a total of 205 tasks while the remaining five from 21-28 age group attempted a total of 222 tasks. This suggests that, on average, the younger group completed 41 tasks per student, whereas the older group completed 44.4 tasks per student.

Though the disparity between the two age groups is minimal, the slight edge in total tasks attempted by 21-28 age group over the younger age group points to a higher level of commitment among those who are most probably in the third year and final year of their graduation compared to those among the fresher men and sophomores.

This pattern tells us that there is a possibility that the students, being further along into their academic journey may have a slightly higher level of engagement towards their studies and that can be linked to their greater familiarity with the topics encountered. On the other hand, those in their 18-20 age group who are most likely to be fresher men and sophomores (first and second years) may still be getting accustomed to the academic environment and also the likelihood of lesser experience in handling the course work.

Nevertheless, the small disparity in number of tasks attempted by both age groups reveals a similar level of overall commitment which was not the case with gender-based comparison.

This may lead to the presumption of the answer to our hypotheses that age may not be significant factor in predicting students' performance in mathematics while gender can be. This will be further probed into in the later sections with the help of ML algorithms.

WHICH GENDER TOOK MORE TIME AND HELP TO SOLVE THE TASKS CORRECTLY?

Here, the point that the meantime and mean help count taken to solve the tasks CORRECTLY has to be stressed because time and help taken when a task has not been solved doesn't contribute properly to the analysis being conducted and only results in a skewed interpretation.

AVERAGE TIME TAKEN

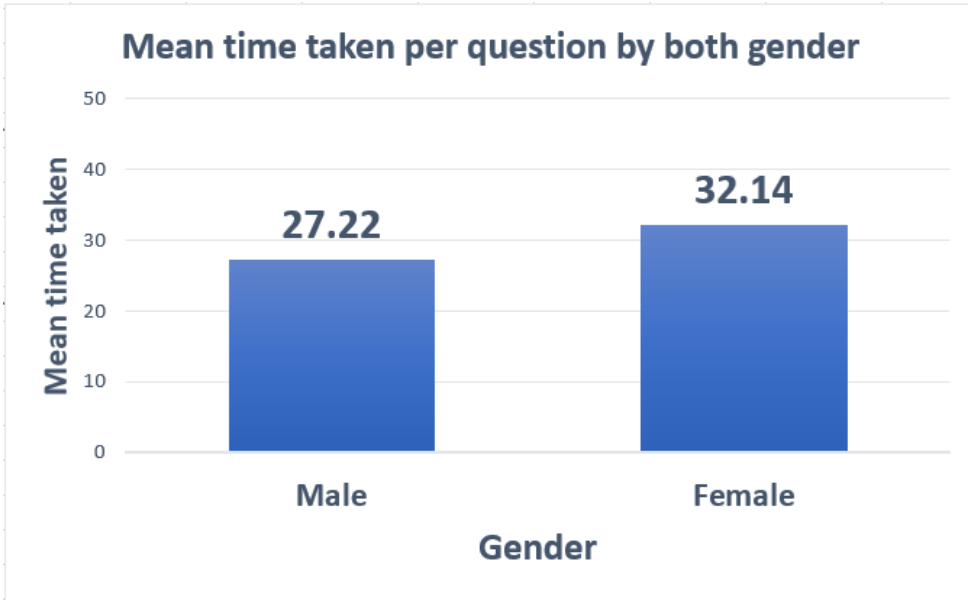


Figure 33: Mean time taken to solve a task correctly by both genders

The male folks on an average took around 27.22 seconds to solve the attempted tasks correctly while the time required by the female folks was higher by approximately around 5 seconds with an average of 32.14 seconds. This difference in average completion time highlights the quicker problem-solving ability among the male folks. But due to the imbalance in the existing data (8 females and 2 males), there is possibility that the smaller group can be influenced by individual outliers whereas the female sample size provides a more stable estimate of the parameter under consideration.

However, there is also a possibility that these 5 seconds edge of male folks over the female students could attribute to factors such as approach in solving the problem, familiarity with the topics and their motivation towards the subject. This brings up the point that a more balanced data is essential in drawing more definitive conclusions about the impact of gender on average time required for correctly solving the tasks.

AVERAGE HELP COUNT

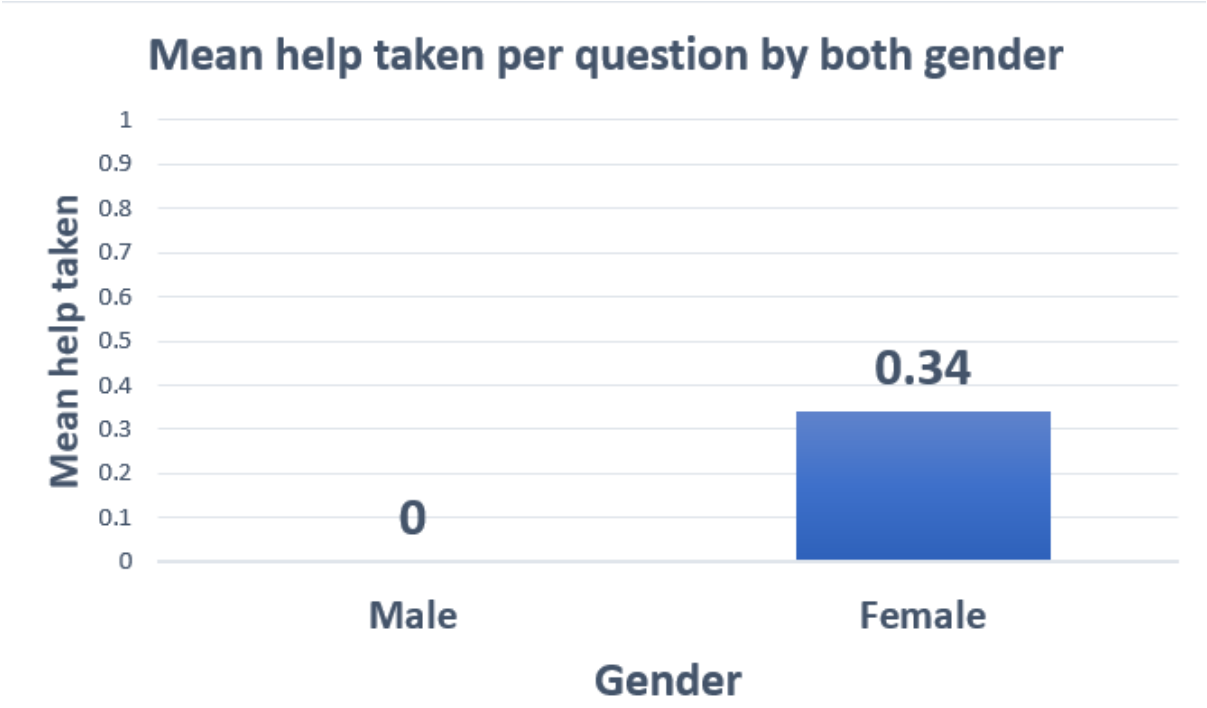


Figure 34: Mean help count per task by both genders

From the above representation, it can be seen that the female folks require around 0.34 help count for every correctly solved task or in other interpretable terms, females required around 3 helps for every correctly solved task and on the other hand, male folks did not access any

help in solving the tasks correctly. Once again, the huge disparity in number of participants between both the gender group skews the interpretation severely as this also had an impact on number of tasks that was attempted by both the groups (56 by males and 371 by females).

There are few factors that comes into play here. Firstly, the disproportionate number of participants throws the analysis off as said and this can only be rectified with a more balanced data. Next, the higher help seeking rate among the females can be a result of larger sample size where more diverse behaviour can easily arise while with only two males, the zero help count can be a statistical anomaly which might not be the case with a larger male group. Furthermore, it could be possible that the tasks solved by male folks were less complicated than the ones solved by the female students.

Some of the measures that could be taken here are making the data set more balanced which reduces individual variances and outliers which severely impacts the analysis, assigning the tasks that are of comparable difficulty to both the gender groups and taking into consideration, other factors such as familiarity with the concepts and approach used to solve the tasks.

WHICH AGE GROUP TOOK MORE TIME AND HELP TO SOLVE THE TASKS CORRECTLY?

Once again, it is to be noted that the tasks that have been solved correctly have been taken into consideration for analysis.

AVERAGE TIME TAKEN

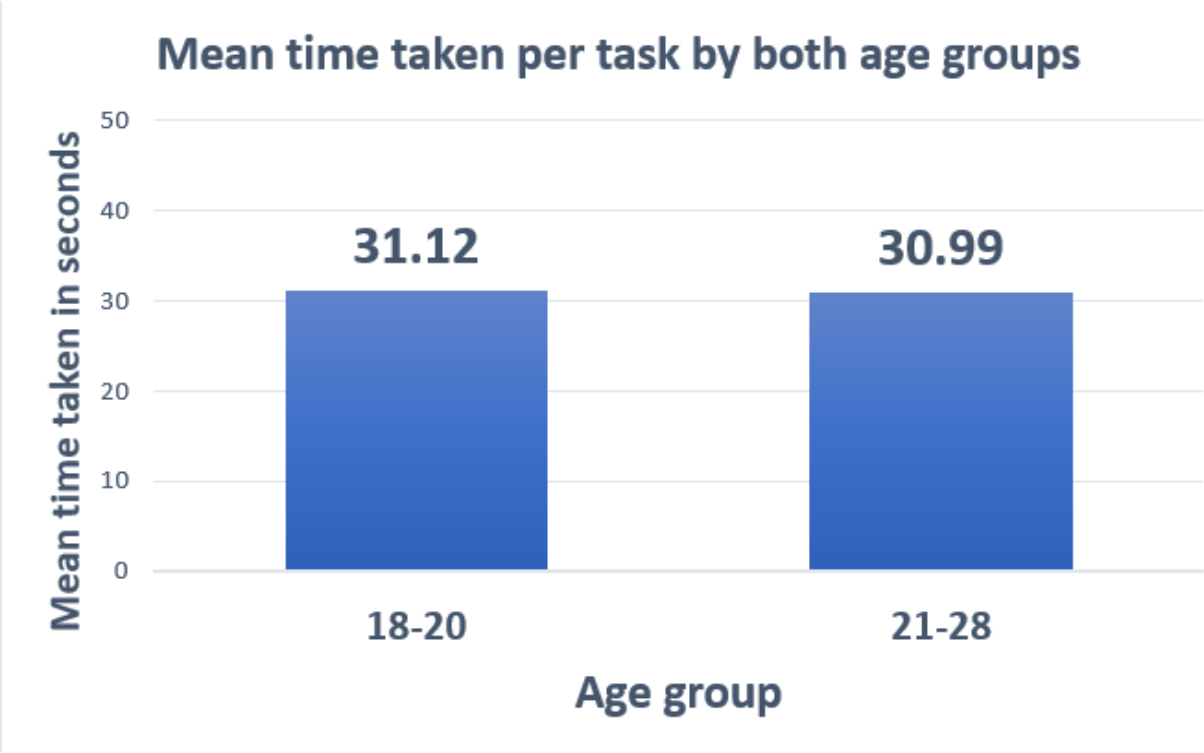


Figure 35: Mean time taken by both age group to solve a task correctly

It can be seen that there exists a tight competition between the two age groups with 18-20 age group peers requiring 31.12 seconds on average while the 21-28 age group needing around 30.99 seconds on an average giving a difference of 0.13 seconds between the two groups with the older population being slightly faster than the younger group.

As the difference here is negligible, it once again points out to the fact that age might not be predominant factor in determining a student's performance although there is a possibility that there might have been significant differences in level of difficulty encountered by both the age a groups in attempting to solve the tasks. Previous familiarity with the concepts can always be a factor which is not taken into consideration here.

Anyhow, it would be advisable to have further studies where both the age groups are presented with problems of similar complexity in order to be able to better underline the impact of age on performance.

AVERAGE HELP COUNT

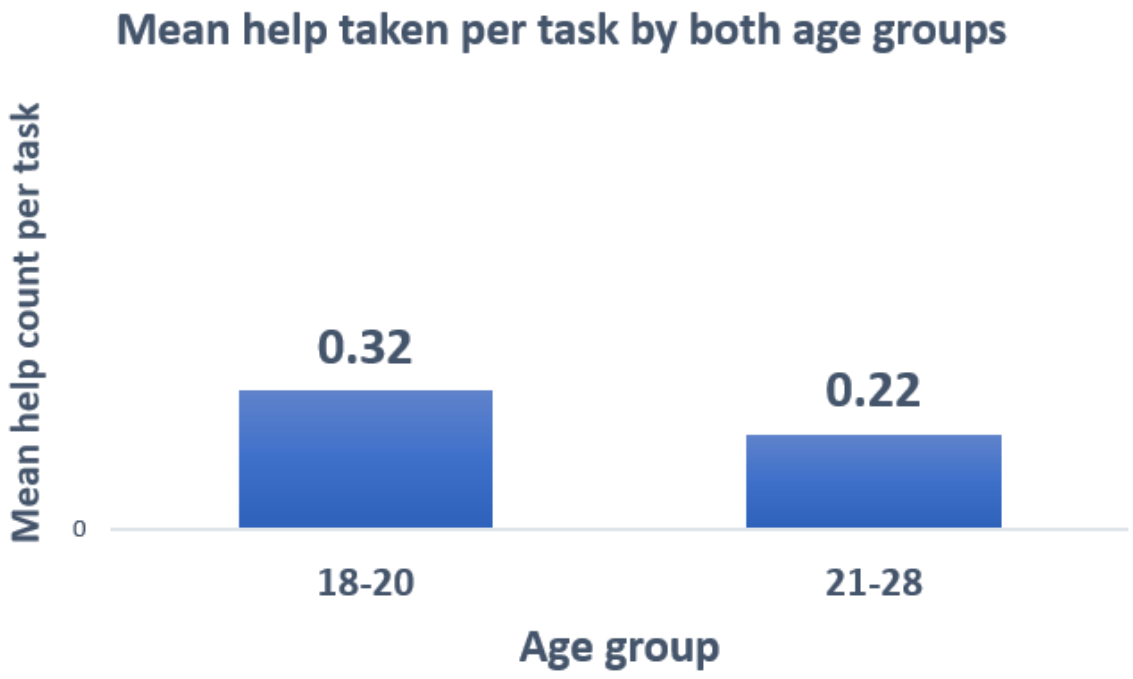


Figure 36: Mean help count per task by both age groups

In terms of average help count, a slightly different pattern can be observed between the two age groups. The 18-20 age group required an average of 0.32 instances of help per question, while the 21-28 age group required slightly less help, averaging 0.22 instances per question.

Here the younger folks appear to have heavily relied on the help to solve the tasks correctly than the older age group which highlights the fact that the older age group might have had a better familiarity and hold over the concepts. It is also possible that the older folks might have encountered these topics more than once due to their failed previous attempts which gives them an edge over the younger population. The fact that the younger folks needed 45% more help than the older students reflect the differences in problem solving strategies, difference in confidence levels and also the cognitive maturity which is out of scope for this project.

Here, the younger students could be provided with more support and guidance along with some targeted materials that could improve the problem-solving approach and thereby reducing their dependence on help to solve the tasks.

IMPLEMENTATION OF MACHINE LEARNING FOR ANSWERING THE HYPOTHESES

Along with the interpretation of university students' performance in mathematics, determining their strengths and weaknesses and making the appropriate suggestions to enhance their grades, answering two hypotheses which were formulated before in the objective section is also a part of this master thesis.

Recalling the two hypotheses formulated in the objective section:

1. How do factors such as gender, age and field of study (STEM/non-STEM) of university students influence their performance in mathematics?

   H0: Age and gender, both have a positive correlation with the students' performance in mathematics

2. What conclusion can be drawn from analysing usage parameters (time taken and help count) to solve the exercises and student information?
   H1: Students from non-STEM background require more time in solving the problems and also tend to access more help in comparison to STEM background students

While the first hypothesis is answerable due to the diversity in the age group and gender inputs from the students, the second hypothesis had to be ruled out due to the fact that the course of study opted by all of the participated students came out to be singular ('andere'). As a result, only the effect of age group and gender on the performance in mathematics has been answered by the following machine learning algorithms.

LOGISTIC REGRESSION

As discussed earlier in the theoretical section, logistic regression is a classification algorithm that helps in binary classification of the data. Here, in our case, logistic regression has been used to classify the data set based on whether or not the students were able to correctly solve the tasks attempted by them based on their age group and gender.

In mathematical terms, for the probability of solving a task correctly by a student as output, their age group and gender parameters were taken as input.

The steps for implementing logistic regression are as follows:

1. Initially, the variable of both age group and gender are one-hot encoded using the 'get_dummies' function in 'pandas' library which converts each categorical value into

binary ones. This is required as the categorical values are in string format and cannot be manipulated by the algorithm.

2. The encoded values are then split into input (X = ['ageGroup_mapped', 'gender_mapped']) and output (Y = ['is_Solved']) variables and the data is divided into to test and train dataset using the train_test_split with a test size of 20% of the data.
3. Next, the logistic model is trained with the training dataset created in the previous step and subsequently, predictions are made on the test data (X_test).
4. Based on the comparison between the predictions results (y_prediction) and the actual test data (y_test), a confusion matrix, a classification report and a Receiver Operating Characteristics curve are generated which gives us information about the accuracy of the model and p-values of the variables involved for hypothesis testing
5. Finally, a logit model from the statsmodel is used in building a logistic regression equation and is presented below.

RESULTS AND DISCUSSION

1. CONFUSION MATRIX

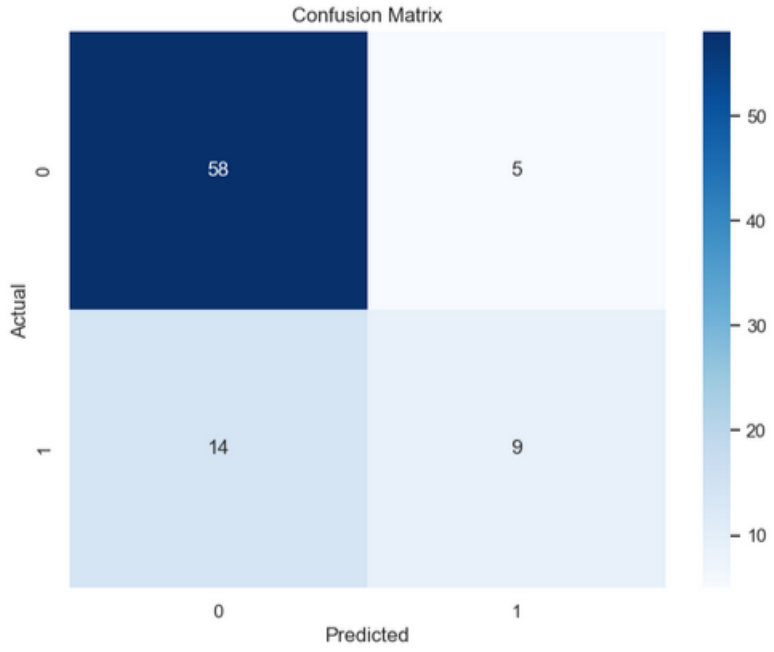


Figure 37: Logistic regression confusion matrix

The confusion matrix can be explained as follows:

- **True Negatives (Top-left: 58)**: These are the correctly predicted instances where the model predicted that a student did not solve the task (0), and they actually did not.

- **False Positives (Top-right: 5)**: The model predicted that a student solved the task (1), but they actually did not.
- **False Negatives (Bottom-left: 14)**: The model predicted that a student did not solve the task (0), but they actually did solve it.
- **True Positives (Bottom-right: 9)**: These are the correctly predicted instances where the model predicted that a student solved the task (1), and they actually did.

  An accuracy of **78%** was obtained for the model. Precision for "solved" (class 1) is **0.64**, and recall is **0.39**. The model struggles to correctly identify students who solved the task (low recall). This could be due to the lack of positive cases (class 1) in the dataset. However, A score of 0.78 shows that the model is fairly balanced in terms of how accurately it identifies both "solved" and "not solved" cases.

2. ROC curve

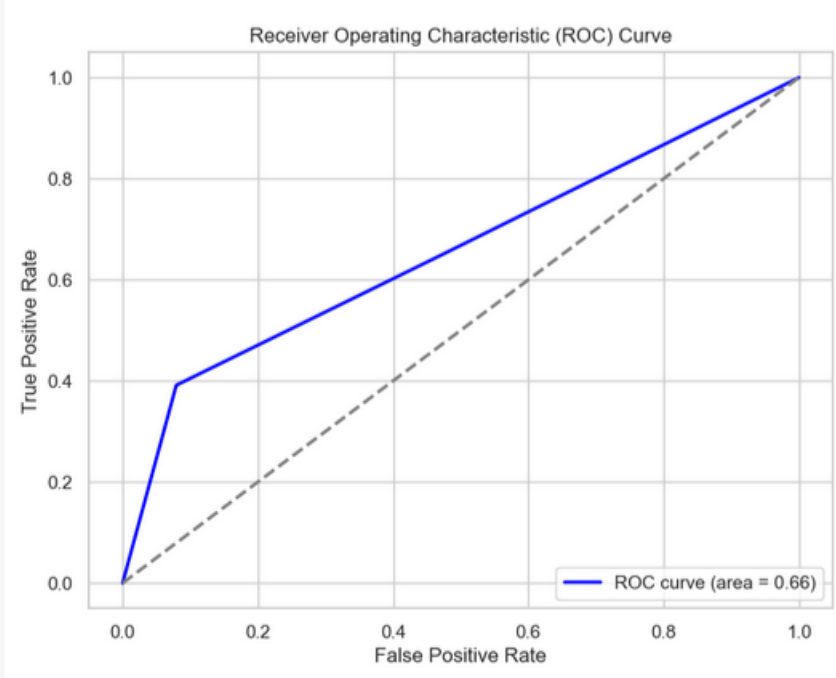


Figure 38: Logistic regression ROC curve

The ROC gives the trade-off that exists between True-Positive-Rate (Sensitivity) and the False-Positive-Rate (1-Specificity). The Area Under the Curve (AUC) came out to be 0.66 which suggests that the model has a moderate ability to distinguish between solved and not solved classes.

3. STATMODEL SUMMARY, HYPOTHESES ANSWERING AND LOGISTIC REGRESSION EQUATION

PARAMETERS	Co_eff	Std_err	Z	P >|z|	[0.025	0.075]
Intercept	-0.1	0.41	0.041	0.968	-0.787	0.82
Age_group	0.38	0.263	1.402	0.161	-0.147	0.885
Gender	-1.08	0.366	-3.322	0.001	-1.932	-0.498

A threshold of **0.05** was set for testing the hypothesis and the result are as follows:

- A p-value of **0.161** for the age group variable shows that the age group has comparatively less significant effect on the output as the **P >|z|**.
- On the other hand, gender with a coefficient of **0.001** indicates that is is a significant predictor of students' probability of correctly solving a task as **P <|z|**.

By making use of the coefficients and intercepts from the table, the following logistic regression equation has been built:

logit(P) = -0.1 + (0.38 * ageGroup) + (-1.08 * gender)

Since the age group doesn't have much effect on the output, it can be taken out of the equation and the final form of the equation is as follows:

logit(P) = -0.1 + (-1.08 * gender)

DECISION TREE

The second classification algorithm utilised for the purpose of answering the hypothesis was decision tree. Once again, the probability of solving a task correctly by a student was set as output and their age group and their gender parameters were set as input.

The steps for implementing decision tree are as follows:

1. Initially, the variable of both age group and gender are one-hot encoded using the 'get_dummies' function in 'pandas' library which converts each categorical value into binary ones. This is required as the categorical values are in string format and cannot be manipulated by the algorithm.

2. The encoded values are then split into input (X = ['ageGroup_mapped', 'gender_mapped']) and output (Y = ['is_Solved']) variables and the data is divided into to test and train dataset using the train_test_split with a test size of 20% of the data.
3. Next, the decision tree model is fitted with the training dataset created in the previous step and subsequently, predictions are made on the test data (X_test).
4. The hyperparameters are tuned using the grid search CV method with the following parameter grid:

```
1. param_grid = {
2.     'criterion': ['gini', 'entropy'],
3.     'max_depth': [None, 10, 20, 30],
4.     'min_samples_split': [2, 10, 20],
5.     'min_samples_leaf': [1, 5, 10]
6.}
```

5. Based on the comparison between the predictions results (y_prediction) and the actual test data (y_test), a confusion matrix, a classification report and a Receiver Operating Characteristics curve are generated which gives us information about the accuracy of the model and p-values of the variables involved for hypothesis testing.
6. As a last step, a feature importance model was built using the feature_importances_ based on the best decision tree built using the GridSearchCV estimated parameters and the formulated hypothesis was answered.

RESULTS AND DISCUSIION

1. The confusion matrix, the accuracy of the model and the ROC curve turned out to be exactly the same as that of logistic regression. A short summary of the results is given below:

   - **Accuracy: 78%**
   - **Area Under Curve (AUC) for ROC: 66%**

   This tells us that the model has a fairly moderate power in differentiating between the solved and the not solved class and the reason again lies in the fact that there exists an imbalance in the data between the gender groups and the total number of tasks attempted by each group. The decision tree obtained from the model that was built is given below:

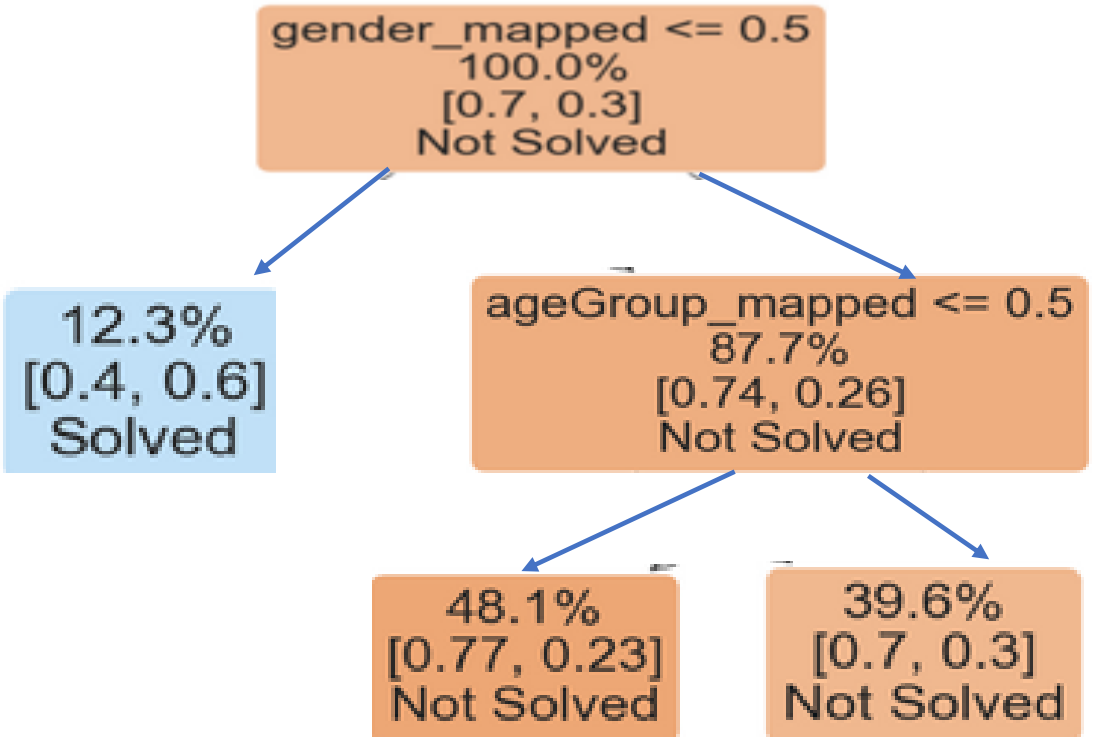


Figure 39: Branching of decision Tree based on gender and age group

Explanation of the tree splitting:

Root Node: gender_mapped <= 0.5

➔ **Interpretation:** The first split is based on gender.

➔ **Node Details:**

- 100% of the cases in this branch were not solved.
- The distribution is [0.7, 0.3] (70% male, 30% female).

Left Child Node: Solved

➔ **Condition:** gender_mapped <= 0.5

➔ **Outcome:** 12.3% of the problems were solved.

➔ **Distribution:** [0.4, 0.6] (40% male, 60% female)

➔ **Inference:** Problems solved in this branch are predominantly by females.

Right Child Node: ageGroup_mapped <= 0.5

➔ **Interpretation:** Further splits the not solved cases by age group.

➔ **Node Details:**

- 87.7% of the remaining cases were not solved.
- The distribution is [0.74, 0.26] (74% in the younger age group (18-20), 26% in the older age group (21-28)).

Leaf Node 1 (Left of ageGroup_mapped): Not Solved

➔ **Condition:** ageGroup_mapped <= 0.5

➔ **Outcome:** 48.1% of the problems were not solved.

➔ **Distribution:** [0.77, 0.23] (77% younger age group, 23% older age group)

➔ **Inference:** A significant portion of the not solved cases comes from the younger age group.

Leaf Node 2 (Right of ageGroup_mapped): Not Solved

➔ **Condition:** ageGroup_mapped > 0.5

➔ **Outcome:** 39.6% of the problems were not solved.

➔ **Distribution:** [0.7, 0.3] (70% younger age group, 30% older age group)

➔ **Inference:** This leaf also shows a higher rate of unsolved problems from the younger age group, but not as pronounced as the previous node.

2. FEATURE IMPORTANCE

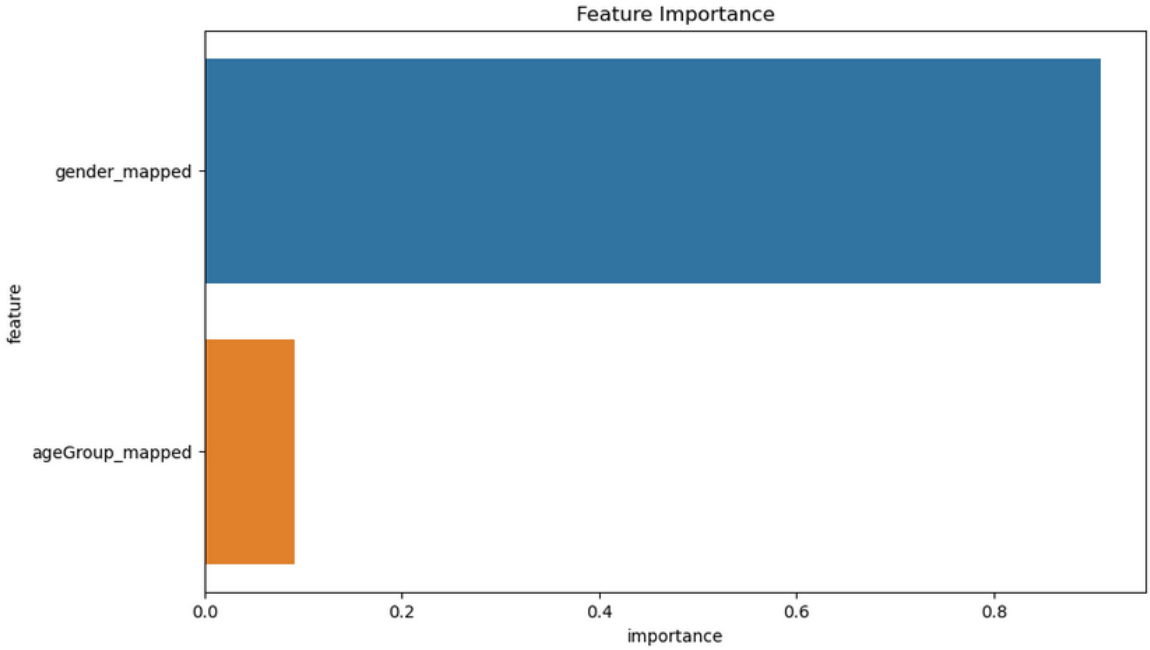


Figure 40: Feature importance based on the best decision tree

Results generated:

1. **Age group: 9.2%**
2. **Gender: 90.8%**

Here, the explanation is straight forward and the feature importance which is based on the best decision tree model (built using the grid search CV method) tells us that gender is the significant predictor of students' performance in mathematics with a whopping importance value of **90.8%.**

With this, we can say that decision tree is also inclined towards the fact that gender is the stronger predictor of the two variables.

SUPPORT VECTOR MACHINE

The final Machine learning algorithm implement for the hypothesis testing was Support Vector machine or SVM. As performed in the other two algorithms, the probability of solving a task correctly by a student was set as output and their age group and their gender parameters were set as input.

The steps for implementing SVM are as follows:

1. Initially, the variable of both age group and gender are one-hot encoded using the ‘get_dummies’ function in ‘pandas’ library which converts each categorical value into binary ones. This is required as the categorical values are in string format and cannot be manipulated by the algorithm.
2. The encoded values are then split into input (X = ['ageGroup_mapped', 'gender_mapped']) and output (Y = [‘is_Solved’]) variables and the data is divided into to test and train dataset using the train_test_split with a test size of 15% of the data.
3. The features need o be standardized for SVM model and this was done using the StandardSclaer function from the preprocessing library.
4. Next, the SVM model is fitted with the training dataset created in the previous step and subsequently, predictions are made on the test data (X_test).
5. Based on the comparison between the predictions results (y_prediction) and the actual test data (y_test), a confusion matrix, a classification report and a Receiver Operating Characteristics curve are generated which gives us information about the accuracy of the model and p-values of the variables involved for hypothesis testing.
6. In order to answer the hypothesis, coefficients for linear SVM were extracted which indicates the significance of features in determining the output parameter.

RESULTS AND DISCUSIION

1. CONFUSION MATRIX

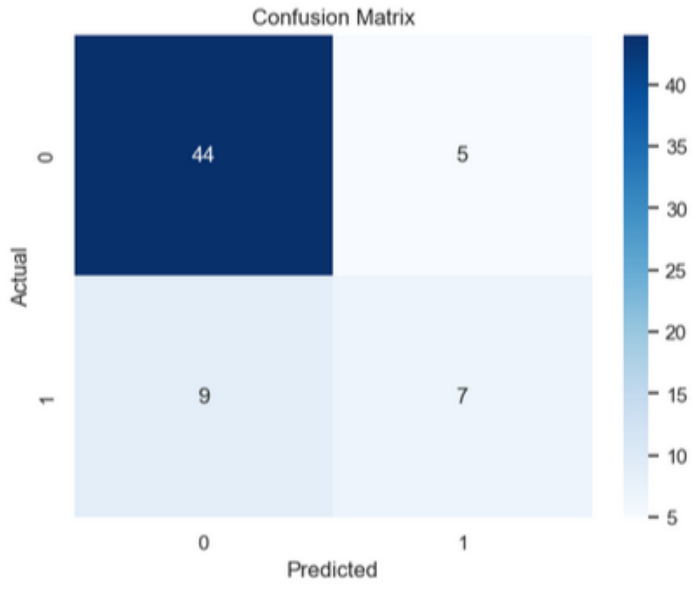


Figure 41: SVM Confusion matrix

The confusion matrix can be explained as follows:

- **True Negatives (Top-left: 44)**: These are the correctly predicted instances where the model predicted that a student did not solve the task (0), and they actually did not.

- **False Positives (Top-right: 5)**: The model predicted that a student solved the task (1), but they actually did not.
- **False Negatives (Bottom-left: 9)**: The model predicted that a student did not solve the task (0), but they actually did solve it.
- **True Positives (Bottom-right: 7)**: These are the correctly predicted instances where the model predicted that a student solved the task (1), and they actually did.

  With this, the precision for "solved" (class 1) is **0.58** and recall is **0.44**. Although the SVM model has a slightly better recall than the Log-Reg model, a recall of 0.44 is still not good enough for predicting the correctly solved classes and this can be rectified by bringing balance between the gender group in the data set.

2. However, the optimum test size determined for SVM was 15% for which the accuracy came out to be **77%** indicating a moderate power in discriminating between the two classes (solved and not solved). Furthermore, this is in close proximity with the accuracy of the other two ML algorithms used

3. **SVM boundary plot**

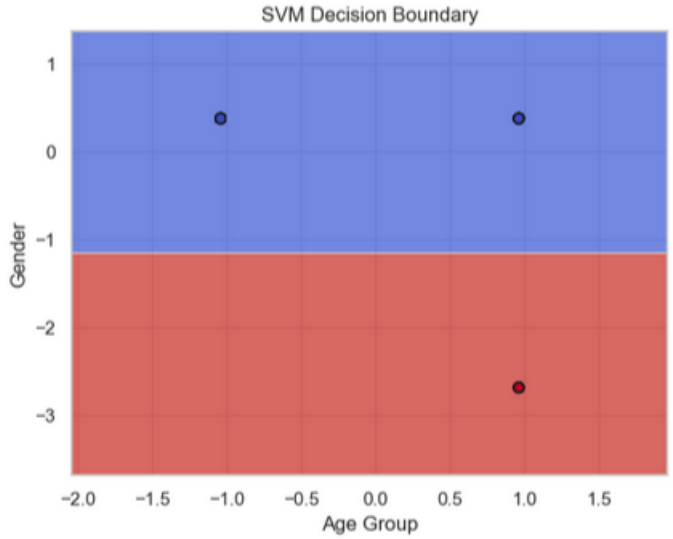


Figure 42: SVM decision boundary

The horizontal decision boundary indicates that the SVM model primarily uses the "Gender" feature for classification in this specific scenario. The "Age Group" feature does not seem to significantly influence the classification, as the boundary does not slope or change with varying "Age Group" values.

4. COEFFICIENTS OF SVM

The coefficients for the gender and age group for the implemented SVM model came out as follows:

- **Age group = 2.0682e-8**
- **Gender = -0.6535**

This relatively larger coefficient of gender (compared to "ageGroup_mapped") indicates that the "Gender" feature significantly impacts the decision boundary.

PRESCRIPTIVE STATISTICS

Now that the analysis on how the students have performed on the math questionnaire has been conducted and few basic questions such which group attempted the greatest number of tasks, who took more time and help in solving the tasks have been answered along with the formulated hypothesis, this section will look at the topics with which each age group and gender group struggled the most by taking into account, the number of incorrectly solved tasks in each topic. The below bar diagram shows us the number of incorrect attempts made by students in each age group and gender:

BASED ON AGE GROUP

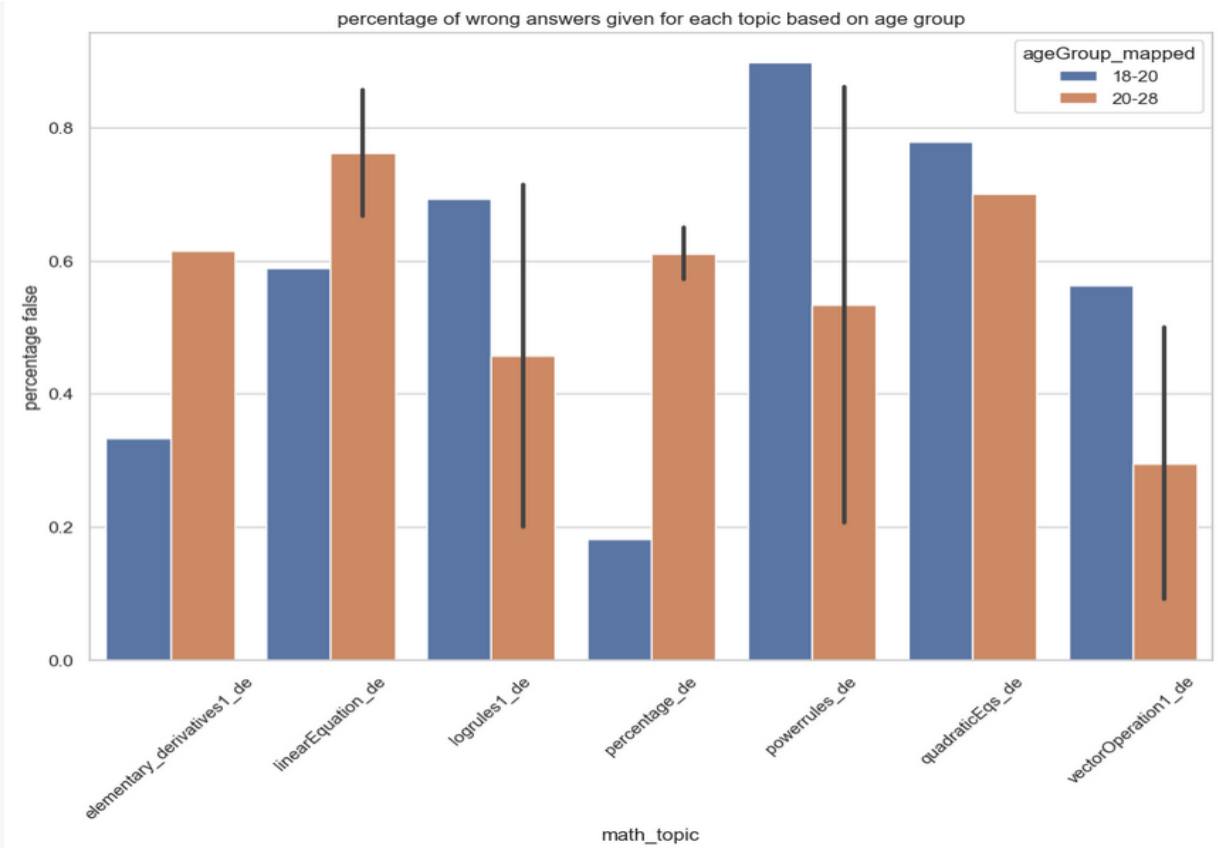


Figure 43: Analysis based on topics attempted by both age groups

While a casual look at it tells us that the older age group have performed slightly better in most of the topics, a closer look at the diagram reveals the following information:

- The younger age group (18-20) tends to have higher percentages of wrong answers in most topics, especially in quadratic_eqns and PowerRules.

- The older age group (21-28) performs better in topics like log rules and vector operation but struggles more with linear equations and quadratic equations.

BASED ON GENDER

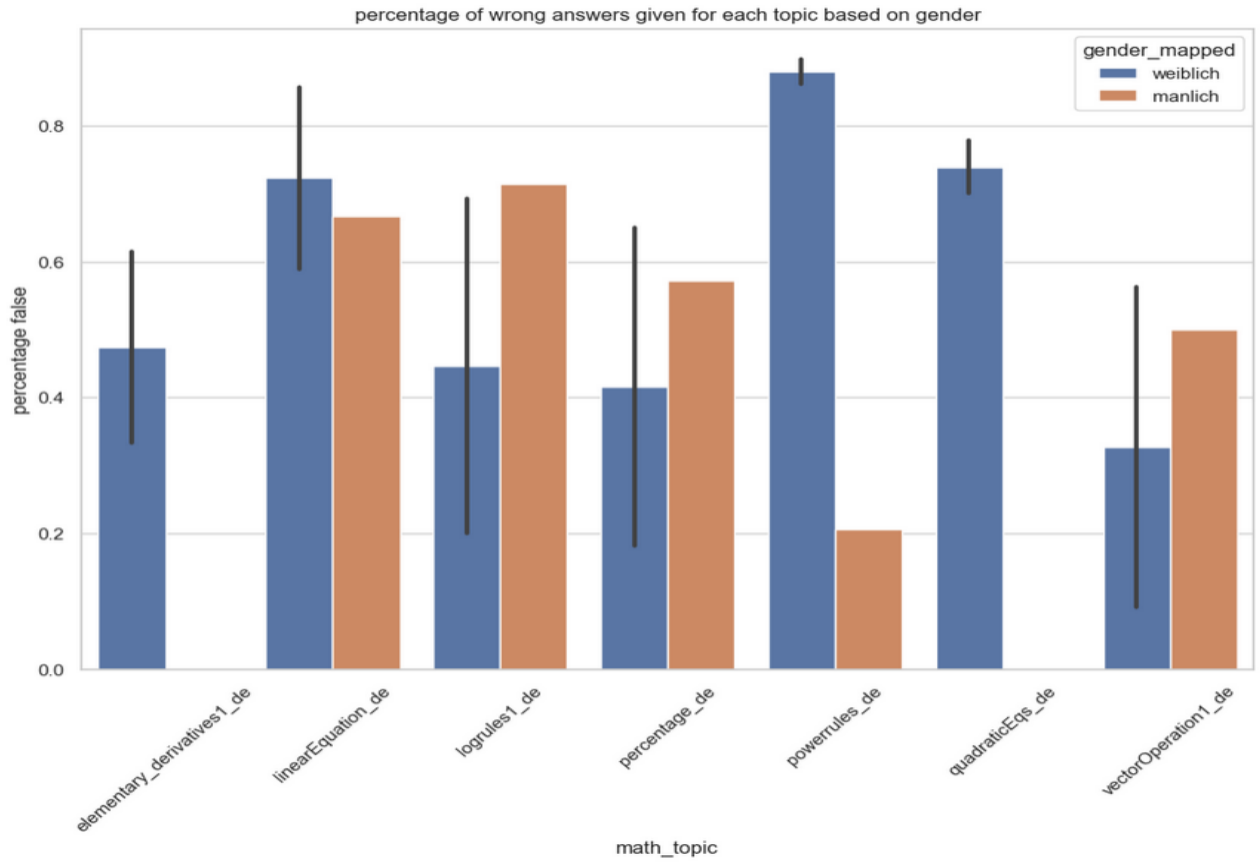


Figure 44: : Analysis based on topics attempted by both genders

From the diagram, it looks like the male folks tends to have fewer percentage of wrong in most of the concepts. However, the fact that only 2 male participants were there had to be considered here as the imbalance has skewed the interpretation. A closer scrutinization of the data lead to the following points

- Male students have a problem with the topics vector operation, log rules and percentage related tasks.
- The female students tend to struggle with linear equations, quadratic equations and power rules.

TOPIC SUGGESTIONS TO THE COMBINATION OF EACH AGE GROUP AND GENDER

By knowing the topics in which each gender and age group requires intervention, a suggestion can be made to a particular age group and gender combined by implementing the following criteria:

1. Number of false counts of a particular age group or gender**>** mean of false counts for that particular topic
2. Mean time taken for correctly solved of a particular age group or gender **>** mean time taken for correctly solved for that particular topic
3. Mean help taken for correctly solved of a particular age group or gender **>** mean help taken for correctly solved for that particular topic

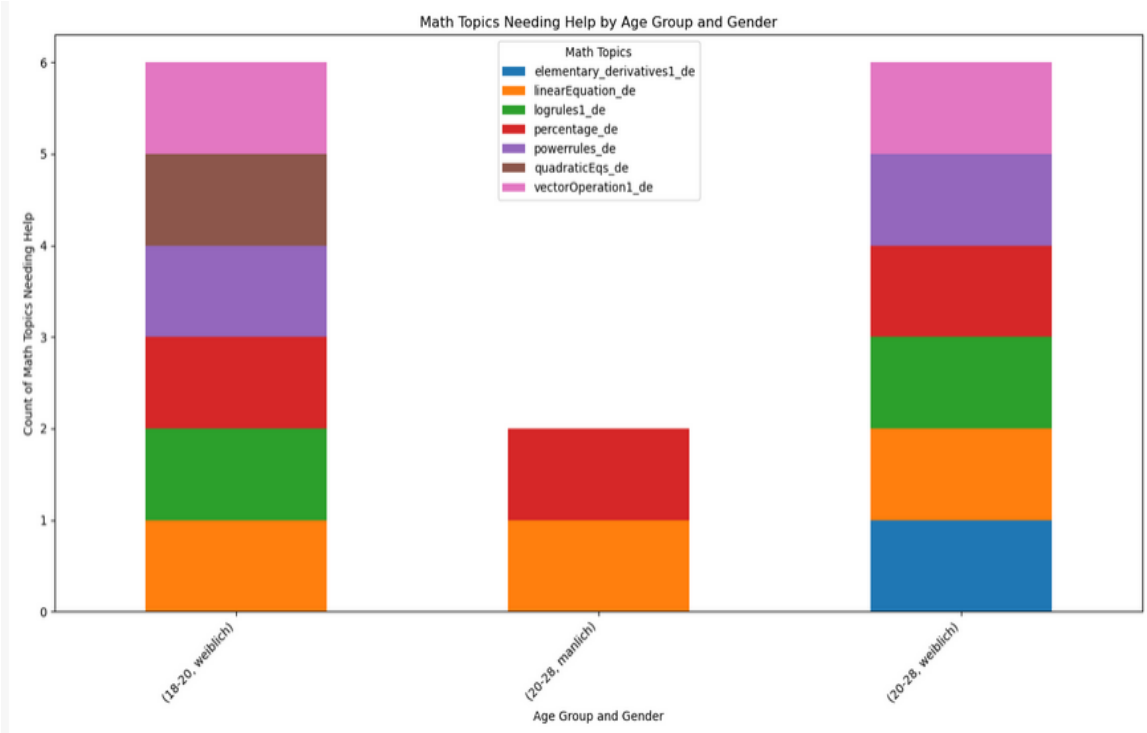
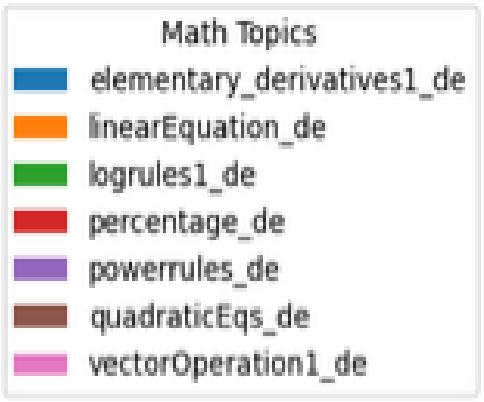


Figure 45: Topics suggested to different age groups and genders



By providing the students with materials, tips and tricks and targeted interventions by the faculty if necessary related to these topics to the combination of age group and gender mentioned in the bar plot above, it would be possible to see an enhancement in the performance of the students and consequentially a decrease in number of incorrectly solved task count in the analysis.

CONCLUSION

Now that we have seen how the whole algorithm functions, from gathering the data to providing prescriptions based on the result of the analysis, we will have a look at some of the challenges faced in the completion of the project, its limitations, some of the implementations that can be incorporated for its betterment in the future and finally summarize the projects findings.

LIMITATIONS AND CHALLENGES ENCOUNTERED

As we have seen in the earlier sections, one of the biggest limitations of the project was the size of the data set we had to work with. Since the data set was limited to 18 students, the scope of the analysis was severely hindered as the implementation of many of the machine Learning algorithms that work mainly with a moderate data set size would lead to results that are inaccurate and some of the unsupervised ML algorithms such as K-means and PCA which require data sets in the size of at least hundreds or a minimum of five to ten cases per variable could not be made use of.

The second challenge faced in the execution of analysis section is that there was an imbalance in the gender ratio (8 female: 2 male) which might have skewed the analysis to some extent as it is possible that the participants might have possessed an entirely different level of fundamental understanding of the concepts which in turn has an effect on the results they achieved and moreover, there was no information of the students' background education

(whether they are from a strong mathematical background) which plays an important role in how they would perform.

The next limitation posed was no information about the course of their study because of which one of the hypothesis questions had to dismissed entirely which further put a damper on us understanding how the students' field of study impacts their performance in mathematics and how can we help them to enhance their performance with this piece of information.

The last challenged faced was unavailability of the data from some students which was essential for the analysis. Either due to the concerns of identity protection or due to their reluctance to enter information about themselves, general data from eight out of eighteen participants were missing which further deteriorated the situation of interpreting the results from the questionnaire. As the general information about these students were unavailable, their attempts to solve the tasks in the questionnaire were simply invalidated as these participants could not be assigned to any of the classes which would serve in answering our hypotheses and eventually helping them out in making suggestions regarding the topics in which they are lagging behind.

All these instances mentioned above directly or indirectly had a consequence on answering the hypotheses questions formulated which in turn repressed the prescriptive phase of the project where appropriate suggestions were made to the combination of each age group and gender.

SUMMARY OF THE RESULTS

The initial stages of analysis with descriptive statistics showed that female participants had a greater overall commitment with around 46 tasks per female participant in comparison the male folks who had an average of 23 tasks per male student when the ratio imbalance between the two genders is not taken into consideration of course. The next comparison was between the two age groups namely 18-20 and 21-28, which showed a negligible difference of 17 tasks with the older age group attempting to solve 222 tasks. With this, the younger population had an average of 41 tasks per students while the older population, an average of 44.4 per student.

In the next section a comparison between the two gender was made to see who required more time and help in solving the tasks and the answers revealed that the female participants took around 5 minutes more on an average to solve the tasks correctly than the males who required 27.22 seconds to solve a task correctly. In terms of help required to solve the task correctly, male folks required almost no help at all while the females required around one help for every 3 question to solve a task. But this again is largely attributed to the fact that there were only two male participants and they had a lesser average in total number of tasks attempted.

Next came the comparison between the two age groups and the first parameter was the time taken to solve the tasks correctly. Here like before, there was a trifling difference of 0.13 seconds between the two groups with the older population taking around 30.99 seconds to solve a task correctly which could be attributed to their stronger foundation and greater experience with the concepts. In regards to the help count required on an average to correctly solve the tasks, the younger ones accessed 45% more help than the older students indicating a difference in confidence levels and problem-solving approaches.

These findings gave a faint idea of the answer to our hypothesis that gender might be a stronger parameter of the two in predicting the students' performance whereas age factor doesn't play as biga role as gender.

The next part of the project introduced the machine learning algorithms in order to answer the hypotheses questions formulated one of which was to determine whether age group and gender had any type of correlation with the students' performance while the other being how a students' background (STEM or non-STEM) influence their performance. The second hypothesis which was about studying the influence of students' background (STEM or non-STEM) had to be taken out due to the unavailability of students' data regarding their educational background.

Logistic regression, decision tree and support vector machine were utilised for the cause with age group and gender as the input and whether or not a students solved the task as the output.

The first ML algorithm implemented was Logistic regression which had an accuracy of 78% and outputted a p-value of 0.161 for age group and 0.001 for gender indicating that gender is a strong predictor of students' performance in mathematics while age group had a negligible effect with a threshold of 0.05.

The second algorithm, decision tree also had an accuracy of 78% and implemented feature importance for determining the importance of age group and gender in predicting a student's performance. It was seen that the gender had an importance of around 91% while age group came out with a paltry 9% importance which pointed out that gender was the more important factor for the second time.

The last ML algorithm SVM or Support Vector Machine showed no different result from the other two algorithms through its coefficient feature and the SVM boundary plot. It was seen in the plot that the SVM boundary did not slope or change with varying age group and gender was primarily used for the purpose of classification. The gender had a coefficient of -0.65 whereas the age group came out with a value of 2.06e-8 which is negligible. This once again proved that age group was not crucial in determining the students' results.

As the result of all three ML algorithm were in coherence with each other highlighting the fact that gender is a critical factor in deciding how the students perform in mathematics and the age group factor had trivial to no effect on their performance, it can be concluded that the

gender should be the primary factor in handling the students for classification purposes and making suggestions.

FUTURE IMPLICATIONS

Based on the challenges and limitation presented in the previous section and also the findings from the analysis, it is evident enough that there are certain changes and implementations in the methodology that could drastically improve the results of the analysis and simultaneously make better suggestions to the study groups that could be helpful for the students in achieving better results in mathematics. Some of the implementations have been discussed below:

- ➔ Increasing the data size and uniformizing the participation
  An effort to gather more data from a wider range of participants helps in ensuring that there is a balance across gender, age groups and educational background which in turn leads to a more robust model. One way to do this is to inform the students the benefits of participation which is helping them improve their grades as this would increase the chances of drawing in more participants. It would also be very convenient for the analysis if the students taking part in the questionnaire receives questions from the same math topics in an order in a similar pattern with gradually increasing difficulty levels. This would offer a better comprehension of students' mathematical capabilities which leads to a better classification and suggestions. It is also of the utmost importance that students' privacy protection is maintained and also bring this to the attention of the students as this could possibly become the reason for hesitating from entering their general details which in turn affects the analysis.

- ➔ Optimizing the model performance
  Now that the model is set in motion, the performance of the students in the exam after they have implemented the suggestions made by the algorithm can be compared with their previous performance which gives an understanding of the model efficiency. By operating the model as a continuous loop, minor tweaks in hyperparameters of the ML algorithms, implementation of cross-validation like K-fold cross validation and exploring and incorporating additional variables such as academic history, usefulness of the subject for their course of study and engagement levels, the predictive power of the model can be enhanced.

- ➔ Developing a feedback system
  It is essential to have a feedback system from the students as this can help us in getting a students' perspective about the model and how lucrative it is for them as they are the end user of the model and any feedback would be directly helpful in enhancing their performance.

➔ Interactive platform and live suggestions
An interactive platform that offers live suggestions when a student is struggling with a particular topic can boost a student's performance significantly. One of the ways to achieve this is with chatbots which directly interact with a student. Furthermore, personalising the learning paths based on their performance, gender or age group helps in strengthening their weak points.

➔ Incorporating a student-faculty interaction
Establishing a communication channel between the students and faculty where the faculty can have an overview about the students' performance on a dashboard based on which a student can be offered a targeted timely intervention can prevent a student from falling behind or dropping out of the course.

Finally, it can be said that the future of this model lies firmly in robust data gathering and utilising the entire feature set and at the same time developing an interactive environment which unites the faculty and the students together. Larger the dataset and more refined the models are, more accurate will the predictions become which can be seen through enhanced students' performance. This along with a balanced participation, live feedback system and robust ML models, that project can eventually evolve into dynamic Learning Management System (LMS) that caters to the need of each student and also help the faculty better understand their students.

REFERENCES

[1] Wiki, “History of artificial intelligence.”

[2] “The New York Times.”

[3] “PISA study 2022.”

[4] H. Luan and C. C. Tsai, “A Review of Using Machine Learning Approaches for Precision Education,” *Educational Technology and Society*, vol. 24, no. 1, 2021.

[5] City University of Hong Kong, Institute of Electrical and Electronics Engineers. Hong Kong Section, Institute of Electrical and Electronics Engineers. Xi’an Section, and Institute of Electrical and Electronics Engineers, *ICSPCC2016 : IEEE International Conference on Signal Processing, Communications and Computing : conference proceedings : City University of Hong Kong, August 5-8, 2016.*

[6] S. Joksimovic, V. Kovanovic, S. Joksimović, V. Kovanović, and S. Dawson, “The Journey of Learning Analytics,” 2019.

[7] B. Dietz-Uhler and J. E. Hurn, “Using Learning Analytics to Predict (and Improve) Student Success: A Faculty Perspective,” *Journal of Interactive Online Learning www.ncolr.org/jiol*, vol. 12, no. 1, 2013, [Online]. Available: www.ncolr.org/jiol

[8] M. Furqon, P. Sinaga, L. Liliasari, and L. S. Riza, “The Impact of Learning Management System (LMS) Usage on Students,” *TEM Journal*, vol. 12, no. 2, 2023, doi: 10.18421/TEM122-54.

[9] B. Dietz-Uhler and J. E. Hurn, “Using Learning Analytics to Predict (and Improve) Student Success: A Faculty Perspective,” *Journal of Interactive Online Learning www.ncolr.org/jiol*, vol. 12, no. 1, 2013, [Online]. Available: www.ncolr.org/jiol

[10] T. Wang *et al.*, “Exploring the Potential Impact of Artificial Intelligence (AI) on International Students in Higher Education: Generative AI, Chatbots, Analytics, and International Student Success,” *Applied Sciences (Switzerland)*, vol. 13, no. 11, 2023, doi: 10.3390/app13116716.

[11] M. Irfan, B. Kusumaningrum, Y. Yulia, and S. A. Widodo, “CHALLENGES DURING THE PANDEMIC: USE OF E-LEARNING IN MATHEMATICS LEARNING IN HIGHER EDUCATION,” *Infinity Journal*, vol. 9, no. 2, 2020, doi: 10.22460/infinity.v9i2.p147-158.

[12] “FERPA guidelines”.

[13] “Datenschutz Germany.”

[14] “HFU data guidelines.”

[15] R. B. Abubakar, R. Bada, and A. O. Dokubo Oguguo, “Age and Gender as Predictors of Academic Achievement of College Mathematics and Science Students,” *Journal of Educational and Social Research*, vol. 1, no. 2, 2011, [Online]. Available: https://www.researchgate.net/publication/371723996

[16] A. Priulla, N. D'Angelo, and M. Attanasio, "An analysis of Italian university students' performance through segmented regression models: gender differences in STEM courses," *Genus*, vol. 77, no. 1, 2021, doi: 10.1186/s41118-021-00118-6.

[17] E. E. C. Cornillez Jr., "Mining educational data in predicting the influence of Mathematics on the programming performance of University students," *Indian J Sci Technol*, vol. 13, no. 26, pp. 2668–2677, Jul. 2020, doi: 10.17485/IJST/v13i26.719.

[18] M. V. Almeda and R. S. Baker, "Predicting Student Participation in STEM Careers: The Role of Affect and Engagement during Middle School."

[19] W. Lake and W. Boyd, "Age, Maturity and Gender, and the Propensity towards Surface and Deep Learning Approaches amongst University Students," *Creat Educ*, vol. 06, no. 22, pp. 2361–2371, 2015, doi: 10.4236/ce.2015.622242.

[20] S. Basaran and G. Berberoglu, "An Exploration of Affective and Demographic Factors Regarding Mathematical Thinking and Reasoning of University Students," *Procedia Soc Behav Sci*, vol. 47, pp. 862–867, 2012, doi: 10.1016/j.sbspro.2012.06.748.

[21] A. Amani, H. Alamolhodaei, and F. Radmehr, "A gender study on predictive factors of mathematical performance of University students," 2011. [Online]. Available: http://www.interesjournals.org/ER

[22] M. S. A. Razak, S. Abdul-Rahman, and Y. Mahmud, "Mathematics Performance Monitoring System Using Data Analytics," in *2021 2nd International Conference on Artificial Intelligence and Data Sciences, AiDAS 2021*, Institute of Electrical and Electronics Engineers Inc., Sep. 2021. doi: 10.1109/AiDAS53897.2021.9574210.

[23] M. Fröhlich, S. Krauss, and S. Hilbert, "Using Machine Learning to Predict Mathematical Performance," in *Bridging the Gap: Empowering and Educating Today's Learners in Statistics. Proceedings of the Eleventh International Conference on Teaching Statistics*, International Association for Statistical Education, Dec. 2022. doi: 10.52041/iase.icots11.T3B3.

[24] S. E. Fienberg, "What is statistics?," *Annu Rev Stat Appl*, vol. 1, pp. 1–9, 2014, doi: 10.1146/annurev-statistics-022513-115703.

[25] Wikipedia, "Statistics," *Wikipedia*, Accessed: Oct. 05, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Statistics

[26] "Difference between Descriptive and Inferential stats, Bradley-university".

[27] "Comprehensive Guide to Descriptive vs Inferential Statistics!" Accessed: Oct. 07, 2024. [Online]. Available: https://www.simplilearn.com/difference-between-descriptive-inferential-statistics-article

[28] Hallahan, "Statistical power: concepts, procedures and applications".

[29] T. Baguley, "Understanding statistical power in the context of applied research," *Appl Ergon*, vol. 35, no. 2, pp. 73–80, 2004, doi: 10.1016/j.apergo.2004.01.002.

[30] E. Paradis, B. O'Brien, L. Nimmon, G. Bandiera, and M. A. T. Martimianakis, "Design: Selection of Data Collection Methods," *J Grad Med Educ*, vol. 8, no. 2, pp. 263–264, May 2016, doi: 10.4300/JGME-D-16-00098.1.

[31] M. Maseeh, "Peer-Reviewed, Multidisciplinary & Multilingual Journal INNOVATIVE DATA COLLECTION METHODS FOR RESEARCH IN THE DIGITAL ERA ABSTRACT: INTRODUCTION", [Online]. Available: http://vidyajournal.org

[32] Wikipedia, "Machine learning," Wikipedia.

[33] A. S. Pinto, A. Abreu, E. Costa, and J. Paiva, "How Machine Learning (ML) is Transforming Higher Education: A Systematic Literature Review," 2023, *IADITI ? International Association for Digital Transformation and Technological Innovation*. doi: 10.55267/iadt.07.13227.

[34] G. T. L. Brown and X. Zhai, "Editorial: Machine learning applications in educational studies," 2023, doi: 10.31219/osf.io/sg9jk.

[35] Wikipedia, "Supervised learning," Wikipedia. Accessed: May 08, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Supervised_learning

[36] L. Antonelli and M. R. Guarracino, "Special Issue on Supervised and Unsupervised Classification Algorithms—Foreword from Guest Editors," Mar. 01, 2023, *MDPI*. doi: 10.3390/a16030145.

[37] Wikipedia, "Regression analysis," Wikipedia. Accessed: May 08, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Regression_analysis

[38] Analytics Vidya, "Linear Regression: A comprehensive guide." Accessed: Oct. 22, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

[39] IBM, "What is Linear regression?", Accessed: Oct. 22, 2024. [Online]. Available: https://www.ibm.com/topics/linear-regression

[40] wiki, "Linear regression." Accessed: Oct. 22, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Linear_regression

[41] M. Omkar Patil, "Simple Liner Regression (SLR)."

[42] Ajitesh Kumar, "F-test & F-statistics in Linear Regression: Formula, Examples".

[43] Quantinsi, "Linear regression: Assumptions and Limitations." Accessed: Oct. 11, 2024. [Online]. Available: https://blog.quantinsti.com/linear-regression-assumptions-limitations/

[44] Engati, "Classification Algorithms." Accessed: Oct. 11, 2024. [Online]. Available: https://www.engati.com/glossary/classification-algorithm

[45] Datascientist, "Classification Algorithms: Definition and main models." Accessed: Oct. 11, 2024. [Online]. Available: https://datascientest.com/en/classification-algorithms-definition-and-main-models

[46] Wikipedia, "Logistic regression, wiki."

[47] Javapoint, “Logistic regression in machine learning.” Accessed: Oct. 11, 2024. [Online]. Available: https://www.javatpoint.com/logistic-regression-in-machine-learning

[48] KDnuggets, “ Become UNSTOPPABLE with data Become UNSTOPPABLE with data Classification Metrics Walkthrough: Logistic Regression with Accuracy, Precision, Recall, and ROC.” Accessed: Oct. 11, 2024. [Online]. Available: https://www.kdnuggets.com/2022/10/classification-metrics-walkthrough-logistic-regression-accuracy-precision-recall-roc.html

[49] Geeksforgeeks, “ Advantages and disadvantages of logistic regression.” Accessed: Oct. 11, 2024. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

[50] Wiki, “Decision Tree.” Accessed: Oct. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree

[51] R. Dulari and R. Nehra, “Decision Tree Algorithm for Data Science,” JETIR, 2021. [Online]. Available: www.jetir.org

[52] IBM, “What is Decision Tree?”, Accessed: Oct. 13, 2024. [Online]. Available: https://www.ibm.com/topics/decision-trees

[53] CodeAcademy, “Feature importance.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.codecademy.com/article/fe-feature-importance-final

[54] Wiki, “Feature selection.” Accessed: Oct. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Feature_selection

[55] IBM, “What are SVM’s?” Accessed: Oct. 14, 2024. [Online]. Available: https://www.ibm.com/topics/support-vector-machine

[56] Wiki, “Support vector machine.” Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine

[57] Javapoint, “Support Vector Machine Algorithms.” Accessed: Oct. 14, 2024. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[58] wiki, “Learning analytics.” Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Learning_analytics

[59] westernsydney, “what are the foundations of learning analytics?” Accessed: Oct. 14, 2024. [Online]. Available: https://lf.westernsydney.edu.au/engage/theory/what-are-the-foundations-of-learning-analytics

[60] D. Gašević, V. Kovanović, and S. Joksimović, “Piecing the learning analytics puzzle: a consolidated model of a field of research and practice,” 2017. doi: 10.1080/23735082.2017.1286142.

[61] valamis, “What is learning analytics?” Accessed: Oct. 14, 2024. [Online]. Available: https://www.valamis.com/hub/learning-analytics#importance

[62] Knowledgeworker, “Learning analytics in E-learning.” Accessed: Oct. 14, 2024. [Online]. Available: https://www.knowledgeworker.com/en/blog/learning-analytics

[63] L. Šereš, V. Pavlićević, G. Petrović, D. Horvat, and R. Ivanišević, “Learning analytics: Prospects and challenges,” *Strategic Management*, vol. 27, no. 3, pp. 48–55, 2022, doi: 10.5937/straman2200020s.

[64] A. Wilson, C. Watson, T. L. Thompson, V. Drew, and S. Doyle, “Learning analytics: challenges and limitations Learning analytics: challenges and limitations Learning analytics: challenges and limitations.”

[65] GeeksforGeeks, “json.loads() in Python.” Accessed: Oct. 20, 2024. [Online]. Available: https://www.geeksforgeeks.org/json-loads-in-python/

[66] GeeksforGeeks, “pandas Tutorials.” Accessed: Oct. 20, 2024. [Online]. Available: https://www.geeksforgeeks.org/pandas-tutorial/

APPENDICES

PYTHON CODE USED FOR DATA ANALYSIS AND ANSWERING HYPOTHESES:

```
import json
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

pd.set_option('display.max_rows', 1838)

with open('D:\Thesis\Thesis_coding\First_log_file\log0', 'r', encoding = 'utf-8') as f:
    data = [json.loads(line) for line in f]

df = pd.DataFrame(data)

df = df.sort_values(by=['client', 'userid'])

df = df.reset_index()

df.drop(columns = 'index', inplace = True)

df['client'].unique()

df = df.applymap(lambda x: {} if isinstance(x, float) and np.isnan(x) else x)

def extract_help_count(events):
    help_count = 0
    for event in events:
        if event['event'] == 'HELP':
            value = event.get('value', 0)   # 0 is the default value returned if no key value exists
            if isinstance(value, (int, float)):
                help_count = int(value) + 1
            elif isinstance(value, str) and value.isdigit():   #isdigit returns True if the string is a number. ex: '56'(string) - True
                help_count = int(value) + 1
    return help_count

# Apply the function to extract help count for each row
df['help_count'] = df['events'].apply(lambda x: extract_help_count(x) if isinstance(x, list) else 0)

from datetime import datetime

def calculate_time_difference(df):
    start_times = {}
    time_differences = []

    for index, row in df.iterrows():
        if row['type'] == 'START':
            start_times[row['seed']] = datetime.strptime(row['_meta']['time'], '%Y-%m-%dT%H:%M:%S.%fZ')
        elif row['type'] == 'FINISH':
            start_time = start_times.get(row['seed'])
            if start_time is not None:
```

```
                finish_time = datetime.strptime(row['_meta']['time'], '%Y-%m-
%dT%H:%M:%S.%fZ')
                time_difference = (finish_time - start_time).total_seconds()
                time_differences.append(time_difference)
            else:
                # If there is no corresponding start time, append None
                time_differences.append(None)

    # Pad with None values to match the length of the DataFrame
    num_pad = len(df) - len(time_differences)
    time_differences.extend([None] * num_pad)

    return time_differences

time_diffs = calculate_time_difference(df)

finish_counter = 0  # Counter to track the number of 'FINISH' events encountered

for index, time_diff in enumerate(time_diffs):
    if df.loc[index, 'type'] == 'FINISH':
        df.loc[index, 'time_taken_seconds'] = time_diffs[finish_counter]
        finish_counter += 1  # Move to the next time difference value

df['ageGroup'] = [row[0]['value']['ageGroup'] if isinstance(row, list) and len(row)>=3 and
row and isinstance(row[0], dict) and 'value' in row[0] and isinstance(row[0]['value'], dict) and
'ageGroup' in row[0]['value'] else None for row in df['events']]
df['course'] = [row[1]['value']['course'] if isinstance(row, list) and row and len(row)>=3
and isinstance(row[1], dict) and 'value' in row[1] and isinstance(row[1]['value'], dict) and
'course' in row[1]['value'] else None for row in df['events']]
df['gender'] = [row[2]['value']['gender'] if isinstance(row, list) and row and len(row)>=3
and isinstance(row[2], dict) and 'value' in row[2] and isinstance(row[2]['value'], dict) and
'gender' in row[2]['value'] else None for row in df['events']]

df['ageGroup_mapped'] = df['ageGroup'].map({"0": "18-20", "1": "21-28", "2":"28+"})
df['course_mapped'] = df['course'].map({"0": "MME", "1": "info", "2":"andere"})
df['gender_mapped'] = df['gender'].map({"0": "manlich", "1": "weiblich", "2":"divers"})

def is_solved_extracted(rows):
        if isinstance(rows, dict) and 'isSolved' in rows:
            return rows['isSolved']
        else:
            None

df['is_Solved'] = df['results'].apply(is_solved_extracted)

df_mapped = df.drop(["ageGroup", "course", "gender"], axis = 1)

print(df_mapped)

df_mapped['client'] = df_mapped['client'].astype(str)

missing_clients = df_mapped.groupby('client').filter(lambda x: x[['ageGroup_mapped',
'course_mapped', 'gender_mapped']].isnull().all().all() or (x[['ageGroup_mapped',
'course_mapped', 'gender_mapped']] == '').all().all())

unique_missing_clients = missing_clients['client'].unique()

print("Clients with missing information:")
print(unique_missing_clients)

df_missing = df_mapped[df_mapped[['ageGroup_mapped', 'course_mapped',
'gender_mapped']].notnull().any(axis=1)]

print(df_missing['client'].unique())
```

```
106. df_data = df_mapped[df_mapped[['ageGroup_mapped', 'course_mapped',
'gender_mapped']].notnull().any(axis=1)]
107.
108. df_data_unique = df_data.drop_duplicates(subset = 'client', keep = 'last') # first - drops
the first duplicates
109.
110. print(df_data_unique)
111.
112. missing_info = df_mapped.groupby('client').filter(lambda x: x[['ageGroup_mapped',
'course_mapped', 'gender_mapped']].isnull().all().any() or
113.                                                    (x[['ageGroup_mapped', 'course_mapped',
'gender_mapped']] == '').all().all())
114.
115. unique_missing_clients = missing_info['client'].unique()
116.
117. df_cleaned = df_mapped[~df_mapped['client'].isin(unique_missing_clients)]
118.
119. df_cleaned = df_cleaned[~df_cleaned['id'].isin([''])]
120.
121. df_cleaned['is_Solved'] = df_cleaned['is_Solved'].apply(lambda x: True if x == True else
False if x == False else False)
122.
123. df_cleaned['is_Solved'] = df_cleaned['is_Solved'].astype(int)
124.
125. df_cleaned = df_cleaned.reset_index()
126.
127. df_cleaned.drop(columns = 'index', inplace = True)
128.
129. columns = ['ageGroup_mapped', 'course_mapped', 'gender_mapped']
130.
131. for v in columns:
132.     # Find the first non-NaN entry
133.     non_na_indices = df_cleaned[pd.notna(df_cleaned[v])].index
134.     if len(non_na_indices) == 0:
135.         continue
136.
137.     start_index = non_na_indices[0]
138.
139.     # Initialize the group value
140.     grp_value = df_cleaned.at[start_index, v]
141.
142.     for i in range(start_index, len(df_cleaned)):
143.         if pd.notna(df_cleaned.at[i, v]):
144.             # Update the group value if a new entry is found
145.             grp_value = df_cleaned.at[i, v]
146.         else:
147.             # Populate the column with the current group value
148.             df_cleaned.at[i, v] = grp_value
149.
150. df_cleaned = df_cleaned[~df_cleaned['id'].isin(['courseSelectionExercise_de',
'surveystats'])]
151.
152. df_cleaned = df_cleaned.reset_index()
153.
154. df_cleaned.drop(columns = ['index'], inplace = True)
155.
156. df_cleaned = df_cleaned[df_cleaned['results'] != {}].reset_index()
157.
158. df_cleaned.drop(columns = ['index'], inplace = True)
159.
160. print(df_cleaned)
161.
162. grouped = df_cleaned.groupby('client')
163.
164. # Apply 'describe' function to each group and store the results in a dictionary
```

```
describe_results = {client: group[['time_taken_seconds', 'help_count']].describe(include='all') for client, group in grouped}

# Print the results
for client, description in describe_results.items():
    client_info = grouped.get_group(client).iloc[0][['ageGroup_mapped', 'course_mapped', 'gender_mapped']]
    print(f"Client: {client}")
    print(f"Age Group: {client_info['ageGroup_mapped']}, Course: {client_info['course_mapped']}, Gender: {client_info['gender_mapped']}")
    print(description)
    print("\n")

grouped = df_cleaned.groupby(['client', 'ageGroup_mapped', 'gender_mapped', 'course_mapped'])

# Calculate and visualize correlation matrix for each client
# for group, data in grouped:
#     client, age_group, gender, course = group
#     print(f"Correlation matrix for Client: {client}")

#     # Selecting only numeric columns for correlation calculation
#     numeric_data = data.select_dtypes(include=['number'])

#     if not numeric_data.empty:
#         corr_matrix = numeric_data.corr()
#         plt.figure(figsize=(8, 6))
#         sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
#         plt.title(f"Correlation Matrix for Client: {client}, Age Group: {age_group}, Gender: {gender}")
#         plt.show()
#     else:
#         print(f"No numeric data available for Client: {client}")

from scipy.stats import zscore

grouped = df_cleaned.groupby('client')

df_cleaned['time_taken_seconds'].fillna(df_cleaned['time_taken_seconds'].mean(), inplace=True)

solved_data = df_cleaned[df_cleaned['is_Solved'] == 1]

# Calculate z-scores for time_taken_seconds and help_count in the filtered data
solved_data['z_time_taken_seconds'] = zscore(solved_data['time_taken_seconds'])
solved_data['z_help_count'] = zscore(solved_data['help_count'])

df_cleaned.loc[solved_data.index, 'z_time_taken_seconds'] = solved_data['z_time_taken_seconds']
df_cleaned.loc[solved_data.index, 'z_help_count'] = solved_data['z_help_count']

# Initialize an empty DataFrame to store results
results = pd.DataFrame()

# Calculate aggregates for each client
for client, data in grouped:
    agg_data = data.groupby(['ageGroup_mapped', 'gender_mapped', 'course_mapped']).agg(
        mean_time_taken=('time_taken_seconds', 'mean'),
        std_time_taken=('time_taken_seconds', 'std'),
        mean_help_count=('help_count', 'mean'),
        std_help_count=('help_count', 'std'),
        count_solved=('is_Solved', 'sum'),
        count_mean=('is_Solved', 'mean'),
        total_tasks=('is_Solved', 'count'),
        mean_z_time_taken=('z_time_taken_seconds', 'mean'),
```

```
        mean_z_help_count=('z_help_count', 'mean')
    ).reset_index()

    agg_data['client'] = client
    results = pd.concat([results, agg_data], ignore_index=True)

# Display results
print(results)

#Total task counts for each group

grouped_age = results.groupby('ageGroup_mapped')

age_df = pd.DataFrame()

for ageGroup_mapped, data in grouped_age:
    agg_data = data.groupby(['ageGroup_mapped']).agg(
    Tasks_count = ('total_tasks', 'sum')
    )

    agg_data['ageGroup_mapped'] = ageGroup_mapped
    age_df = pd.concat([age_df, agg_data], ignore_index = True)

Total_tasks_18_20 = age_df.loc[age_df['ageGroup_mapped'] == '18-20',
'Tasks_count'].values[0]
Total_tasks_21_28 = age_df.loc[age_df['ageGroup_mapped'] == '21-28',
'Tasks_count'].values[0]

plt.figure(figsize=(12, 8))
sns.barplot(x='ageGroup_mapped', y='Tasks_count', data=age_df)
plt.title('Total_tasks_attempted by each age_group')
plt.xlabel('Age Group')
plt.ylabel('Total_tasks_attempted')
plt.xticks(rotation=45)

# Annotate the plot with percentage difference
x1, x2 = 0, 1  # positions of the two bars
y1, y2 = Total_tasks_18_20, Total_tasks_21_28  # heights of the two bars

plt.text(x1, y1 + 1, f'{Total_tasks_18_20:.2f}', ha='right', va='bottom', color='blue')
plt.text(x2, y2 + 1, f'{Total_tasks_21_28:.2f}', ha='right', va='bottom', color='orange')
plt.show()

grouped_gender = results.groupby('gender_mapped')

gender_df = pd.DataFrame()

for gender_mapped, data in grouped_gender:
    agg_data = data.groupby(['gender_mapped']).agg(
    Tasks_count = ('total_tasks', 'sum')
    )

    agg_data['gender_mapped'] = gender_mapped
    gender_df = pd.concat([gender_df, agg_data], ignore_index = True)

Total_tasks_male = gender_df.loc[gender_df['gender_mapped'] == 'manlich',
'Tasks_count'].values[0]
Total_tasks_female = gender_df.loc[gender_df['gender_mapped'] == 'weiblich',
'Tasks_count'].values[0]

plt.figure(figsize=(12, 8))
sns.barplot(x='gender_mapped', y='Tasks_count', data=gender_df)
plt.title('Total_tasks_attempted by each gender')
plt.xlabel('gender')
plt.ylabel('Total_tasks_attempted')
```

```
284. plt.xticks(rotation=45)
285.
286. # Annotate the plot with percentage difference
287. x1, x2 = 0, 1  # positions of the two bars
288. y1, y2 = Total_tasks_male, Total_tasks_female  # heights of the two bars
289.
290. plt.text(x1, y1 + 1, f'{Total_tasks_male:.2f}', ha='right', va='bottom', color='blue')
291. plt.text(x2, y2 + 1, f'{Total_tasks_female:.2f}', ha='right', va='bottom', color='orange')
292. plt.show()
293.
294. #Average time taken by ech age group and gender
295.
296. grouped_solved = solved_data.groupby('client')
297.
298. results_solved = pd.DataFrame()
299.
300. for client, data in grouped_solved:
301.     agg_data = data.groupby(['ageGroup_mapped', 'gender_mapped', 'course_mapped']).agg(
302.         mean_time_taken=('time_taken_seconds', 'mean'),
303.         mean_help_count=('help_count', 'mean'),
304.         mean_z_time_taken=('z_time_taken_seconds', 'mean'),
305.         mean_z_help_count=('z_help_count', 'mean')
306.     ).reset_index()
307.
308.     agg_data['client'] = client
309.     results_solved = pd.concat([results_solved, agg_data], ignore_index=True)
310.
311.
312. # Calculate means for each age group
313. mean_time_18_20 = results_solved[results_solved['ageGroup_mapped'] == '18-
20']['mean_time_taken'].mean()
314. mean_time_21_28 = results_solved[results_solved['ageGroup_mapped'] == '21-
28']['mean_time_taken'].mean()
315.
316. # Compute the percentage difference
317. percentage_diff = ((mean_time_18_20 - mean_time_21_28) / mean_time_21_28) * 100
318.
319. # Bar plot for time taken by age group and gender
320. plt.figure(figsize=(12, 8))
321. sns.barplot(x='ageGroup_mapped', y='mean_time_taken', data=results_solved)
322. plt.title('Time Taken by Age Group')
323. plt.xlabel('Age Group')
324. plt.ylabel('Mean Time Taken (seconds)')
325. plt.xticks(rotation=45)
326.
327. # Annotate the plot with percentage difference
328. x1, x2 = 0, 1  # positions of the two bars
329. y1, y2 = mean_time_18_20, mean_time_21_28  # heights of the two bars
330.
331. plt.text(x1, y1 + 1, f'{mean_time_18_20:.2f}', ha='right', va='bottom', color='blue')
332. plt.text(x2, y2 + 1, f'{mean_time_21_28:.2f}', ha='right', va='bottom', color='orange')
333. plt.text((x1 + x2) / 2, max(y1, y2) + 3, f'{percentage_diff:.2f}% difference', ha='center',
va='bottom', color='black', fontsize = 30)
334. plt.show()
335.
336. # Calculate means for each gender group
337. mean_time_male = results_solved[results_solved['gender_mapped'] ==
'weiblich']['mean_time_taken'].mean()
338. mean_time_female = results_solved[results_solved['gender_mapped'] ==
'manlich']['mean_time_taken'].mean()
339.
340. # Compute the percentage difference
341. percentage_diff_gender = ((mean_time_male - mean_time_female) / mean_time_female) * 100
342.
343. plt.figure(figsize=(12, 8))
```

```
344. sns.barplot(x='gender_mapped', y='mean_time_taken', data=results_solved)
345. plt.title('Time Taken by Gender')
346. plt.xlabel('Gender')
347. plt.ylabel('Mean Time Taken (seconds)')
348. plt.xticks(rotation=45)
349.
350. # Annotate the plot with percentage difference
351. x1, x2 = 0, 1  # positions of the two bars
352. y1, y2 = mean_time_male, mean_time_female  # heights of the two bars
353.
354. plt.text(x1, y1 + 1, f'{mean_time_male:.2f}', ha='right', va='bottom', color='blue')
355. plt.text(x2, y2 + 1, f'{mean_time_female:.2f}', ha='right', va='bottom', color='orange')
356. plt.text((x1 + x2) / 2, max(y1, y2) + 2, f'{percentage_diff_gender:.2f}% difference',
ha='center', va='bottom', color='black', fontsize = 30)
357. plt.show()
358.
359. #Average help taken by each group
360.
361. # Calculate means for each age group
362. mean_help_18_20 = results_solved[results_solved['ageGroup_mapped'] == '18-
20']['mean_help_count'].mean()
363. mean_help_21_28 = results_solved[results_solved['ageGroup_mapped'] == '21-
28']['mean_help_count'].mean()
364.
365. # Compute the percentage difference
366. percentage_diff = ((mean_help_18_20 - mean_help_21_28) / mean_help_21_28) * 100
367.
368. # Bar plot for time taken by age group and gender
369. plt.figure(figsize=(12, 8))
370. sns.barplot(x='ageGroup_mapped', y='mean_help_count', data=results_solved)
371. plt.title('Help Taken by Age Group')
372. plt.xlabel('Age Group')
373. plt.ylabel('Mean Help count (per question)')
374. plt.xticks(rotation=45)
375.
376. # Annotate the plot with percentage difference
377. x1, x2 = 0, 1  # positions of the two bars
378. y1, y2 = mean_help_18_20, mean_help_21_28  # heights of the two bars
379.
380. plt.text(x1, y1, f'{mean_help_18_20:.2f}', ha='right', va='bottom', color='blue')
381. plt.text(x2, y2, f'{mean_help_21_28:.2f}', ha='right', va='bottom', color='orange')
382. plt.text((x1 + x2) / 2, max(y1, y2) + 0.3, f'{percentage_diff:.2f}% difference',
ha='center', va='bottom', color='black', fontsize = 30)
383. plt.show()
384.
385. # Calculate means for each gender group
386. mean_help_female = results_solved[results_solved['gender_mapped'] ==
'weiblich']['mean_help_count'].mean()
387. mean_help_male = results_solved[results_solved['gender_mapped'] ==
'manlich']['mean_help_count'].mean()
388.
389. # Compute the percentage difference
390. percentage_diff_gender = ((mean_help_female - mean_help_male) / mean_help_female) * 100
391.
392. plt.figure(figsize=(12, 8))
393. sns.barplot(x='gender_mapped', y='mean_help_count', data=results_solved)
394. plt.title('Help Taken by Gender')
395. plt.xlabel('Gender')
396. plt.ylabel('Mean Help count (per question)')
397. plt.xticks(rotation=45)
398.
399. # Annotate the plot with percentage difference
400. x1, x2 = 0, 1  # positions of the two bars
401. y1, y2 = mean_help_female, mean_help_male  # heights of the two bars
402.
```

```
plt.text(x1, y1, f'{mean_help_female:.2f}', ha='right', va='bottom', color='blue')
plt.text(x2, y2, f'{mean_help_male:.2f}', ha='right', va='bottom', color='orange')
plt.text((x1 + x2) / 2, max(y1, y2) + 0.2, f'{percentage_diff_gender:.2f}% difference',
ha='center', va='bottom', color='black', fontsize = 30)
plt.show()

#Hypotheses answering

# Logistic Regression

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
import statsmodels.api as sm

df_encoded = pd.get_dummies(df_cleaned, columns=['ageGroup_mapped', 'gender_mapped'],
drop_first=True)

df_encoded.rename(columns={'ageGroup_mapped_21-28': 'ageGroup_mapped',
'gender_mapped_weiblich': 'gender_mapped'}, inplace=True)

# Define features and target
# X = df_encoded[[col for col in df_encoded.columns if col.startswith('ageGroup_mapped') or
col.startswith('gender_mapped')]]
X = df_encoded[['ageGroup_mapped', 'gender_mapped']]
y = df_encoded['is_Solved']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

X_train = X_train.astype(int)

y_train = y_train.astype(int)

# Make predictions
y_pred = model.predict(X_test)
y_pred_proba = model.predict_proba(X_test)[:, 1]

# Evaluate the model
cm = confusion_matrix(y_test, y_pred)
print(f"confusion matrix: {cm}")
print(classification_report(y_test, y_pred))

# Plot the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Plot the ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='grey', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
```

```
464. plt.legend(loc="lower right")
465. plt.show()
466.
467. # Fit the logistic regression model using statsmodels
468. X_sm = sm.add_constant(X_train)
469. logit_model = sm.Logit(y_train, X_sm)
470. result = logit_model.fit()
471.
472. # Print the summary of the model
473. print(result.summary())
474.
475. # Extract coefficients and intercept
476. coefficients = model.coef_[0]
477. intercept = model.intercept_[0]
478.
479. print(f"Coefficients: {coefficients}")
480. print(f"Intercept: {intercept}")
481.
482. # Form the logistic regression equation
483. equation = f"logit(P) = {intercept} + ({coefficients[0]} * ageGroup) + ({coefficients[1]} *
gender)"
484. print("Logistic Regression Equation:")
485. print(equation)
486.
487. from sklearn.tree import plot_tree
488. from sklearn.model_selection import train_test_split
489. # from sklearn.ensemble import RandomForestClassifier
490. from sklearn.tree import DecisionTreeClassifier
491. from sklearn import tree
492. from sklearn.metrics import confusion_matrix, classification_report, roc_curve,
roc_auc_score, accuracy_score
493. from sklearn.model_selection import GridSearchCV
494.
495. X = df_encoded[['ageGroup_mapped', 'gender_mapped']]
496. y = df_encoded['is_Solved']
497.
498. # Create and train the Random Forest model
499. # rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
500. # rf_model.fit(X_train, y_train)
501.
502. X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
503.
504. dt = DecisionTreeClassifier(random_state=42)
505. dt.fit(X_train, y_train)
506.
507. # Hyperparameter tuning using GridSearchCV
508. param_grid = {
509.     'criterion': ['gini', 'entropy'],
510.     'max_depth': [None, 10, 20, 30],
511.     'min_samples_split': [2, 10, 20],
512.     'min_samples_leaf': [1, 5, 10]
513. }
514.
515. grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=5, n_jobs=-1,
scoring='accuracy')
516. grid_search.fit(X_train, y_train)
517.
518. # Best model
519. best_dt = grid_search.best_estimator_
520.
521. # Make predictions
522. y_pred = best_dt.predict(X_test)
523. y_pred_prob = best_dt.predict_proba(X_test)[:, 1]
524.
525. # Confusion matrix and classification report
```

```
accuracy = accuracy_score(y_test, y_pred)
cm_rf = confusion_matrix(y_test, y_pred)
cr_rf = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print(f"confusion matrix: {cm_rf}")
print(cr_rf)

# ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
roc_auc = roc_auc_score(y_test, y_pred_prob)

# Visualize one of the trees in the Random Forest
# plt.figure(figsize=(20, 10))
# tree_num = 0  # Index of the tree to plot
# plot_tree(rf_model.estimators_[tree_num], feature_names=X.columns, filled=True, rounded=True, class_names=True)
# plt.show()

plt.figure(figsize=(20,10))
tree.plot_tree(best_dt, feature_names=['ageGroup_mapped', 'gender_mapped'], class_names=['Not Solved', 'Solved'], filled=True, label='none', impurity=False, node_ids=False, proportion=True, rounded=True, precision=2, fontsize=26)
plt.show()

# Get feature importances
importances = best_dt.feature_importances_
features = X.columns

for feature, importance in zip(features, importances):
    print(f'{feature}: {importance}')

# Create a DataFrame for visualization
feature_importance_df = pd.DataFrame({'feature': features, 'importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='importance', ascending=False)

# Plot feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x='importance', y='feature', data=feature_importance_df)
plt.title('Feature Importance')
plt.show()

print(feature_importance_df)

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Encode categorical variables
df_encoded = pd.get_dummies(df_cleaned, columns=['ageGroup_mapped', 'gender_mapped'], drop_first=True)

df_encoded.rename(columns={'ageGroup_mapped_21-28': 'ageGroup_mapped', 'gender_mapped_weiblich': 'gender_mapped'}, inplace=True)

# Define features and target
X = df_encoded[['ageGroup_mapped', 'gender_mapped']]
y = df_encoded['is_Solved']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=42)
```

```
586.
587. # Standardize the features
588. scaler = StandardScaler()
589. X_train = scaler.fit_transform(X_train)
590. X_test = scaler.transform(X_test)
591.
592. # Initialize and train the SVM
593. svm_model = SVC(kernel='linear', random_state=42)
594. svm_model.fit(X_train, y_train)
595.
596. # Make predictions
597. y_pred = svm_model.predict(X_test)
598.
599. # Evaluate the model
600. print("Confusion Matrix:")
601. print(confusion_matrix(y_test, y_pred))
602. print("\nClassification Report:")
603. print(classification_report(y_test, y_pred))
604. accuracy = accuracy_score(y_test, y_pred)
605. print("\nAccuracy:", accuracy)
606.
607. # Plot confusion matrix
608. conf_matrix = confusion_matrix(y_test, y_pred)
609. sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues")
610. plt.title('Confusion Matrix')
611. plt.xlabel('Predicted')
612. plt.ylabel('Actual')
613. plt.show()
614.
615. # Extracting coefficients for linear SVM
616. coefficients = pd.DataFrame(svm_model.coef_, columns=['ageGroup_mapped', 'gender_mapped'])
617. print(coefficients)
618.
619. # Assuming a binary classification with two features
620. def plot_decision_boundary(X, y, model):
621.     h = .02  # step size in the mesh
622.     x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
623.     y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
624.     xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
625.     Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
626.     Z = Z.reshape(xx.shape)
627.     plt.contourf(xx, yy, Z, cmap=plt.cm.coolwarm, alpha=0.8)
628.     plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.coolwarm, edgecolors='k')
629.     plt.xlabel('Age Group')
630.     plt.ylabel('Gender')
631.     plt.title('SVM Decision Boundary')
632.     plt.show()
633.
634. plot_decision_boundary(X_test, y_test, svm_model)
635.
636. #gridsearch cv
637.
638. #Making suggestions
639.
640. grouped_unsolved = df_cleaned.groupby('id')
641.
642. results_unsolved = pd.DataFrame()
643.
644. for id, data in grouped_unsolved:
645.     agg_data = data.groupby(['ageGroup_mapped', 'gender_mapped']).agg(
646.         total_tasks=('is_Solved', 'count'),
647.         True_count=('is_Solved', 'sum'),
648.         False_count=('is_Solved', lambda x: (x == 0).sum()),
649.         False_count_mean=('is_Solved', lambda x: (x == 0).mean())
650.     ).reset_index()
```

```
651.
652.     agg_data['id'] = id
653.     results_unsolved = pd.concat([results_unsolved, agg_data], ignore_index=True)
654.
655. prescriptive = df_cleaned.groupby('id')
656.
657. results_sug = pd.DataFrame()
658.
659. for id, data in prescriptive:    #whatever parameter is used in groupby should be used as
the first parameter in the for loop here
660.
661.     agg_data = data.groupby(['ageGroup_mapped', 'gender_mapped']).agg(
662.         total_tasks=('is_Solved', 'count'),
663.         True_count=('is_Solved', 'sum'),
664.         False_count=('is_Solved', lambda x: (x == 0).sum()),
665.         False_count_mean=('is_Solved', lambda x: (x == 0).mean())
666.     ).reset_index()
667.
668.     solved_data = data[data['is_Solved'] == 1]
669.
670.     # Aggregate mean_time_taken and mean_help_count only for solved_data
671.     solved_agg_data = solved_data.groupby(['ageGroup_mapped', 'gender_mapped']).agg(
672.         mean_time_taken=('time_taken_seconds', 'mean'),
673.         mean_help_count=('help_count', 'mean')
674.     ).reset_index()
675.
676.     # Merge the two aggregated DataFrames on 'ageGroup_mapped' and 'gender_mapped'
677.     merged_agg_data = pd.merge(agg_data, solved_agg_data, on=['ageGroup_mapped',
'gender_mapped'])
678.
679.     # Add the 'id' column
680.     merged_agg_data['id'] = id
681.
682.     # Concatenate the merged data to the results DataFrame
683.     results_sug = pd.concat([results_sug, merged_agg_data], ignore_index=True)
684.
685. threshold = pd.DataFrame()
686.
687. threshold = results_sug.groupby('id').agg(
688.     mean_false_count=('False_count', 'mean'),
689.     mean_time_taken=('mean_time_taken', 'mean'),
690.     mean_help_count=('mean_help_count', 'mean')
691. ).reset_index()
692.
693. merged_df = pd.merge(results_sug, threshold, on='id', suffixes=('', '_mean'))
694.
695. # Calculate the thresholds
696. false_count_threshold = merged_df['mean_false_count']
697. time_taken_threshold = merged_df['mean_time_taken_mean']
698. help_count_threshold = merged_df['mean_help_count_mean']
699.
700. # Apply the conditions
701. merged_df['suggestion_needed'] = (
702.     (merged_df['False_count'] > false_count_threshold) |
703.     (merged_df['mean_time_taken'] > time_taken_threshold) |
704.     (merged_df['mean_help_count'] > help_count_threshold)
705. ).astype(int)
706.
707. df_suggestions = merged_df[merged_df['suggestion_needed'] == 1]
708.
709. grouped = df_suggestions.groupby(['ageGroup_mapped', 'gender_mapped',
'id']).size().reset_index(name='counts')
710.
711. # Create a pivot table for the bar plot
```

```
712. pivot = grouped.pivot_table(index=['ageGroup_mapped', 'gender_mapped'], columns='id',
values='counts', fill_value=0)
713.
714. # Plotting
715. pivot.plot(kind='bar', stacked=True, figsize=(14, 8))
716.
717. # Add title and labels
718. plt.title('Math Topics Needing Help by Age Group and Gender')
719. plt.xlabel('Age Group and Gender')
720. plt.ylabel('Count of Math Topics Needing Help')
721.
722. # Show the plot
723. plt.xticks(rotation=45, ha='right')
724. plt.tight_layout()
725. plt.legend(title='Math Topics')
726. plt.show()
```