Learning Style Identification in MOOC Environment using Machine Learning

Phase II project report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Muhammed Adnan Palath(FIT19CS077)
Rohit S Mathews(FIT19CS096)
Roshan Raj S(FIT19CS099)
Saheen Usman M(FIT19CS102)

Federal Institute of Science And Technology (FISAT)®
Angamaly, Ernakulam

Affiliated to

APJ Abdul Kalam Technological University
CET Campus, Thiruvananthapuram
May 2023

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)

Mookkannoor(P.O), Angamaly-683577

CERTIFICATE

This is to certify that the report entitled "**Learning Style Identification in MOOC Environment using Machine Learning**" is a bonafide record of the project submitted by **Muhammed Adnan Palath (FIT19CS077), Rohit S Mathews (FIT19CS096), Roshan Raj S (FIT19CS099), Saheen Usman M (FIT19CS102)**, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (B.Tech) in Computer Science & Engineering during the academic year 2022-23.

Dr. Paul P Mathai	**Ms.Lakshmi S**	**Dr. Jyothish K John**
Project Coordinator	Project Guide	Head of the Department

ABSTRACT

This project focuses on predicting student learning styles in Massive Open Online Course (MOOC) environments using machine learning techniques. For this, the Felder Silverman learning style model (FSLSM) is adopted since it is one of the most commonly used models in technology-enhanced learning. The study utilizes a diverse dataset comprising various attributes related to student behavior and engagement on an online platform. Preprocessing techniques are employed to handle missing data, normalize features, address class imbalances, and ensure the dataset's suitability for training and evaluation.

To accurately identify learning styles, four classifiers, namely k-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM), are trained and evaluated. These classifiers serve as the foundation for the FSLSM (Feature Selection and Learning Style Model) developed in this project. Additionally, an ensemble model combining SVM and RF is explored to assess its performance compared to individual classifiers. Each classifier undergoes appropriate training methodologies and hyperparameter tuning to optimize predictive performance.

The project contributes to the field of educational data mining by providing insights into predicting student learning styles in MOOC environments using machine learning. The findings emphasize the importance of ensemble models to improve prediction accuracy and identify future research and development opportunities in this domain.

Contribution by Author

Working on the learning style prediction project has been an incredibly rewarding experience that has expanded my knowledge and skills in machine learning. my contribution was instrumental in several key areas, ensuring the accuracy and effectiveness of the predictive model. Specifically, I played a vital role in acquiring the dataset, preprocessing it, balancing the dataset using SMOTE, training the model, tuning its hyperparameters, ensemble modeling, and evaluation. I played a crucial role in ensuring the quality and comprehensiveness of the dataset, as well as preparing it for analysis through meticulous preprocessing. Collaborating with the team and sharing insights has been invaluable, as it has not only deepened my understanding of the subject matter but also opened my eyes to the power of teamwork in achieving project goals. This project has provided me with an opportunity to apply my expertise, make meaningful contributions, and witness the transformative impact of machine learning in the field of education. Finally, I actively participated in evaluating the model's performance and contributed to the final year project report with comprehensive documentation of the entire process.

Muhammed Adnan Palath

Contribution by Author

By actively participating in the learning style prediction system project, I played a vital role in advancing the knowledge within the field of machine learning. Possessing a profound understanding of the fundamental principles and algorithms, I explored inventive approaches and techniques to enhance the accuracy and efficacy of the prediction system. Through extensive research and rigorous experimentation, I contributed to the development of novel methodologies and insights that pushed the boundaries of machine learning in the specific context of learning style prediction. My endeavors not only enriched the existing knowledge base but also paved the way for future research in the field. I played a role in acquiring the preprocessed data required for the project. This involved meticulous data collection, ensuring the dataset's quality and relevance. Also, I implemented the K-nearest neighbors (KNN) algorithm, to model the learning style prediction system. Through careful parameter tuning and experimentation, I achieved optimal results, enhancing the system's performance and prediction accuracy. My contribution in data acquisition, data scaling, and application of the KNN algorithm was pivotal in the successful development and effectiveness of the learning style prediction system. Furthermore, my commitment to documentation facilitated clear communication and knowledge sharing among team members, promoting efficient work flows and facilitating future project maintenance. Overall, my contributions in these general aspects were instrumental in creating an environment conducive to success and maximizing the project's outcomes.

Rohit S Mathews

Contribution by Author

I played a crucial role in formulating the problem statement and research objectives of our project.I identified the need to develop an accurate model for predicting individual learning styles based on various educational and behaviour patterns.I determined the appropriate data collection techniques and data preprocessing steps to reliability and validity of the study.I also played a crucial role in the data preprocessing techniques such as cleaning,normalization and feature extraction to prepare the data for analysis.

I also made significant contributions to feature engineering by identifying relevant features that could potentially influence learning styles.I also implemented the decision tree algorithm and used hyperparameter tuning to enhance the performance metrics.I was also involved in identifying the best model for learning style prediction by comparing the accuracy obtained by various learning algorithms such as Decision tree,Random forest,SVM and KNN.I also conducted statistical analysis to identify significant factors or features contributing to precision accuracy.I was also involved in interpreting the results obtained from the learning style prediction model.

This whole process was an enriching process for me in terms of gaining sufficient knowledge regarding the project and gave an insight on how to work efficiently within a team environment.In the documentation process,i was involved in ensuring clarity, coherence and readability throughout the document,adhering to appropriate technical standards.I was also involved in creating visual aids,such as flowcharts in the documentation process.

Roshan Raj S

Contribution by Author

Engaging in the learning style prediction project has been an immensely rewarding journey, enriching my knowledge and honing my machine learning skills. My contributions played a pivotal role in ensuring the accuracy and efficacy of the predictive model across various critical aspects. I actively participated in acquiring the dataset, analysing the useful ones. By analysing the different data I understood various factors needed for the prediction. Additionally, I took a hands-on approach to model training, hyperparameter tuning, and implementing the final ensemble model for the prediction. Specifically I implemented SVM and SVM+RF ensembled model.
I actively engaged in team discussions and brainstorming sessions, offering valuable insights and ideas to enhance the project's direction and methodology. As challenges arose, I was quick to identify potential solutions and implement necessary adjustments. I also prioritized continuous learning and knowledge sharing within the team. I actively kept up-to-date with the latest research and advancements in machine learning, attended relevant course, and sharing valuable resources and insights with my teammates.

Overall, my contributions extended beyond the specific technical tasks, encompassing teamwork, adaptability, continuous learning, and documentation. By actively engaging in these general aspects, I helped create a collaborative and productive environment that fostered innovation, problem-solving, and effective project management.

Saheen Usman M

ACKNOWLEDGMENT

We are extremely thankful to God Almighty for showering his grace and blessings without which we would not have completed the project. We would like to take this opportunity to thank the Chairman of FISAT, **Mr. Shimith P R** and Principal, **Dr. Manoj George** and **Dr.Jyothish K John**, Head of the Department of Computer Science and Engineering for providing me with such an environment, where students can explore their creative ideas and support. We also like to pay our gratitude to **Dr. Paul P Mathai, Dr. Hema Krishnan** and **Ms. Lakshmi S** for the constant encouragement, support and for guiding us with patience and enthusiasm in all the stages. We would also like to express our sincere gratitude to all the faculties of the department of Computer Science And Engineering, FISAT. Also we sincerely thank our parents and friends for giving us moral support and encouragement in all possible ways.

Muhammed Adnan Palath
Rohit S Mathews
Roshan Raj S
Saheen Usman M

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

People take in and process information in different ways. A learning style is the method a person uses to learn. A style of learning refers to an individual's preferred way to absorb, process, comprehend and retain information. Styles influence how students learn, how teachers teach, and how the two interact. Style can be considered a "contextual" variable or construct because what the learner brings to the learning experience is as much a part of the context as are the important features of the experience itself.
The project aims to explore and identify the key factors that influence learning styles and develop a machine learning model to accurately predict them. Each learner has distinct and consistent preferred ways of perception, organization and retention. These learning styles are characteristic cognitive, effective, and physiological behaviors that serve as pretty good indicators of how learners perceive, interact with, and respond to the learning environment.
The methodology involves collecting diverse data-sets, conducting feature engineering, training the model using various algorithms, and evaluating its performance. The expected outcomes include a robust predictive model, insights into influential factors, validation of the model's effectiveness, and potential applications in personalized education.

Over the last decade, MOOCs have acquired considerable popularity as educational and learning environments. They represent a recent and innovative method in education that is redefining the limits of the teaching and learning landscape. Their characteristics, such as massiveness, openness, accessibility and flexibility, have interested all stakeholders, regardless of whether or not they are closely involved in education.
The learning style prediction project holds significant potential for personalized education. By accurately predicting learning styles, personalized instruction and tailored learning experiences can be provided to enhance engagement and knowledge retention. The project's outcomes can contribute to optimizing curriculum design, integrating the predictive model into adaptive learning platforms, and developing student support systems. Ultimately, this project aims to revolutionize educational practices by enabling educators to identify struggling students and provide targeted interventions based on their predicted learning styles. The implementation of this project has the potential to improve overall learning outcomes and promote effective and personalized education.
Recent advancements in machine learning techniques and big data analysis have created new opportunities for a better understanding of how learners

behave and learn in environments known for their massiveness and openness. The Project is about predicting the learners' learning styles based on their learning traces.

1.2 Problem Statement

In the rapidly evolving landscape of Massive Open Online Courses (MOOCs), understanding and accommodating individual learning styles is critical for providing personalized educational experiences. However, accurately predicting the learning style of a student in a MOOC environment based on their behavior and engagement patterns presents a complex challenge.

The problem includes identifying and collecting data that are related to student behaviour. Various types of data available for learning style prediction, but those are only applicable in offline learning. So collecting data related to online study is a significant problem here.

Selecting a learning style model that effectively includes the learning style in online learning. And finally the main problem at hand is to develop an advanced machine learning model that can effectively analyze and classify student data in MOOCs, enabling precise identification of learning styles and facilitating tailored instructional strategies to enhance learning outcomes.

1.3 Objective

The objective of this project is to explore and identify the key factors influencing an individual's learning style, develop an accurate predictive model using machine learning algorithms, evaluate its performance, and investigate the potential benefits of personalized learning based on predicted learning styles.

The project seeks to enhance the effectiveness of personalized education by leveraging learning style prediction to optimize learning outcomes, engagement, and overall educational experiences for learners.

The project will present the findings, methodology, implementation details, and analysis of the predictive model, along with recommendations for practical applications and future research in the field of personalized education.

Chapter 2

Related Works

1.A Predictive Model for the Identification of Learning Styles in MOOC Environments

In 2019 Brahim Hmedna, Ali El Mezouary, and Omar Baz [1] proposed a predictive model for the identification of learning styles in MOOC environments with machine learning techniques. This study aimed to investigate the possibility of forecasting a learner's learning preferences by analyzing the digital footprints they generate while engaging with a MOOC platform. Collected dataset from edx course was preprocessed by cleaning, feature extraction and normalization. The study employed an unsupervised clustering technique to group students based on their learning style preferences. The technique combined learning preferences for each dimension and determined their dominance to produce labeled datasets. Additionally, four machine learning methods, including decision tree, random forest, K-nearest neighbors, and neural network, were employed in the study. To ensure optimal performance, performed grid search to fine-tune the hyper-parameters for each technique. Additionally, utilized the learning curve approach to assess whether the models suffered from overfitting or underfitting. The study revealed that the decision tree model exhibited remarkable accuracy, with a rate of 98%, in predicting learning styles based on the three dimensions of the FSLSM.

2.Student Behaviour Analysis to Detect Learning Styles in Moodle Learning Management System

In 2020, Yunia Ikawati and M. Udin Harun Al Rasyid and Idris Winarno [2] proposed a model to detect learning style prediction in Moodle Learning Management System by analyzing Student Behaviour. In this research, a new approach for identifying learners' evolving and adaptive learning styles is proposed. This method is based on an analysis of student attitudes towards the LMS. The log data from moodle was collected and preprocess of dataset was done in four steps. Firstly, student behavior data extraction that focused on taking only essential log attribute that is helpful for identifying learning style.Attributes are full user name, event context, event name, and description. These attributes provide information about the actions taken by students on various items in Moodle. Next preprocessing stage is for behavior feature, In essence, the feature selection process involves the task identifying attributes within the logs file data that are associated with the dimensions of the FSLSM learning style. Then normalization was applied on the collected the data on student behavior and comparing it with the results of the ILS questionnaire, the data is organized to identify relationships. These relationships are then presented in a suitable format to create

Behavior Classification Rules. A classification rule is devised for learning styles, which takes into account student behavior and involves the mapping of features based on the dimensions of the Felder-Silverman Learning Style Model. The classification rules from the learning style table are applied to identify a pattern for arranging a dataset based on learning objects. These objects are then employed as attributes or features, aligned with the Felder-Silverman Learning Style Model. The Decision Tree and Ensemble Process, particularly the Gradient Boosted Tree, are employed for classifying the data. The Rapid Miner tool is used in the classification process, which employs 10 Fold Cross-Validation. The Gradient Boosted Tree ensemble technique is employed for classification, achieving an accuracy rate of 85.95%, surpassing that of the Decision Tree.Upon evaluating the classification effectiveness of both algorithms, it is found that Gradient Boosted Tree outperforms the Decision Tree with an accuracy of 85.95% while the latter yields 85.71%.

3.A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM

In 2022, Abdulaziz Salamah Aljaloud, Diaa Mohammed Uliyan, Adel Alkhalil ,magdy Abd Elrhman, Azizah Fhad Mohammed Alogali,yaser Mohammed Altameemi , Mohammed Altamimi ,and Paul Kwan [3] proposed a Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM. The study utilized CNN and LSTM to extract significant features from data and model the temporal dependencies of time series data. The data collection process initially involved gathering student data from Blackboard, which is a commonly used LMS for efficiently storing university student data. The chosen features comprised metrics like the total time spent on the course and the number of logins. The prediction model employed CNN for extracting time series data related to student features, while LSTM was used for performance prediction. This method utilized the time sequence of student data to improve the reliability of predictions. The prediction and training were executed using LSTM. It is important to consider that if the size of the CNN layers, filters, and LSTM batch size are enlarged, it may cause the CNN-LSTM model to become time-consuming. Furthermore, diverse feature selection methods were utilized to expose student performance. In conclusion, the multi-layer CNN-LSTM deep learning model presents an opportunity to improve learning effectiveness and augment computational power; nevertheless, it requires more time for training.

4.A Proposed Architectural Model for an Automatic Adaptive ELearning System Based on Users Learning Style

In 2014, Adeniran Adetunji and Akande Ademola [4] proposed a Architectural Model for an Automatic Adaptive E-Learning System Based on Users Learning Style. The objective of this project is to establish an e-learning system that can adjust to the individual learning preferences of each user. As the user engages with the material, the system will learn about their learning style, creating a two-way learning process where both the user and the system are learning simultaneously. The system includes User model,

domain model, and adaptation model. The study proposes an architecture consisting of three models: a domain model that structures knowledge on the subject matter, a learner's model that characterizes the learner's understanding, and an adaptation model that implements the adaptation rules. The architecture includes a feedback mechanism to detect changes in the learner's knowledge and learning preferences. The adaptation model then adjusts itself accordingly, facilitating an automatic and seamless learning experience for the learner.

5.E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification

In 2010, Aleksandra Klasnja-Milicevic, Boban Vesin , Mirjana Ivanovic, Zoran Budimac [5] developed a E-Learning personalization based on hybrid recommendation strategy and learning style identification. The e-learning system discussed in this work adjusts automatically to suit learners' preferences, routines, and abilities. The differentiation among learners is based on their prior knowledge, preferred learning methods, styles, and objectives. Experiments were conducted using an educational dataset to assess the system's performance. First, by running the experiment with a learning resource based on Felder and Soloman's ILS theory, several behavioural patterns in various learning style variables were discovered. Then, in each learning style, a frequent sequence of navigational patterns was found using the AprioriAll method. Following that, recommendations based on the collaborative filtering method were produced using these sequences. The study found that the use of the AprioriAll algorithm to mine frequent sequences in Web logs and test learners' learning preferences can improve the quality of an intelligent tutoring system and keep its recommendations up to date. This collaborative filtering approach can be used to achieve this.

6.A Fuzzy Model for Predicting Learning Styles using Behavioral Cues in an Conversational Intelligent Tutoring System

In 2013, Keeley Crockett, Annabel Latham, David Mclean, James O'Shea [6] proposed a Fuzzy Model for Predicting Learning Styles using Behavioural Cues in an Conversational Intelligent Tutoring System. The Conversational Intelligent Tutoring System's design involves two main components: the CITS and the Fuzzy Learning Styles Predictor. The predictor utilizes conversational cues and a fuzzy expert system to predict the learner's style of learning. Within CITS there is a Tutorial Knowledge Base comprises topic content and tutorial material, along with associated assessments. The Conversational Interface captures dialogue, which the Conversational Agent exchanges in natural language. The Agent accesses a database of tutorial Dialogue Scripts to produce responses based on input rules.The Fuzzy Learning Styles Predictor receives conversational cues extracted by the Behavior Knowledge Base. Meanwhile, the Controller utilizes information from the Student Model, such as the student's knowledge level, test scores, topics visited, and learning style. To create a generalized fuzzy learning styles predictor, an automated method was necessary to induce fuzzy rules from data sets that contained varying conversational cues. This study utilized

the concept of fuzzy decision tree forests to interpret the trees as sets of Fuzzy IF-THEN rules, resulting in a knowledge base with a fuzzy singleton for each learning style dimension. This approach has been shown to be successful in previous research, where fuzzy decision trees were used to build trees tailored to specific learning style dimensions. These trees can effectively manage both continuous and discrete input data.

7.AI Based Learning style prediction in online learning for primary education

In 2022, Bens Pardamean , Teddy Suparyanto, Tjeng Wawan Cenggoro ,digdo Sudigyo , And Andri Anugrahana [7] proposed a AI-Based Learning Style Prediction in Online Learning for Primary Education. The research involved two distinct stages: (1) an online learning session, and (2) AI modeling. In the online learning session, students were provided with six learning materials related to numbers, personalized to their visual, auditory, and kinesthetic learning styles. They rated these materials on a scale of 1 to 5, and the resulting data were compiled into a ranking matrix to train the AI. The AI modeling stage involved cleaning the data and splitting it into training and test sets. Hyperparameters , , and were optimized through a five-fold cross-validation process. The optimal configuration was then utilized to train the main model with the entire training set, which was evaluated on the test set using Root Mean Squared Error (RMSE) as the evaluation metric. The performance of the AI model was deemed satisfactory with an average RMSE of 0.9035 on a scale of 1 to 5. The model performed better than the typical MF-based model, with an RMSE value lower by 0.0313.

8.Identification of Learning Styles in Distance Education Through the Interaction of the Student With a Learning Management System

In 2020, Roberto Douglas da Costa , Gustavo Fontoura de Souza, Thales Barros de Castro, Ricardo Alexsandro de Medeiros Valentim, and Aline de Pinho Dias [8] proposed a model for Identification of Learning Styles in Distance Education Through the Interaction of the Student With a Learning Management System. The proposal presented in this paper suggests a correlation between artificial intelligence methods and the principles of Learning Styles (LS). This study adopts the hypothetico-deductive method to explore the research question and find a solution through hypothetical investigations. The work is divided into five stages, beginning with a Literary Review (LR) in the first stage. The second stage involves collecting data through the CHAEA-32 questionnaire (Honey-Alonso Questionnaire on Learning Styles) filled out by the participating students. In the third stage, the data from the questionnaires is analyzed to identify the predominant Learning Styles of each student.In the fourth stage, SQL queries are executed in the LMS database to identify students' behavior while accessing resources and the LMS. The fifth and final stage is the delivery of results, where a neural network is utilized to classify students' standard behaviors based on their known learning styles.

9.Automatic Detection of Learning Styles on Learning Management Systems using Data Mining Technique

In 2016, Samina Rajper, Noor A. Shaikh, Zubair A. Shaikh and Ghulam Ali Mallah [9] proposed a model for automatic detection of Learning Styles on Learning Management Systems using Data Mining Technique. In order to establish a link between classroom learning styles and E-learner activities on the LMS, a survey of E-learners was necessary. In order to acquire this information, a group of individuals was chosen from an internet-based institution in Pakistan, where students enrolled in computer science courses were taking part in e-learning. The survey provided valuable initial data to achieve the research objectives. In this study, a Bayesian Network (BN) Data Mining technique was employed to map classroom learning styles to the E-learning environment using a large survey dataset. The incorporation of Learning Style (LS) models into E-learning systems is crucial, and the use of a BN, an acyclic graph capable of graphically representing uncertain facts for imprecise solutions, provides significant value. Following is the basic equation of BN.

$$p(Cj \mid d) = \frac{p(d \mid Cj)\,p(Cj)}{p(d)}$$

Where
$p(Cj \mid d)$ = probability of instance 'd' being in class 'Cj'
$p(Cj)$ = probability of class 'Cj' occurrence
$p(d)$ = probability of occurrence of instance 'd'.

The survey data results were processed using data mining software, which generated Conditional Probability Tables (CPT) for each learning style based on the attributes. These probabilities are updated as students interact with the LMS, performing activities that continuously inform the BN inference mechanism of the students' learning styles.

10.Predicting Students' Learning Styles Using Regression Techniques

In 2022, Mohammad Azzeh, Ahmad Mousa Altamimi, Mahmoud Albashayreh [10] proposed a model to predict Students' Learning Styles Using Regression Techniques. For the study, 72 students were randomly selected from higher education institutes, and their learning styles were identified using the VARK inventory questionnaire, which includes 16 questions related to preferred learning and teaching methods. The questionnaire results were imported into an Excel file, and each answer was represented as a binary vector. The dataset was preprocessed and divided into four matrices, one for each learning style, containing the probabilities of the learning styles and the selected learning style label. The study developed models for prediction in both regression and classification tasks, employing diverse algorithms such as Multi-Layers Perceptron Neural Network (NN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (kNN). The output for regression models was probabilities, and for classification models, it was learning style labels. Finally, the results for

each learning style were aggregated.

11.Prediction Learning Style Based on Prior Knowledge for Personalized Learning

In 2018, MS.Hasibuan and LE.Nugroho and PI.Santosa [11] developed a model to Predict Learning Styles Based on Prior Knowledge for Personalized Learning. Prior knowledge refers to the knowledge and skills that learners possess, which serve as the basis for determining their learning styles. There are four levels of prior knowledge, including Knowledge of Fact, Knowledge of Meaning, Integration of Knowledge, and Application of Knowledge, which are closely related to learning styles. In this study, the Weight Cosine Coefficient (WCC) is used to ensure that learners' answers are consistent with their prior knowledge. The WCC algorithm compares the answers given by the machine with those given by the teacher, using a formula that incorporates the Reference Assessment (RA) and Student Assessment (SA). The resulting answer from the machine is then weighted according to a priority scale established by the teacher. This approach allows for an initial or pretest assessment of the learners' prior knowledge, which is mapped onto the levels of knowledge (LOK). The LOK is then used to determine the learners' VARK learning styles. The WCC Algorithm Model is teacher-defined and can be used to provide manual assessment, thereby increasing the accuracy of the assessment results.

12.LSBCTR: A Learning Style-Based Recommendation Algorithm

In 2020, Thayron C. H. Moraes, Itana Stiubiener, Juliana C. Braga and Edson P. Pimentel [12] proposed a Learning Style-Based Recommendation Algorithm, LSBCTR. The goal is to predict the primary learning objectives (LO) that align with a student's learning style (LS) based on the input variables I, J, and R. The R classification matrix is used to calculate similarity scores as follows: first, students are asked to provide their grades to identify relevant LO (J). Next, based on their LS, similarity scores between the student's LS (i) and each item (j) are calculated. The LSBCTR algorithm utilizes Pearson's correlation to determine the similarity between two LS and generates a set of LS scores for all users (S). These scores are then used to compute confidence weights.

Algorithm 1: LSBCTR Recommender Algorithm

Input: Items' LS, Rating matrix R
Output: LS Score of users for the S Items'

Initialize S to an empty list;
for *all* $i \in I$ **do**
  $P := \{j | R_{ij} = 1\}$;
  Initialize S_{ij} to 0;
  for $j = 1$ *to* $|P| - 1$ **do**
    $S_{ij} :=$ Pearson-Similarity$(i_{EA}, P_{j_{EA}})$;
  end
end
return S

13.A Conceptual Framework for Detecting Learning Style in an Online Education Using Graph Representation Learning

In 2020, Bello Ahmad Muhammad, Zhenqiang Wu, Hafsa Kabir Ahmad [13] proposed a Conceptual Framework for Detecting Learning Style in an Online Education Using Graph Representation Learning. This study focuses on identifying learning styles automatically through the analysis of learners' behavior data. The first step involves creating a bipartite graph based on the learner's activity data to establish a relationship between learners and learning resources. In step 2, graph representation learning (GRL) techniques are used to transform the high-dimensional graph into a low-dimensional vector space while preserving its structure. Next, in step 3, the FSLSM assessment tool is chosen to identify independent and general vertices for each dimension. In step 4, graph clustering algorithms are used to map the latent representation and the adopted learning styles theory. This step helps to identify groups of learners who share similar learning styles. In the final step (step 5), the latent representation's vertices are utilized to divide learners into distinct groups. If a cluster group displays a significant correlation with a specific set of learning style dimensions, then the corresponding vertex is included in the feature vector, which is utilized to determine a learner's learning style classification.

14.Fuzzy-logic based learning style prediction in e-learning using web interface information

In 2015, L Jegatha Deborah, R Sathiyaseelan ,S Audithan And P Vijayakumar [14] proposed a Fuzzy-logic based learning style prediction model in e-learning using web interface information. This study, introduced a model that utilizes MediaWiki e-learning servers as a foundation for disseminating E-Learning materials in various formats. The specific E-Learning server utilized in this model includes course materials for the C programming language in textual, audio, and video forms. Once the learner is authenticated, they can access any type of content available on the MediaWiki E-Learning server. The proposed model focuses on learners who prefer textual formats of course content. The model was evaluated and tested to determine the main dimension of Felder Silverman learning style preferences. To authenticate users and identify their learning style, learners are requested to provide their original profile information. The learners' learning styles were accurately classified based on their profile information and online web usage activity, which was carefully monitored and recorded for analysis. Various parameters were examined, including but not limited to the number of mouse movements made in the y-axis, the ratio of document length to time spent on a page, the ratio of image area to document length and scroll distance, and the frequency of document visits. The MediaWiki e-learning server was used as a platform for posting course content in various formats, including textual, audio, and video formats for C programming language. For a Fuzzy set A, that represents the learning styles of the learners is represented by

$$f(x;\sigma,c) = e^{-\frac{(x-c)^2}{2\sigma^2}}.$$

The width parameter σ and the center parameter c modify the membership function curve's width of the Fuzzy set A, which corresponds to the learning style based on the input value x. The parameter c in this function represents the mean of the membership function curve. To classify learners into different categories based on their learning style, the proposed model employs a symmetric Gaussian fuzzy membership function. The four categories include active, medium active, medium reflective, and reflective. By using the rule base and applying the specified symmetric Gaussian fuzzy membership function, the model can predict the appropriate learning style for each learner.

15.Automatic Student Modelling for Detection of Learning Styles and Affective States in Web Based Learning Management Systems

In 2019,Farman Ali Khan , Awais Akbar, Muhammad Altaf,shujaat Ali Khan Tanoli, And Ayaz Ahmad [15] proposed a automatic student modelling for detecting of learning styles and the affective states in web based learning management system. The proposed approach for automatic identification of learning styles and affective states can be segmented into two components. The first part focuses on determining the relevant preferences and behavior of learners, while the second part involves collecting and organizing data on these preferences and behavior to infer learning styles and affective states, respectively. When it comes to identifying preferred learning styles and learning behavior, the selection of patterns and features in learning management systems (LMSs) depends on two requirements. Firstly, the patterns must be relevant for identifying learning styles based on the FSLSM and affective states. Secondly, information on these patterns must be readily available in LMSs. This requires selecting features that are commonly integrated in most LMSs, tracked by most LMSs, and frequently used by course developers and teachers. Bayesian network is used for calculation of learning style.

16.Design and Usability Evaluation of Adaptive E-Learning Systems Based on Learner Knowledge and Learning Style

In 2015, Mohammad Alshammari , Rachid Anane , and Robert J. Hendley [16] developed adaptive e-learning systems based on learner knowledge and learning style. AdaptLearn is composed of three main components: the domain model, the learner model, and the adaptation model. The domain model is structured hierarchically and contains knowledge elements related to the application domain, which is a common representation for domain models in similar research. The learner model considers the learner's learning style and knowledge level, which are determined using a pre-test and maintained through test items based on the interaction between the learner and the system. The learning style is identified through the Felder-Silverman questionnaire. The adaptation model provides recommendations for instructional materials to learners based on their interaction goals, uti-

lizing information from both the learner model and the domain model. The adaptation model offers personalized learning paths and adaptive guidance based on the learner's knowledge level and learning style, which are constructed and updated accordingly. The output of the adaptation model is then transferred to the interface, and AdaptLearn provides examples of personalized learning paths.

17.Detecting Learning Style Using Hybrid Model

In 2016, M S Hasibuan and LE Nugroho [17] predicted learning styles using Hybrid Model. The detection of learning styles can be accomplished through two methods: literature-based and data-driven detection. The literature-based detection method involves analyzing the time learners spend on learning materials, comparing it with the specified time for the materials, and generating learning styles based on these calculations. Specifically, the time learners spend visiting the FOCEE website is calculated to determine their learning styles. The system will monitor and record learners' interactions with learning materials presented in various forms, including visual, audio, reading, and kinesthetic. During these interactions, the system will analyze log data of content and outline visits and interactions. The same analysis will be conducted to measure the duration of learners' visits to the outline and content. The visit duration will be compared to a predefined duration, and the time difference will be calculated. In the final step, the system will evaluate learners' responses to questions in the example and exercise sections. These three processes are conducted to determine learning styles, which will then be used to generate recommendations through the identification mechanism.

18.A Hybrid Machine Learning Approach to Predict Learning Styles in Adaptive E-Learning System

In 2019, Ouafae El Aissaoui , Yasser El Madani El Alami ,Lahcen Oughdir1, and Youssouf El Allioui [18] proposed a hybrid machine learning approach to predict learning styles in adaptive e-learning systems. After identifying the sequences of learners, the primary aim is to classify them into particular combinations of learning styles using the FSLSM. This will allow the labeled sequences to be used as a training set for predicting the learning style of a new sequence. The process begins by using a clustering algorithm, followed by a classification algorithm. The proposed approach employs the following two algorithms:
The K-means algorithm is used to assign a label to each sequence based on FSLSM. Learners' sequences are extracted from the log file using data mining techniques, and then transformed into a matrix with M rows representing M sequences and sixteen columns to record attribute values. The attributes of each sequence correspond to the sixteen Los presented and are utilized as input for the K-means algorithm.
The Naïve Bayes classifier is used to classify a new learner or a new sequence of an existing learner based on the FSLSM.

19.Student's Learning Style Detection using Tree Augmented Naive Bayes

In 2018, Ling Xiao Li and Siti Soraya Abdul Rahman [19] proposed a model that detects Student's learning style using tree augmented naive Bayes. The system initially has pre-set learning styles for the students. However, their learning styles will be updated according to their individual learning behaviors as they interact with the system. The model used to detect these learning styles is based on a tree augmented naive Bayesian network, which utilizes data mining techniques to extract information from the students' learning behaviors. These learning behaviors can include a variety of activities such as visiting forums, sending and receiving emails, watching videos, completing exercises, communicating with others, and more.

20.Use of Deep Multi-Target Prediction to Identify Learning Styles

In 2020, Everton Gomede, Rodolfo Miranda de Barros and Leonardo de Souza Mendes [20] proposed a model that identifies learning styles using deep multi-target prediction. Data on the students' behavior was gathered using a learning management system (LMS) that was custom-made for this study. The learning materials employed included content, outlines, self-assessments, exercises, quizzes, forums, questions, navigation, and examples. Used the Felder–Silverman questionnaire, an adaptation to collect each student's learning style. After normalizing artificial neural network (ANN) algorithm chosen for multi-target prediction.

Chapter 3

Design

The project is designed to predict the learning style of a student, in order to personalize their learning experience to be more interactive and exciting for them.

3.1 Design Methodologies

1. Data Collection: The design methodology begins with a comprehensive data collection process, ensuring the gathering of diverse educational data, including learner profiles, academic performance, learning activities, and self-reported learning preferences. This involves utilizing appropriate data sources, such as surveys, questionnaires, or educational platforms, to collect the necessary information.
2. Data Preprocessing: Following data collection, a meticulous data preprocessing phase is employed. This involves handling missing values, removing irrelevant features, and performing necessary data transformations and feature engineering techniques to ensure data quality and compatibility for further analysis.
3. Feature Extraction: After data preprocessing, feature extraction techniques are applied to identify and extract relevant features that capture the different dimensions of learning styles. These techniques involve analyzing the preprocessed data and selecting informative and discriminative features that contribute to accurate learning style prediction.
4. Data Balancing: To address any class imbalance issues within the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. This technique synthetically generates new instances of minority classes, ensuring a balanced dataset for improved model training and prediction accuracy.
5. Data Splitting: The balanced dataset is then split into an 80-20 ratio for training and testing, respectively. The larger portion, 80 percent, is utilized for training the predictive models, while the remaining 20 percent is reserved for evaluating their performance.
6. Model Training: Four machine learning algorithms, namely decision tree, random forest, K-nearest neighbors (KNN), and support vector machine (SVM), are employed for training predictive models. Each algorithm is trained on the preprocessed and balanced dataset, leveraging their respective strengths to capture the underlying patterns and relationships within the data.
7. Model Tuning: To optimize the performance of the trained models, hyperparameter tuning techniques are applied. This involves systematically adjusting the parameters of each model to achieve the best possible accuracy and generalization capabilities.
8. Model Ensemble: The two models with the highest accuracy, SVM and Random Forest, are selected for ensemble modeling. The predictions from these models are

combined using a voting or averaging approach, leveraging their complementary strengths to further enhance prediction accuracy.

9. Prediction and Output: The ensemble model is then used to predict the learning styles of new, unseen data. The final output of the system includes the predicted learning styles based on the ensemble of SVM and Random Forest models. This output can be presented to users in a clear and interpretable format or stored for further analysis and application in personalized education systems.

   By following these design methodologies, the learning style prediction project ensures a systematic and rigorous approach to learning style prediction.

3.2 System Architecture

The system architecture for the learning style prediction project employs a methodical approach, starting with comprehensive data collection and preprocessing to handle missing values, eliminate irrelevant features, and extract relevant learning style dimensions. The dataset is balanced using SMOTE, and an 80-20 split is performed for training and testing. Four powerful algorithms, including decision tree, random forest, KNN, and SVM, are employed for model training, followed by hyperparameter tuning to optimize their performance. The two most accurate models, SVM and Random Forest, are selected for ensemble modeling, where their predictions are combined using a voting or averaging approach. The ensemble model is then utilized to predict learning styles for unseen data, generating outputs that can be presented to users or stored for further analysis and utilization in personalized education systems. Overall, this system architecture aims to deliver accurate and efficient learning style prediction, enhancing personalized education and improving learning outcomes.

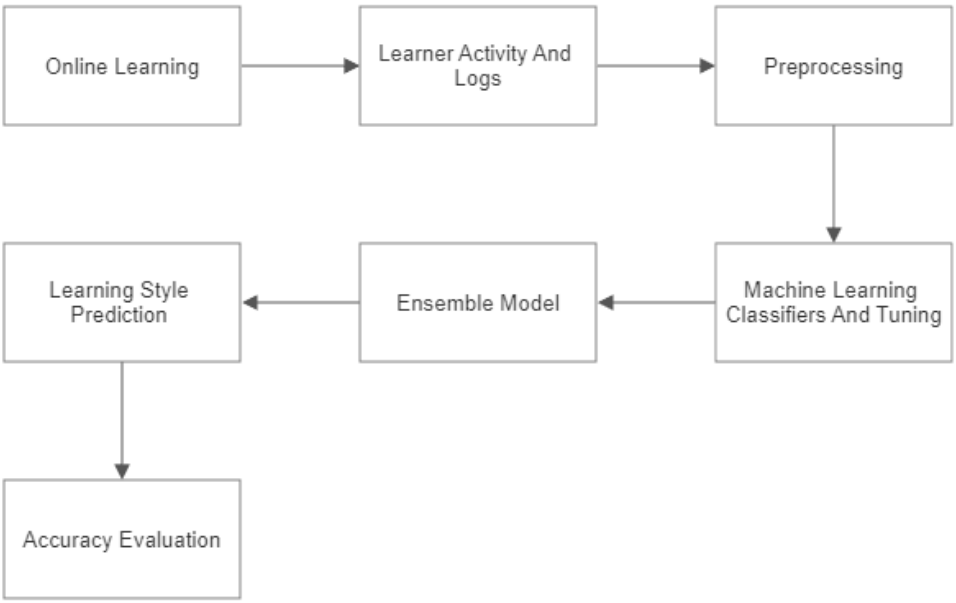

Figure 3.1: System Architecture

3.3 Flow Chart

The collected data undergoes a preprocessing phase to address missing values, eliminate irrelevant features, and apply essential transformations. Feature extraction techniques are then employed to identify and extract pertinent features that encapsulate the diverse dimensions of learning styles.

To counter potential class imbalance within the dataset, the architecture incorporates the Synthetic Minority Over-sampling Technique (SMOTE) to effectively balance the data, thus facilitating accurate learning style prediction. Following the data balancing step, the dataset is partitioned into an 80-20 ratio, allocating 80 percent for training the predictive model and reserving the remaining 20 percent for testing. The system leverages four robust algorithms, namely decision tree, random forest, K-nearest neighbors (KNN), and support vector machine (SVM), to train models on the preprocessed and balanced dataset.

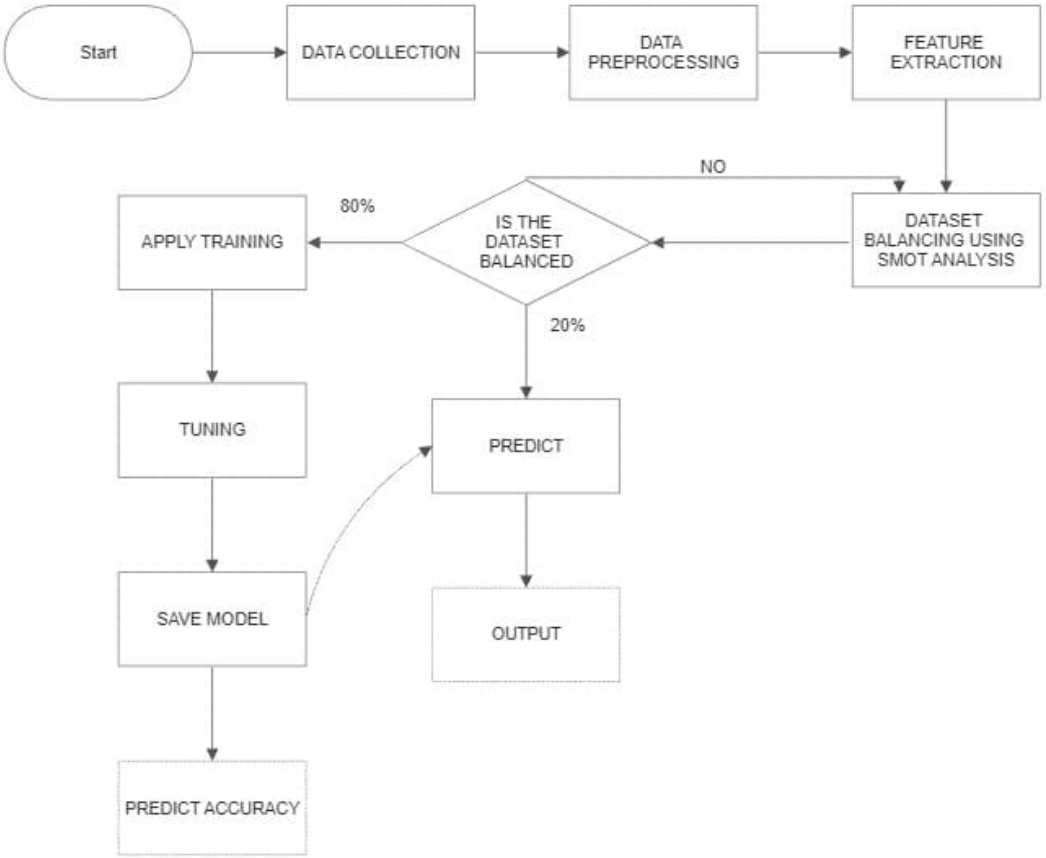

Figure 3.2: Flow Chart

Moreover, the system integrates model tuning to optimize the performance of each algorithm. By fine-tuning the models' hyperparameters, the aim is to achieve the highest levels of accuracy and generalization. The models are then evaluated using suitable metrics on the testing dataset, with the two models displaying the highest accuracy, SVM and Random Forest, being selected for ensemble modeling. Through a voting or averaging approach, the predictions of these models are combined, creating an ensemble model that further enhances the accuracy of learning style prediction.

In the final stage, the ensemble model is deployed to predict learning styles for unseen data. The system generates outputs presenting the predicted learning styles based on the ensemble of SVM and Random Forest models. These outputs can be effectively presented to users in a comprehensible format or stored for subsequent analysis and utilization within personalized education systems.

Chapter 4

Implementation

4.1 Implementation Details

The implementation of the learning style prediction project involves several key steps to ensure accurate and robust predictions. The project follows a systematic approach, starting from data collection and ending with model evaluation and ensemble prediction. The implementation details are as follows:

4.1.1 Data Collection

Gather a comprehensive dataset consisting of learner profiles, academic performance, learning activities, and self-reported learning preferences. Ensure data integrity, privacy, and compliance with ethical considerations.

4.1.2 Data Preprocessing

Clean the collected data by handling missing values, removing outliers, and normalizing or scaling numerical features. Perform necessary data transformations and encoding categorical variables to make the data suitable for further analysis.

4.1.3 Feature Extraction

Extract relevant features from the preprocessed data that capture the different dimensions of learning styles. Use techniques such as dimensionality reduction, principal component analysis (PCA), or domain-specific feature engineering methods to derive informative features.

4.1.4 Data Balancing

Apply the Synthetic Minority Over-sampling Technique (SMOTE) to address any class imbalance issues in the learning style labels. SMOTE generates synthetic samples of minority classes to balance the dataset and prevent biased predictions.

4.1.5 Dataset Splitting

Split the preprocessed and balanced dataset into an 80 percent training set and a 20 percent testing set. Ensure the split is stratified to maintain the distribution of different learning styles in both sets.

4.1.6 Model Training

Train models using four different algorithms: decision tree, random forest, K-nearest neighbors (KNN), and support vector machine (SVM). Implement and configure each algorithm using suitable libraries or frameworks. Train the models on the training set and tune their hyperparameters to optimize performance.

4.1.7 Model Evaluation

Evaluate the trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score on the testing set. Additionally, employ cross-validation techniques, such as k-fold cross-validation, to assess the models' generalization ability and mitigate overfitting.

4.1.8 Model Ensemble

Identify the two models with the highest accuracy, which in this case are SVM and Random Forest. Combine their predictions using a voting or averaging approach to create an ensemble model that leverages the strengths of both algorithms.

4.1.9 Ensemble Prediction

Apply the ensemble model to new, unseen data to predict the learning styles of individuals. Generate the output, which includes the predicted learning styles based on the ensemble of SVM and Random Forest models.

4.1.10 Model Evaluation and Comparison

Evaluate the performance of the ensemble model using appropriate metrics and compare it with individual model performances. Perform statistical analysis and interpret the results to assess the effectiveness and robustness of the learning style prediction system.

4.1.11 Documentation

Document the entire implementation process, including data collection methods, preprocessing techniques, feature extraction approaches, model training

details, hyperparameter tuning, evaluation metrics, and ensemble modeling. Clearly present the findings, limitations, and future directions of the project in a final year project report.

By following these implementation details, the learning style prediction project aims to provide a comprehensive and accurate system that can assist in personalizing education and improving learning outcomes.

4.2 Dataset

A dataset refers to a structured and curated collection of data that is organized for analysis and interpretation. It encompasses multiple data points or records, each representing specific observations, and contains variables or attributes that capture different aspects of the data. A dataset can originate from diverse sources, undergo preprocessing to ensure data quality, and include metadata for documentation purposes. By serving as the foundation for statistical analysis, data mining, and machine learning, datasets enable researchers and analysts to derive insights, discover patterns, and build models, facilitating evidence-based decision-making and advancements in various fields.

4.2.1 Dataset Description

Real dataset extrapolated from the Central University in Uttarakhand for the study. Which has 1000 rows and initially 26 features. And target variable learning-style: 0, 1, 2, 3.
Following are the initial dataset features

T_image	Time spent on viewing images.
T_video	Time spent on watching videos.
T_read	Time spent on reading textual content.
T_audio	Time spent on listening to audio.
N_excercise_after_read	Number of exercises attempted after reading.
N_excercise_after_graphic	Number of exercises attempted after viewing graphics.
T_hierarchies	Time spent on understanding hierarchical structures.
T_powerpoint	Time spent on studying PowerPoint presentations.
T_abstract	Time spent on studying abstract concepts.
T_concrete	Time spent on studying concrete examples.
T_result	Time spent on reviewing learning outcomes.
N_standard_questions_correct	Number of correctly answered standard questions.
N_creative_questions_correct	Number of correctly answered creative questions.
N_msgs_posted	Number of messages posted in forums or discussions.
N_excercises_visited	Number of exercises visited.
T_reading_in_forum	Time spent on reading forum posts.

T_solve_excercise	Time spent on solving exercises.
T_submit_assignment	Time spent on submitting assignments.
N_group_discussions	Number of group discussions participated in.
T_outlines	Time spent on reviewing outlines of course content.
Skipped_los	Whether learning objectives were skipped.
N_next_button_used	Number of times the "Next" button was used.
T_spent_in_session	Total time spent in a single study session.
N_questions_on_details	Number of questions asked about specific details.
N_questions_on_outlines	Number of questions asked about course outlines.

Features selected for application in project are
T_image, T_video, T_read, T_audio, T_hierarchies, T_powerpoint, T_concrete, T_result, N_standard_questions_correct, N_msgs_posted, T_solve_excercise, N_group_discussions, Skipped_los, N_next_button_used, T_spent_in_session, N_questions_on_details and N_questions_on_outlines

4.3 Libraries/Applications

In a machine learning project, several libraries and applications play a vital role in facilitating data manipulation, model development, evaluation, and deployment

4.3.1 SkLearn

Scikit-learn, commonly referred to as sklearn, is a widely-used open-source machine learning library in Python. It provides a comprehensive set of tools, algorithms, and utilities for various stages of a machine learning project. Sklearn offers a range of functionalities that are highly valuable in the development and deployment of machine learning models.

Model Selection

The model-selection module in the scikit-learn library (sklearn) provides essential functionalities for model evaluation and selection in a machine learning project.Here are the imported methods used,

1. train-test-split
2. GridSearchCV
3. cross-val-score

Neighbours

The scikit-learn library's neighbors module provides a set of powerful algorithms and tools for implementing various types of nearest neighbors-based learning in machine learning projects.Here are the imported methods used,

1. KNeighborsClassifier

Metrics

The sklearn.metrics module in the scikit-learn library provides a wide range of evaluation metrics for assessing the performance and accuracy of machine learning models. This module plays a crucial role in machine learning projects as it allows practitioners to quantitatively evaluate the effectiveness of their models and make informed decisions.Here are the imported methods used,

1. classification-report
2. accuracy-score
3. confusion-matrix
4. f1-score
5. ConfusionMatrixDisplay

Inspection

The sklearn.inspection module in scikit-learn provides useful tools for inspecting and interpreting machine learning models, allowing practitioners to gain insights into model behavior and performance. It offers several functions and classes that aid in understanding the internal workings of models, feature importances, and model evaluation.Here are the imported methods used,

1. permutation-importance

Tree

The "Tree" module in scikit-learn (sklearn) is a valuable component that provides various decision tree-based machine learning algorithms for classification and regression tasks. It offers a range of functionalities for building, training, and evaluating decision tree models.Here are the imported methods used,

1. DecisionTreeClassifier

Ensemble

The sklearn.ensemble module in the scikit-learn library provides a set of powerful ensemble methods for machine learning projects. Ensemble learning combines multiple individual models to create a more accurate and robust predictive model.Here are the imported methods used,

1. RandomForestClassifier
2. VotingClassifier

SVM

The scikit-learn (sklearn) library's SVM module provides an implementation of Support Vector Machines (SVM), a powerful algorithm used in machine learning projects. SVM is a supervised learning method that can be applied to both classification and regression tasks.Here are the imported methods used,

1. SVC

Preprocessing

The sklearn.preprocessing module in scikit-learn (sklearn) provides a wide range of functions and classes for data preprocessing in a machine learning project. It offers various techniques to transform and prepare the input data before training a machine learning model.Here are the imported methods used,

1. MinMaxScaler

Feature Selection

The scikit-learn library's "Feature Selection" module provides a set of techniques and algorithms that aid in selecting the most informative and relevant features from a given dataset. Feature selection plays a crucial role in machine learning projects by identifying the subset of features that contribute the most to the predictive power of a model, thereby improving its efficiency, interpretability, and generalization ability.Here are the imported methods used,

1. mutual-info-classif

Impute

The sklearn.impute module in scikit-learn provides methods for handling missing data in a machine learning project. Missing data is a common challenge in real-world datasets and can negatively impact the performance and accuracy of machine learning models.Here are the imported methods used,

1. SimpleImputer

4.3.2 Imblearn

The oversampling module in imblearn offers various algorithms and techniques to generate synthetic samples for the minority class, thus balancing the class distribution. These techniques aim to alleviate the bias towards the majority class, which can lead to poor performance in machine learning models.

Over-Sampling

This technique randomly selects samples from the minority class and duplicates them until the class distribution is balanced.

1. SMOTE

4.3.3 Matplotlib

Matplotlib is a powerful and widely-used data visualization library in the field of machine learning. It provides a flexible and comprehensive set of tools for creating various types of high-quality plots and visualizations, which are essential for analyzing data, understanding patterns, and effectively communicating results in a machine learning project.

4.3.4 Numpy

Numpy, short for Numerical Python, is a fundamental library in Python specifically designed for numerical and scientific computing tasks. It provides a powerful set of functions and tools that are crucial in various aspects of machine learning projects.

4.3.5 Pandas

Pandas is a powerful and widely used open-source library in Python that provides data manipulation and analysis tools. It offers data structures and functions that are essential for handling structured data, making it highly valuable in machine learning projects. In a professional context.

Chapter 5

Results

5.1 Sample

5.1.1 Feature Selection

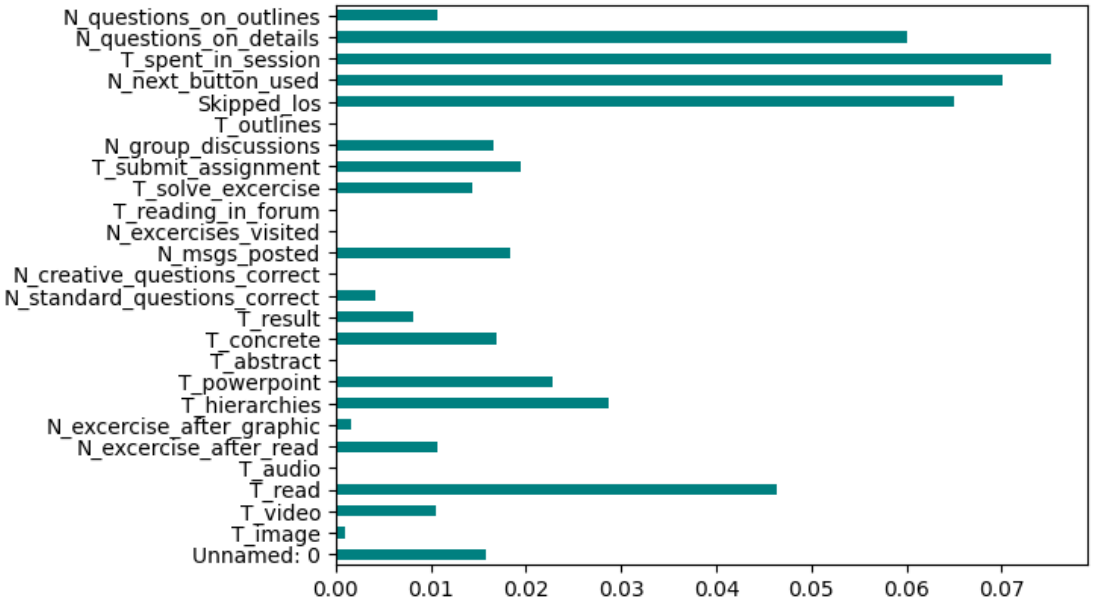

Figure 5.1: Mutual information Scores

5.1.2 KNN

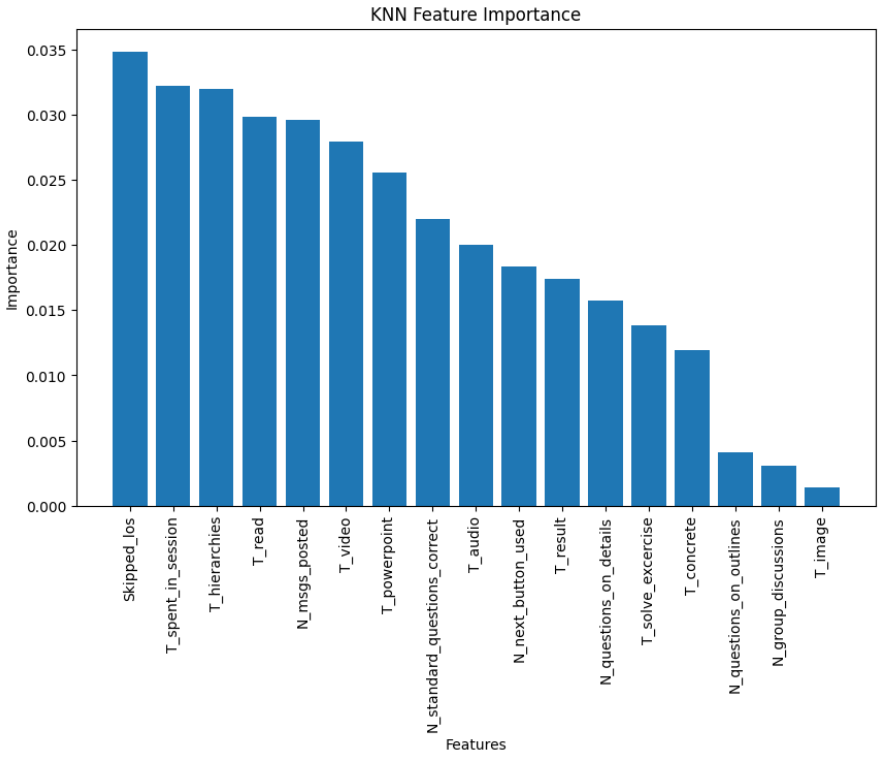

Figure 5.2: KNN feature importance

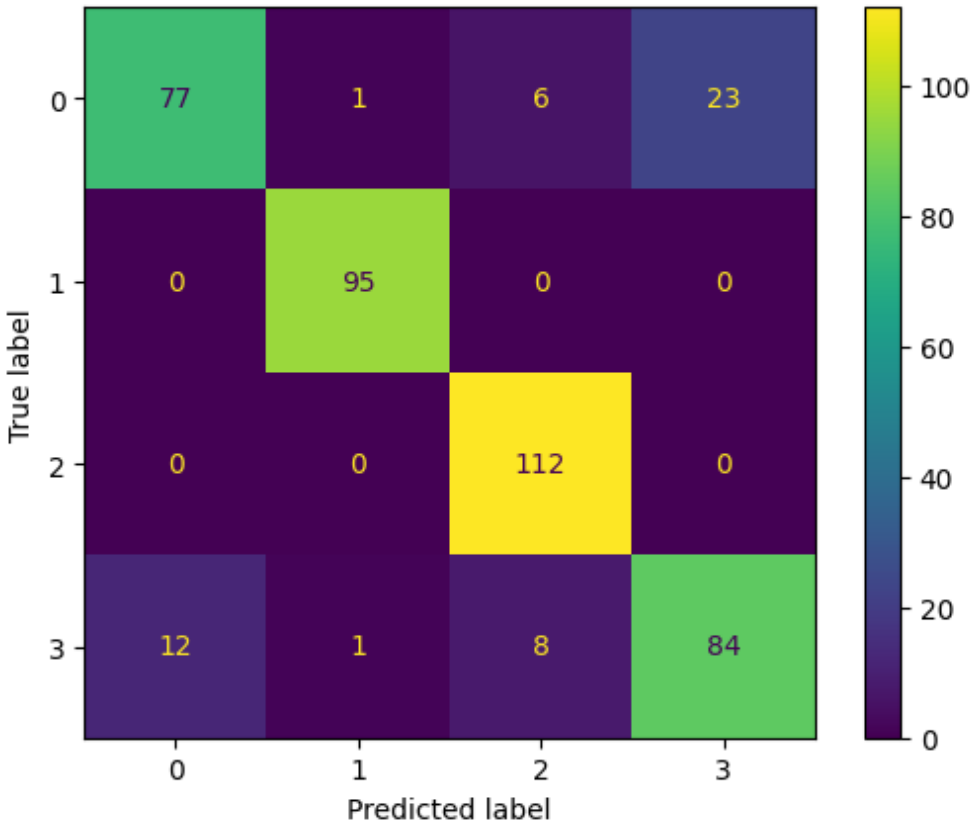

Figure 5.3: Confusion Matrix KNN

	precision	recall	f1-score	support
0	0.87	0.72	0.79	107
1	0.98	1.00	0.99	95
2	0.89	1.00	0.94	112
3	0.79	0.80	0.79	105
accuracy			0.88	419
macro avg	0.88	0.88	0.88	419
weighted avg	0.88	0.88	0.88	419

Figure 5.4: KNN Classification Report

```
Cross-validation scores: [0.83880597 0.84776119 0.84776119 0.84431138 0.88323353]
Mean accuracy: 0.8523746536777193
Accuracy: 85.24 %
Standard Deviation: 1.58 %
```

Figure 5.5: KNN Cross Validation

5.1.3 Decision Tree

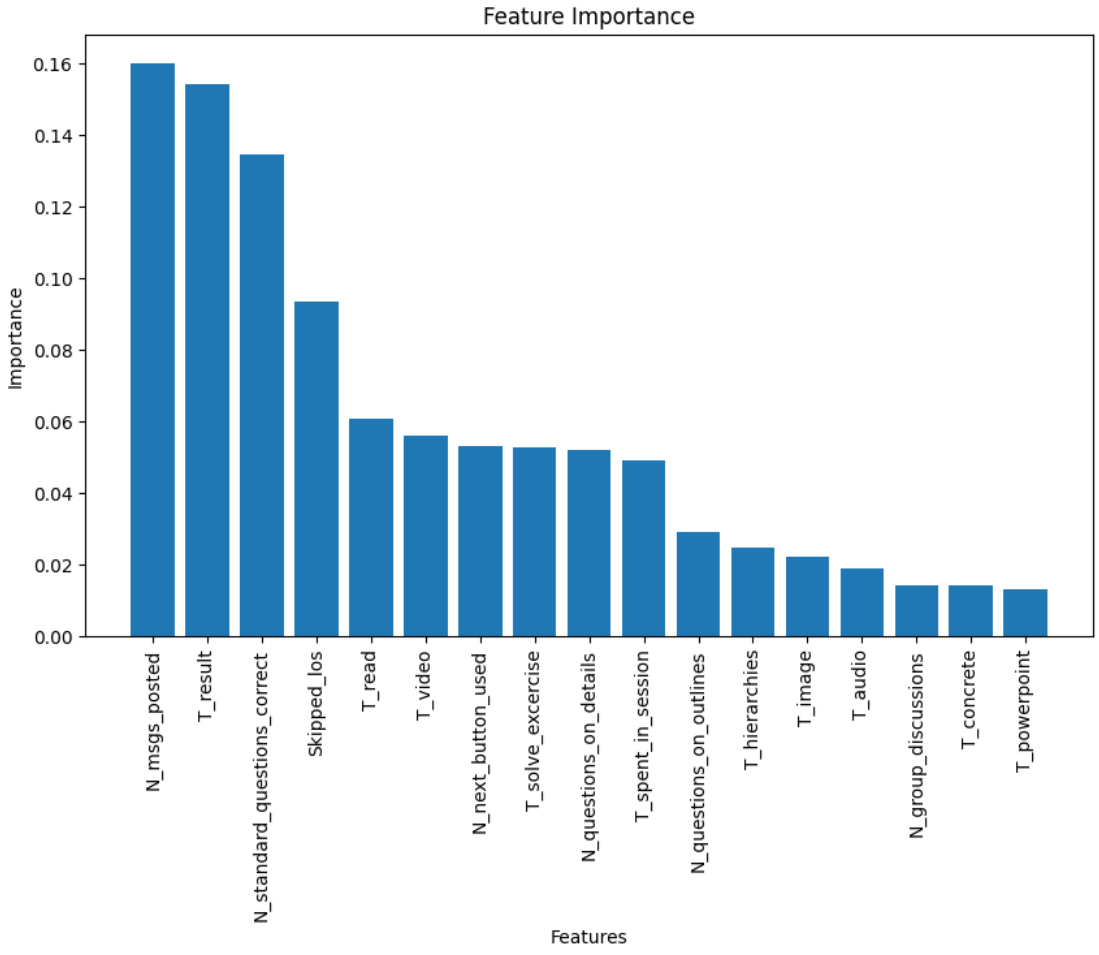

Figure 5.6: DT Feature importance

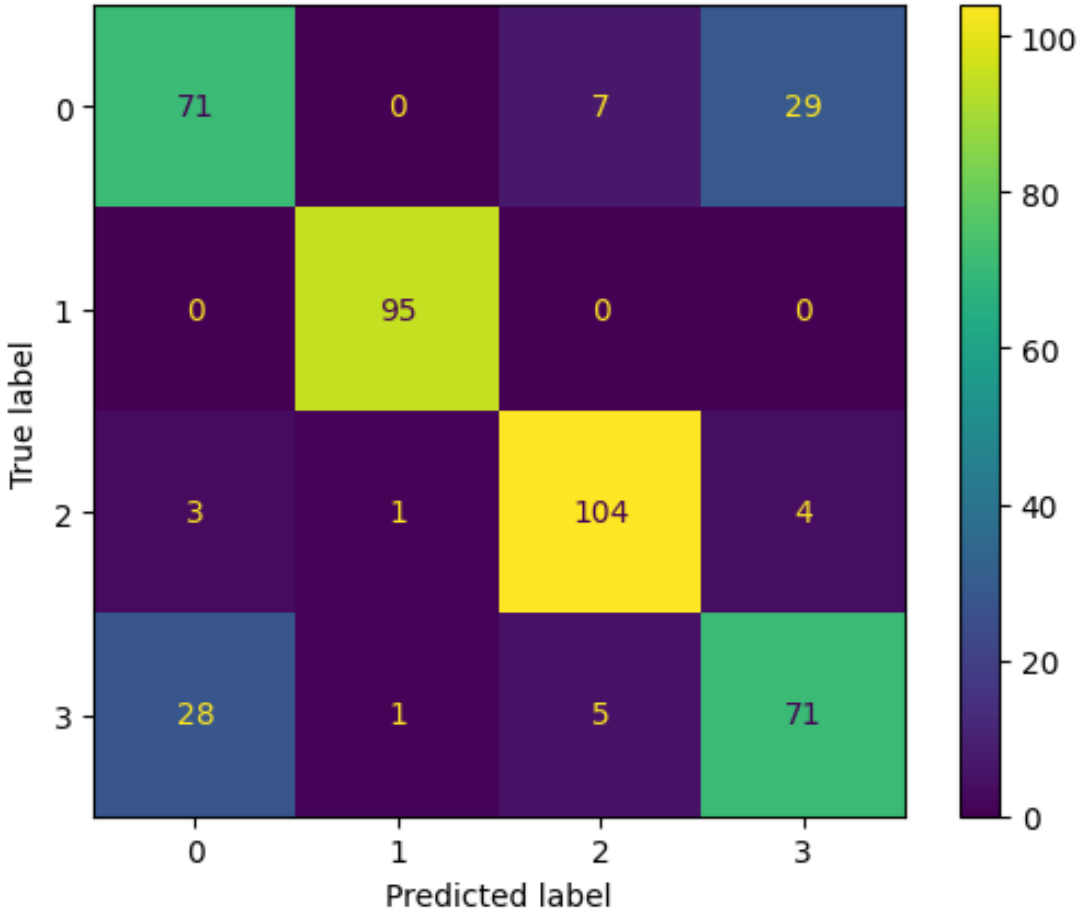

Figure 5.7: Confusion Matrix DT

	precision	recall	f1-score	support
0	0.70	0.66	0.68	107
1	0.98	1.00	0.99	95
2	0.90	0.93	0.91	112
3	0.68	0.68	0.68	105
accuracy			0.81	419
macro avg	0.81	0.82	0.82	419
weighted avg	0.81	0.81	0.81	419

Figure 5.8: DT Classification Report

```
Cross-validation scores: [0.80298507 0.80298507 0.81492537 0.81437126 0.81736527]
Mean accuracy: 0.8105264098668336
Accuracy: 81.05 %
Standard Deviation: 0.62 %
```

Figure 5.9: DT Cross Validation

5.1.4 Random Forest

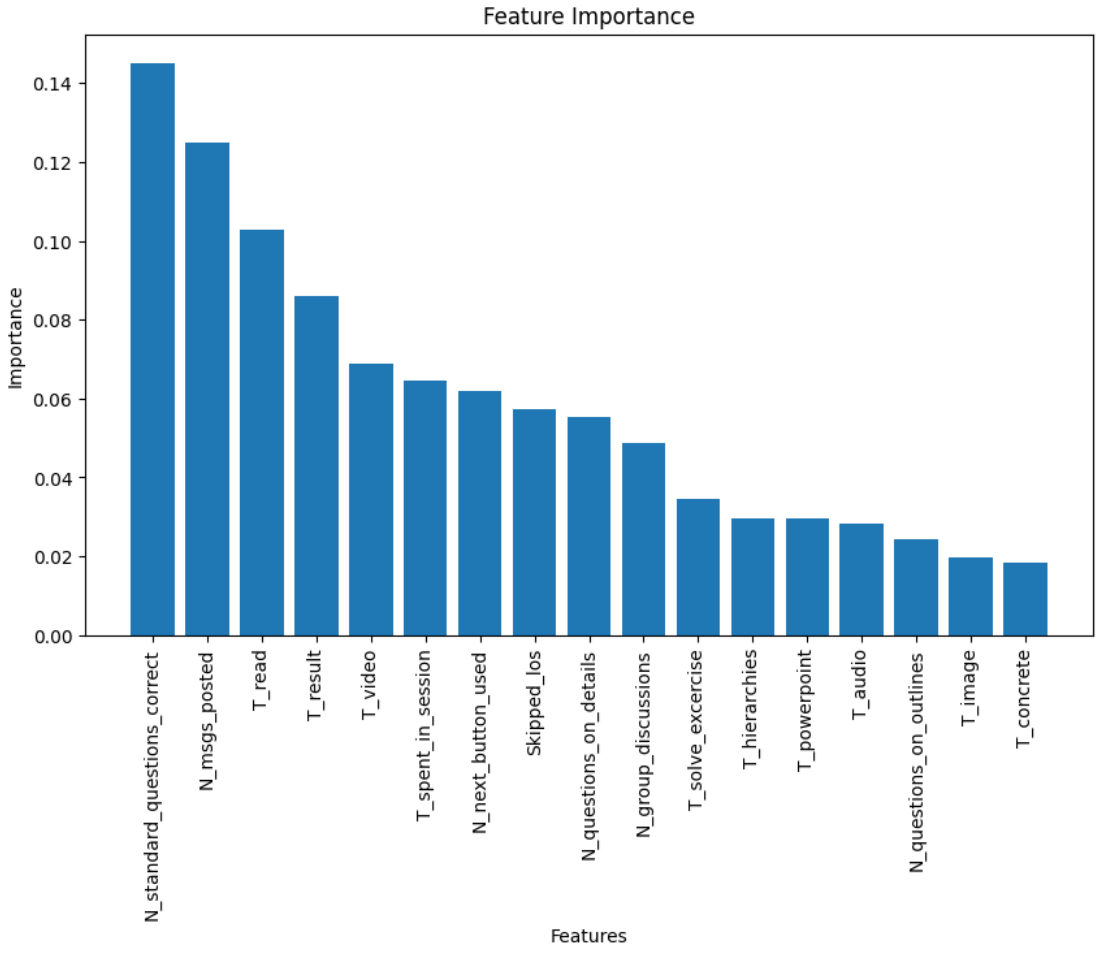

Figure 5.10: Feature Importance RF

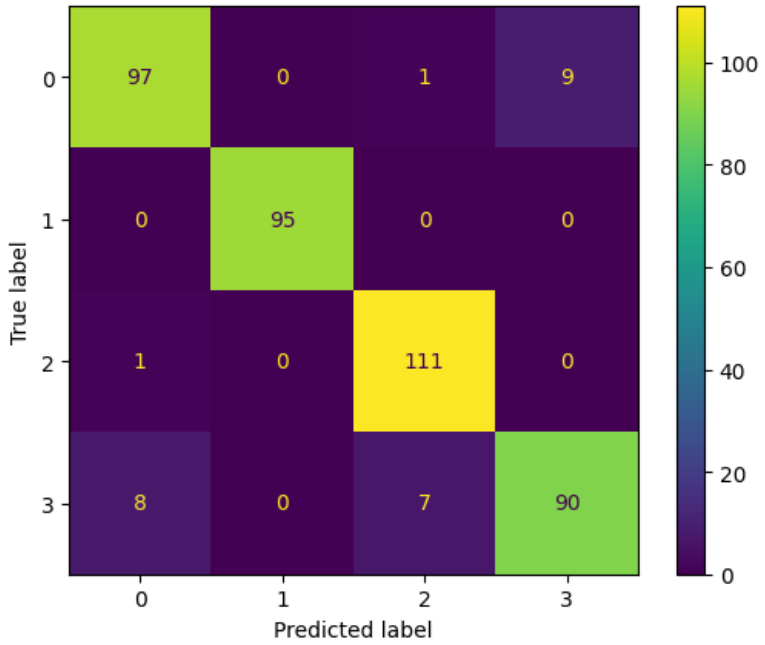

Figure 5.11: Confusion Matrix RF

	precision	recall	f1-score	support
0	0.90	0.90	0.90	107
1	1.00	1.00	1.00	95
2	0.93	0.99	0.96	112
3	0.90	0.84	0.87	105
accuracy			0.93	419
macro avg	0.93	0.93	0.93	419
weighted avg	0.93	0.93	0.93	419

Figure 5.12: RF Classification Report

```
Cross-validation scores: [0.92537313 0.91641791 0.92537313 0.93413174 0.9251497 ]
Mean accuracy: 0.9252891232460453
Accuracy: 92.53 %
Standard Deviation: 0.56 %
```

Figure 5.13: RF Cross Validation

5.1.5 SVM

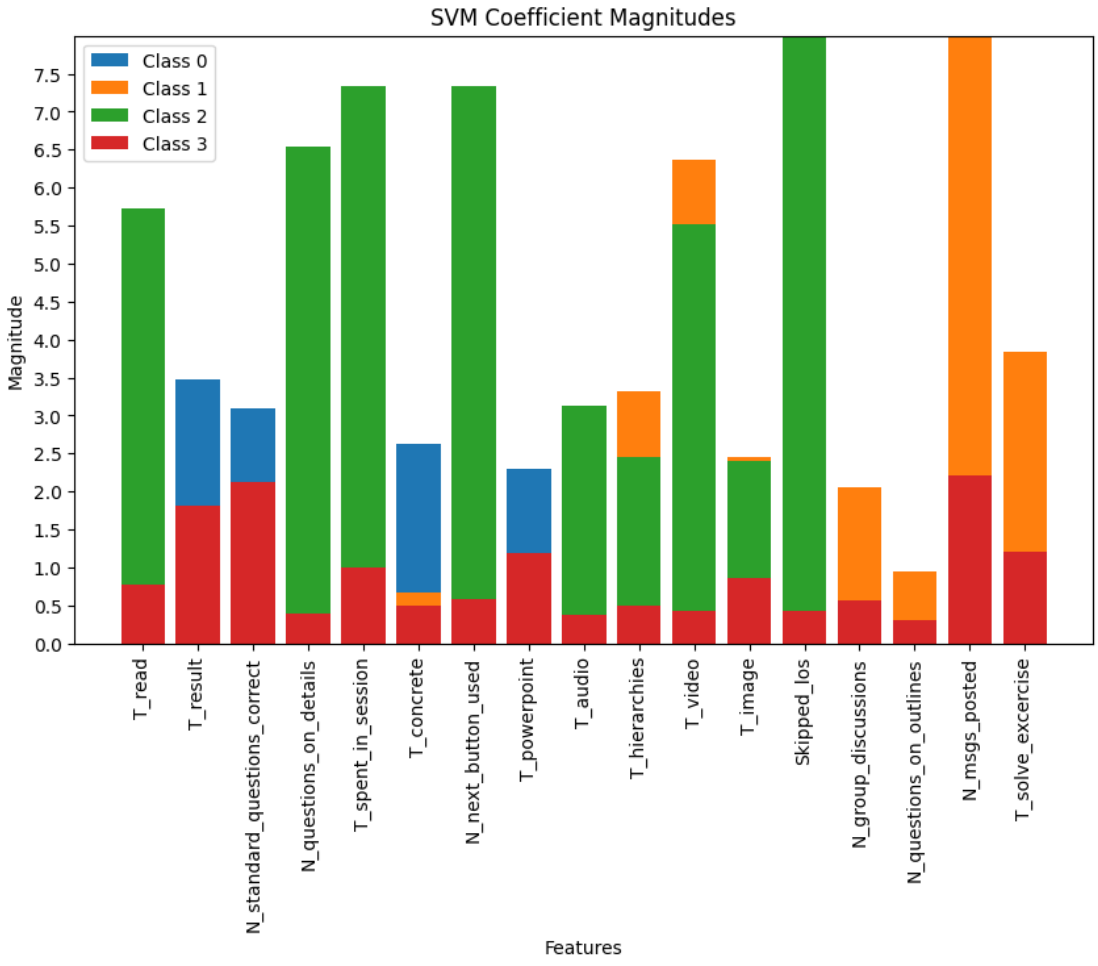

Figure 5.14: SVM Feature Contribution

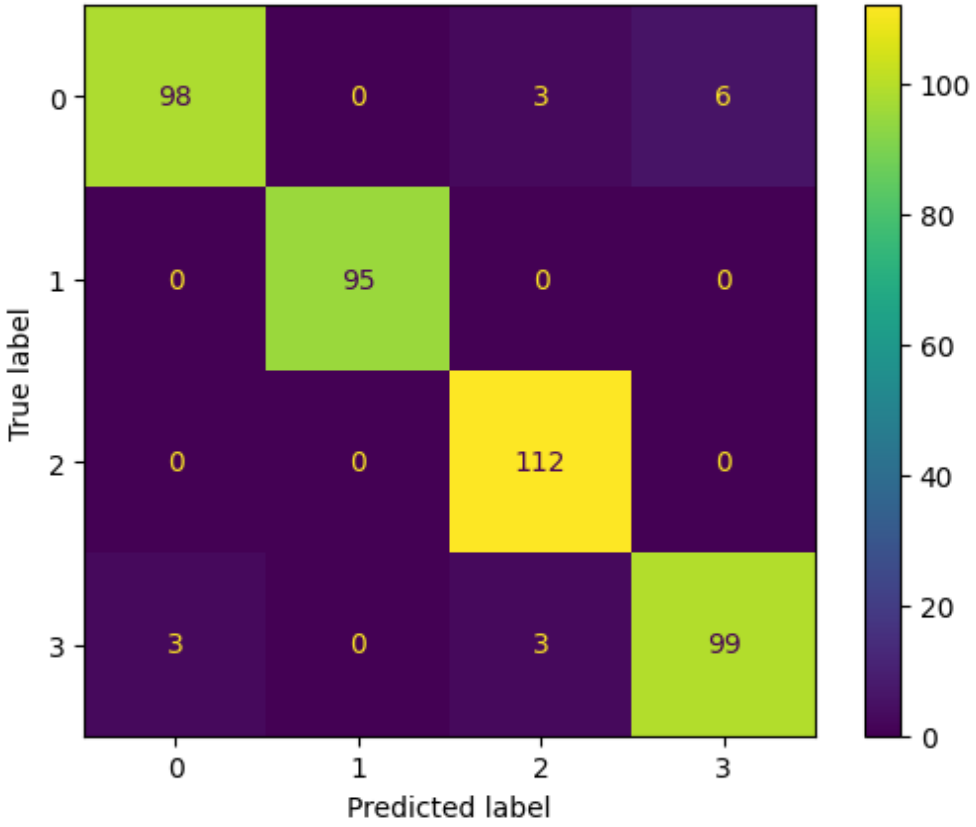

Figure 5.15: SVM Confusion Matrix

	precision	recall	f1-score	support
0	0.97	0.92	0.94	107
1	1.00	1.00	1.00	95
2	0.95	1.00	0.97	112
3	0.94	0.94	0.94	105
accuracy			0.96	419
macro avg	0.97	0.96	0.96	419
weighted avg	0.96	0.96	0.96	419

Figure 5.16: SVM Classification Report

```
Cross-validation scores: [0.95820896 0.95223881 0.96716418 0.96706587 0.97005988]
Mean accuracy: 0.9629475377603003
Accuracy: 96.29 %
Standard Deviation: 0.67 %
```

Figure 5.17: SVM Cross Validation

5.1.6 Ensemble Model SVM+RF

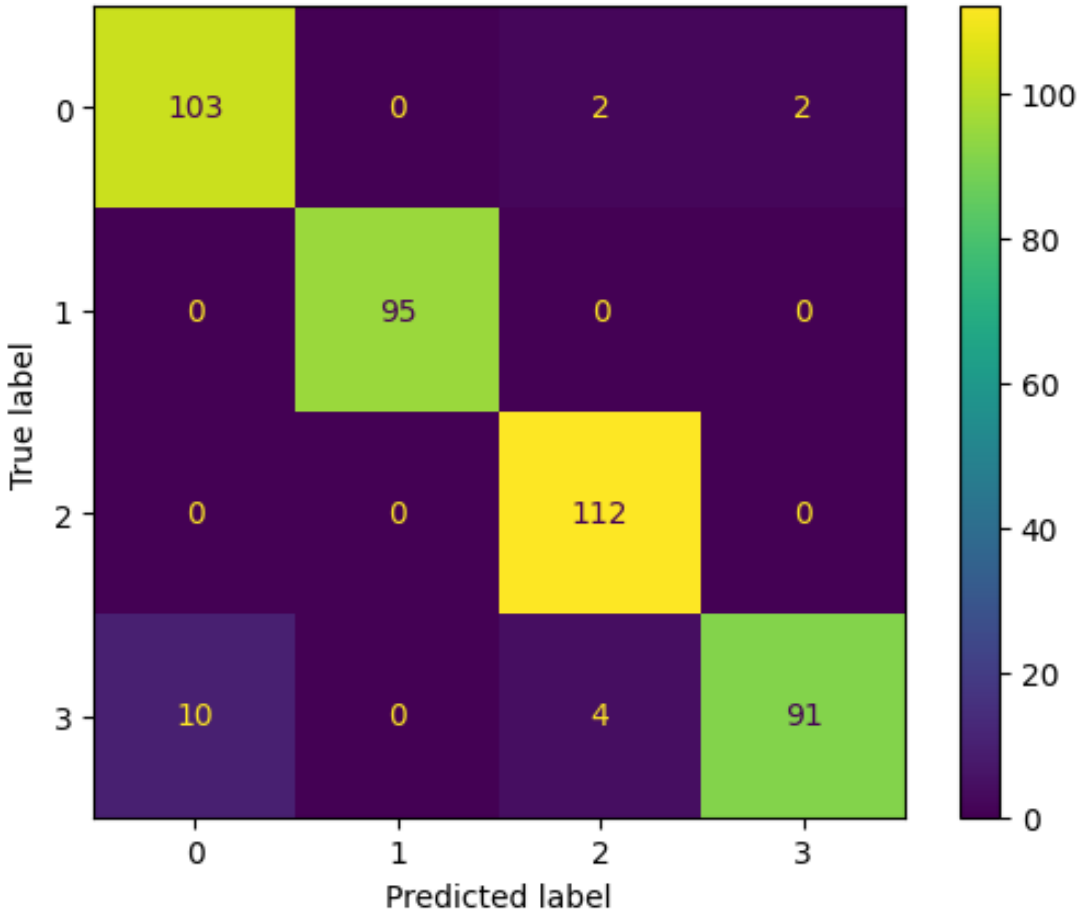

Figure 5.18: Ensemble Model Confusion Matrix

	precision	recall	f1-score	support
0	0.91	0.96	0.94	107
1	1.00	1.00	1.00	95
2	0.95	1.00	0.97	112
3	0.98	0.87	0.92	105
accuracy			0.96	419
macro avg	0.96	0.96	0.96	419
weighted avg	0.96	0.96	0.96	419

Figure 5.19: Ensemble Model Classification Report

```
Cross-validation scores: [0.92537313 0.94925373 0.93432836 0.95808383 0.96107784]
Mean accuracy: 0.9456233801054607
Accuracy: 94.56 %
Standard Deviation: 1.37 %
```

Figure 5.20: Ensemble Model Cross Validation

5.2 Comparison

The project introduces a novel approach by training four distinct algorithms, namely Decision Tree, Random Forest, k-Nearest Neighbors (KNN), SVM, and a ensemble model for learning style prediction. Through rigorous experimentation and evaluation, it is determined that the SVM and Random Forest algorithms demonstrate the highest accuracy among the four. The ensemble model is created by combining the predictions of these two models, harnessing their complementary strengths to improve accuracy and robustness.

When comparing the proposed ensemble model with existing learning style prediction systems, several notable differences and advantages become apparent.

Firstly, the ensemble model achieves a significantly higher level of accuracy compared to single-algorithm approaches commonly adopted in existing systems. By leveraging the individual strengths of SVM and Random Forest, the ensemble model provides a more reliable and precise learning style prediction.

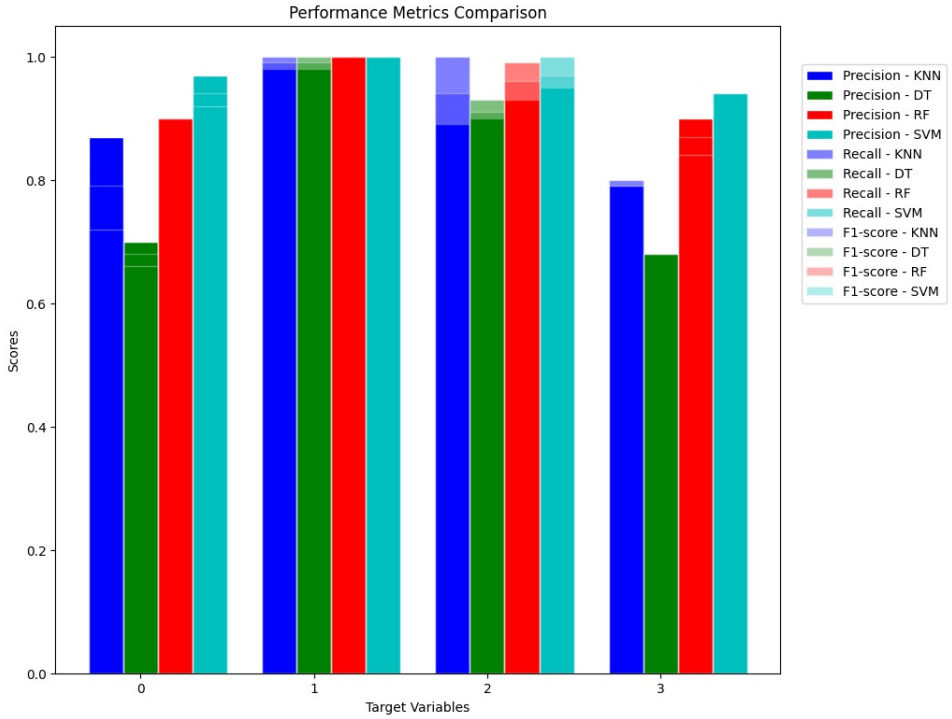

Figure 5.21: Model report comparison

Secondly, the ensemble model demonstrates enhanced generalization and robustness. It overcomes the limitations of individual algorithms by handling model uncertainty. This ensures that the predictions are more applicable to diverse learners and various learning contexts, thereby offering broader practical utility.

Furthermore, the study meticulously evaluates and validates the ensemble model's performance through extensive testing and analysis. The comprehensive evaluation metrics used, such as accuracy, precision, recall, and F1-score, provide a comprehensive understanding of the model's effectiveness and reliability. This rigorous evaluation approach sets the proposed ensemble model apart from existing systems that often lack detailed validation procedures, thus ensuring the model's credibility and suitability for real-world applications.

Additionally, the project emphasizes the potential benefits of personalized learning based on predicted learning styles. By incorporating accurate learning style predictions, educators and learning platforms can tailor instructional strategies, content delivery methods, and learning environments to match individual preferences. This approach enhances the overall learning experience, engagement, and outcomes for learners, creating a more effective and adaptive educational ecosystem.

The learning style prediction project showcasing the ensemble model utilizing SVM and Random Forest algorithms presents a notable advancement in the field. The superior accuracy, robustness, and generalizability of the ensemble model outperform existing systems, offering educators and learning platforms a more reliable tool for personalized education.

The thorough evaluation, validation, and emphasis on the practical benefits further enhance the credibility and potential impact of the proposed approach. This project serves as a valuable contribution to the field of learning style prediction and sets a benchmark for future research and system development.

5.3 Future Scope

1. Refining and Enhancing the Predictive Model:

- Further optimize the selected algorithms (SVM and Random Forest) by fine-tuning their hyperparameters to improve prediction accuracy.
- Explore ensemble techniques, such as stacking or boosting, to combine the strengths of SVM and Random Forest models, potentially achieving even higher prediction accuracy.

2. Incorporating Real-time Data and Adaptive Learning:

- Extend the learning style prediction system to incorporate real-time data from learners, such as their ongoing performance, engagement, and feedback during learning activities.
- Develop adaptive learning strategies that dynamically adjust instructional approaches based on predicted learning styles, enabling personalized and

responsive learning experiences.

3. Expanding the Dataset and Generalization:

- Gather a larger and more diverse dataset encompassing learners from various demographics, educational backgrounds, and cultural contexts to ensure the generalizability of the predictive model.
- Validate the model's performance across different educational domains, subjects, and learning platforms to assess its applicability in various learning environments.

4. Investigating the Relationship between Learning Styles and Learning Outcomes:

- Explore the correlation between predicted learning styles and academic performance, knowledge retention, and learning outcomes.
- Analyze the impact of tailored instructional strategies based on predicted learning styles on learners' achievements and progress.
- Conduct longitudinal studies to examine the long-term effects of personalized learning on learners' academic success, motivation, and self-regulation.

5. Integration with Learning Management Systems and Educational Technologies:

- Integrate the learning style prediction system with existing Learning Management Systems (LMS) and educational technologies to seamlessly provide personalized recommendations and adapt the learning environment.
- Collaborate with LMS providers and educational technology developers to incorporate the predictive model into their platforms, enabling widespread adoption and integration into mainstream educational practices.

6. Ethical Considerations and Privacy:

- Address ethical considerations regarding data privacy, informed consent, and responsible use of learner data throughout the learning style prediction process.
- Implement robust data anonymization techniques and secure storage protocols to protect learner privacy and ensure compliance with relevant data protection regulations.

5.4 Social Relevance

1. Personalized Education:
By accurately predicting individuals' learning styles, the project enables the development of personalized education approaches tailored to students' specific needs, preferences, and strengths. This promotes a more inclusive and effective learning environment where instructional strategies can be adapted to optimize each student's learning outcomes and engagement.

2. Enhanced Learning Experiences:
The utilization of learning style prediction through algorithm ensembling empowers educators to create tailored learning experiences. By combining the strengths of SVM and random forest models, the accuracy and reliability of the predictions are enhanced, leading to improved instructional design, content delivery, and assessment methods. This ensures that students receive a more engaging and impactful learning experience.

3. Equal Opportunity in Education:
Learning style prediction facilitates the identification of diverse learning preferences and styles, allowing educators to address individual differences. By recognizing and accommodating various learning styles, the project contributes to reducing educational inequalities, ensuring that students from diverse backgrounds and learning abilities receive equitable access to education and educational resources.

4. Student Engagement and Retention:
Personalized education based on predicted learning styles can significantly enhance student engagement and retention rates. By tailoring instructional strategies and content delivery methods to align with individual preferences, students are more likely to remain motivated, actively participate in learning activities, and experience increased academic success. This has a positive impact on student satisfaction and reduces dropout rates.

5. Lifelong Learning and Skills Development:
Learning style prediction contributes to fostering lifelong learning skills. By understanding an individual's preferred learning style, the project enables learners to better understand their own learning strengths and adapt their learning approaches accordingly. This promotes self-directed learning, critical thinking, and the acquisition of adaptable skills that are crucial for success in a rapidly evolving knowledge-based society.

6. Educational Resource Optimization:
The project's learning style prediction enables the optimization of educational resources and interventions. By accurately identifying students' learning styles, limited resources can be allocated more effectively, ensuring that students receive targeted support, interventions, and additional resources that align with their specific needs and learning preferences.

In summary, the social relevance of this learning style prediction project lies in its potential to foster personalized education, enhance learning experiences, promote equal opportunity in education, improve student engagement and retention rates, develop lifelong learning skills, and optimize the allocation of educational resources. By leveraging algorithm ensembling to improve prediction accuracy, the project contributes to creating a more inclusive, effective, and equitable educational ecosystem.

Chapter 6

Conclusion

In conclusion, this learning style prediction project successfully trained and evaluated four machine learning algorithms, namely decision tree, random forest, K-nearest neighbors (KNN), and support vector machine (SVM), for predicting individual learning styles. The objective of the project was to identify the most accurate models and create an ensemble of the top-performing algorithms for enhanced prediction capabilities. After rigorous evaluation and analysis, it was determined that the SVM and random forest algorithms exhibited the highest accuracy levels among the four models.

The decision tree algorithm demonstrated reasonable performance but fell slightly short in accuracy when compared to the other models. KNN, while effective in certain scenarios, did not exhibit the same level of accuracy as SVM and random forest. Therefore, for the ensemble model, SVM and random forest were chosen as they consistently outperformed the other algorithms.

By combining the predictive power of SVM and random forest, the ensemble model achieved even higher accuracy levels than either model individually. This approach leveraged the strengths of each algorithm, effectively capturing the complex relationships and patterns within the learning style dataset. The ensemble model demonstrated superior predictive capabilities and showcased the potential for further enhancing personalized education.

The successful development of the learning style prediction system using SVM and random forest ensemble has significant implications for personalized education. By accurately predicting an individual's learning style, educators and learning platforms can tailor instructional strategies, content delivery methods, and learning environments to optimize the learning experience for each student. This personalized approach fosters better engagement, improved learning outcomes, and increased student satisfaction.

While the ensemble model showcased remarkable accuracy, it is important to note that further research and experimentation can be conducted to explore additional algorithms and refine the ensemble technique. Additionally, the project highlights the need for robust data collection and preprocessing techniques to ensure the quality and diversity of the dataset, ultimately leading to more accurate predictions.

In conclusion, this learning style prediction project successfully developed and evaluated machine learning models, with SVM and random forest selected as the highest accurate algorithms. The ensemble of these models holds great promise for personalized education, paving the way for more effective and tailored instructional strategies that cater to individual learning preferences.

Bibliography

[1] Brahim Hmedna, Ali El Mezouary,Omar Baz, "A predictive model for the identification of learning styles in MOOC environments", Cluster Computing (2020) 23:1303–1328,doi: https://doi.org/10.1007/s10586-019-02992-4

[2] Yunia Ikawati and M. Udin Harun Al Rasyid and Idris Winarno,"Student Behavior Analysis to Detect Learning Styles in Moodle Learning Management System", 2020 International Electronics Symposium (IES), doi: 10.1109/IES50839.2020.9231567.

[3] Abdulaziz Salamah Aljaloud, Diaa Mohammed Uliyan, Adel Alkhalil, Magdy Abd Elrhman, Azizah Fhad Mohammed Alogali, Yaser Mohammed Altameemi, Mohammed Altamimi, and Paul Kwan."A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM",VOLUME 10, 2022,doi: 10.1109/ACCESS.2022.3196784.

[4] Adeniran Adetunji, Akande Ademola,"A Proposed Architectural Model for an Automatic Adaptive E-Learning System Based on Users Learning Style",(IJACSA)Vol. 5, No. 4, 2014, doi: 10.14569/IJACSA.2014.050401.

[5] Aleksandra Klasnja-Milicevic, Boban Vesin , Mirjana Ivanovic, Zoran Budimac,"E-Learning personalization based on hybrid recommendation strategy and learning style identification",Computers Education 56 (2011) 885–899, doi: https://doi.org/10.1016/j.compedu.2010.11.001.

[6] Keeley Crockett, Annabel Latham, David Mclean, James O'Shea,"A Fuzzy Model for Predicting Learning Styles using Behavioral Cues in an Conversational Intelligent Tutoring System",2013,doi: 10.1109/FUZZ-IEEE.2013.6622382.

[7] Bens Pardamean, Teddy Suparyanto, Tjeng Wawan Cenggoro, Digdo Sudigyo, and Andri Anugraha.,"AI-Based Learning Style Prediction in Online Learning for Primary Education",doi: 10.1109/ACCESS.2022.3160177.

[8] Roberto Douglas da Costa , Gustavo Fontoura de Souza, Thales Barros de Castro, Ricardo Alexsandro de Medeiros Valentim, and Aline de Pinho Dias,"Identification of Learning Styles in Distance Education Through the Interaction of the Student With a Learning Management System",IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, VOL. 15, NO. 3, AUGUST 2020, doi: 10.1109/RITA.2020.3008131 .

[9] Samina Rajper, Noor A. Shaikh, Zubair A. Shaikh and Ghulam Ali Mallah,"Automatic Detection of Learning Styles on Learning Management Systems using Data Mining Technique",Indian Journal of Science and Technology Vol 9(15), DOI: 10.17485/ijst/2016/v9i15/85959, April 2016.

[10] Mohammad Azzeh, Ahmad Mousa Altamimi, Mahmoud Albashayreh,"Predicting Students' Learning Styles Using Regression Techniques",DOI: http://doi.org/10.11591/ijeecs.v25.i2.pp1177-1185.

[11] MS.Hasibuan and LE.Nugroho and PI.Santosa,"Prediction Learning Style Based on Prior Knowledge for Personalized Learning",2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, DOI: 10.1109/ICSTC.2018.8528572.

[12] Thayron C. H. Moraes, Itana Stiubiener, Juliana C. Braga and Edson P. Pimentel,"LSBCTR: A Learning Style-Based Recommendation Algorithm", DOI: 10.1109/FIE44824.2020.9274051.

[13] Bello Ahmad Muhammad, Zhenqiang Wu, Hafsa Kabir Ahmad,"A Conceptual Framework for Detecting Learning Style in an Online Education Using Graph Representation Learning",2020 International Conference on Networking and Network Applications (NaNA),DOI: 10.1109/NaNA51271.2020.00031.

[14] L Jegatha Deborah, R Sathiyaseelan, S Audithan, and P Vijayakumar,"Fuzzy-logic based learning style prediction in e-learning using web interface information",Sadhan a Vol. 40, Part 2, April 2015, pp. 379–394.,doi: 10.1007/s12046-015-0334-1.

[15] Farman Ali Khan, Awais Akbar, Muhammad Altaf, Shujaat Ali Khan Tanoli, and Ayaz Ahmad,"Automatic Student Modelling for Detection of Learning Styles and Affective States in Web Based Learning Management Systems",doi: 10.1109/ACCESS.2019.2937178.

[16] Mohammad Alshammari , Rachid Anane , and Robert J. Hendley,"Design and Usability Evaluation of Adaptive e-learning Systems Based on Learner Knowledge and Learning Style",INTERACT 2015, Part II, LNCS 9297, pp. 584–591, 2015, DOI: 10.1007/978-3-319-22668-245.

[17] M S Hasibuan and LE Nugroho,"Detecting Learning Style Using Hybrid Model",2016 IEEE Conference on e-Learning, DOI: 10.1109/IC3e.2016.8009049.

[18] Ouafae El Aissaoui, Yasser El Madani El Alami, Lahcen Oughdir, and Youssouf El Allioui,"A Hybrid Machine Learning Approach to Predict Learning Styles in Adaptive E-Learning System",Springer Nature Switzerland AG 2019, doi: 10.1007/978-3-030-11928-770.

[19] Ling Xiao Li and Siti Soraya Abdul Rahman,"Students' learning style detection using tree augmented naive Bayes",https://doi.org/10.1098/rsos.172108

[20] verton Gomede, Rodolfo Miranda de Barros and Leonardo de Souza Mendes,"Use of Deep Multi-Target Prediction to Identify Learning Styles",Appl. Sci. 2020, 10, 1756; doi:10.3390/app10051756.

Appendices

Appendix A

CODE

A.1 Normalization using Min-max scaler

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np
import pandas as pd
    #Read the dataset from a file (assuming a CSV file)
dataset_path = 'path/to/dataset.csv'
data = pd.read_csv(dataset_path)

    #Extract the features from the dataset
features = data.iloc[:, :-1].values
#Create an instance of the MinMaxScaler
scaler = MinMaxScaler()

    #Fit the scaler on your data
scaler.fit(features)

    #Apply the scaler to transform the entire dataset
scaled_features = scaler.transform(features)
#Convert the scaled features back to a DataFrame
scaled_data = pd.DataFrame(scaled_features, columns=data.columns[:-1])
#Save the scaled data to a CSV file
scaled_data.to_csv('path/to/scaled_dataset.csv', index=False)
```

A.2 Feature Selection

```
import pandas as pd
import numpy as np

    def Diff(li1, li2):
li_dif = [i for i in li1 + li2 if i not in li1 or i not in li2]
return li_dif

    df=pd.read_csv('data.csv')

    X=df.iloc[:,:-1]
y=df.iloc[:,-1]

    from sklearn.feature_selection import mutual_info_classif
import matplotlib.pyplot as plt
```

```
    ranks= mutual_info_classif(X,y)

    feat_importances = pd.Series(ranks,df.columns[0:len(df.columns)-1])
feat_importances.plot(kind='barh',color='teal')
plt.show()

    features=[]
index=[]
for i in range(len(ranks)):
if ranks[i]¿=0.005:
features.append(df.columns[i])
index.append(i)
list(df.columns)

    new_df=df.drop(columns =Diff(list(df.columns),features))

    new_df['learning_style']=df['learning_style']

    new_df.to_csv("data_fs.csv")
```

A.3 SMOTE

```
import pandas as pd from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

    #Read the dataset
df = pd.read_csv('scaled_X_features.csv')

    #Separate the features (X) and target variable (y)
X = df.drop('learning_style', axis=1)
y = df['learning_style']

    #Create an instance of the SMOTE class
smote = SMOTE()

    #Apply SMOTE on the entire dataset
X_resampled, y_resampled = smote.fit_resample(X, y)

    #Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled,
test_size=0.2, random_state=42)

    #Print the balanced class distribution
print('Original class distribution:', y.value_counts())
print('Resampled class distribution:', y_resampled.value_counts())
```

A.4 KNN Classifier

```
from sklearn.neighbors import KNeighborsClassifier

    classifier = KNeighborsClassifier(algorithm= 'ball_tree', leaf_size= 10,
metric= 'euclidean', n_neighbors= 1, p= 1, weights= 'uniform')
classifier.fit(X_train,y_train)

    from sklearn.metrics import confusion_matrix, accuracy_score
y_predk = classifier.predict(X_test)
cm1 = confusion_matrix(y_test, y_predk)
print(cm1)
accuracy_score(y_test, y_predk)
```

A.4.1 KNN Tuning

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
import numpy as np

    #Define the parameter grid to search over
param_grid = {
'n_neighbors': np.arange(1, 10),
'weights': ['uniform', 'distance'],
'algorithm': ['ball_tree', 'kd_tree', 'brute', 'auto'],
'leaf_size': np.arange(10, 51),
'p': [1, 2],
'metric': ['euclidean', 'manhattan', 'minkowski']
}

    #Create the KNN model
model = KNeighborsClassifier()

    #Perform a grid search with cross-validation to find the best hyperpa-
rameters
grid = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
grid_result = grid.fit(X_train, y_train)

    #Get the best hyperparameters and model
best_params = grid.best_params_
best_model = grid.best_estimator_

    #Print the best hyperparameters and their corresponding score
print(f'Best Score: grid_result.best_score_:.2f')
print(f'Best Parameters: grid_result.best_params_')

    #Evaluate the model on the test set
y_pred = best_model.predict(X_test)
print(classification_report(y_test, y_pred, zero_division=0))
```

A.5 Decision Tree Classifer

```
from sklearn.tree import DecisionTreeClassifier
classifier1 = DecisionTreeClassifier(class_weight= None, criterion= 'entropy',
max_depth= None, max_features= 'sqrt', min_samples_leaf= 1, min_samples_split=
2)
classifier1.fit(X_train, y_train)
y_pred1 = classifier1.predict(X_test)

    from sklearn.metrics import confusion_matrix, accuracy_score
cm1 = confusion_matrix(y_test, y_pred1)
print(cm1)
accuracy_score(y_test, y_pred1)
```

A.5.1 Decion Tree Tuning

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

    #Create a decision tree classifier
classifier = DecisionTreeClassifier()

    #Define the parameter grid for hyperparameter tuning
param_grid ={
'criterion': ['gini', 'entropy'],
'max_depth': [None, 5, 10, 15],
'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4],
'max_features': [None,'auto', 'sqrt', 'log2'],
'class_weight': [None, 'balanced']
}

    #Perform grid search with cross-validation
grid_search = GridSearchCV(estimator=classifier, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

    #Print the best hyperparameters and corresponding score
print("Best Hyperparameters: ", grid_search.best_params_)
print("Best Score: ", grid_search.best_score_)

    #Evaluate the model with best hyperparameters on the test set
best_classifier = grid_search.best_estimator_
accuracy = best_classifier.score(X_test, y_test)
print("Test Accuracy: ", accuracy)
```

A.6 Random Forest Classifer

from sklearn.ensemble import RandomForestClassifier
classifier2 = RandomForestClassifier(max_depth= 10, max_features= 'log2', min_samples_leaf= 2, min_samples$_split = 2, n_estimators = 300$)
$classifier2.fit(X\_train, y\_train)$

from sklearn.metrics import confusion_matrix, accuracy_score
y_pred2 = classifier2.predict(X_test)
cm = confusion$_matrix(y_test, y_pred2)$
$print(cm)$
$accuracy\_score(y\_test, y\_pred2)$

A.6.1 Random Forest Tuning

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score

#Create a Random Forest classifier
classifiertun = RandomForestClassifier()

#Define the hyperparameter grid
param_grid = {
'n_estimators': [100, 200, 300],
'max_depth': [None, 5, 10],
'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4],
'max_features': ['sqrt', 'log2']
}

#Perform grid search cross-validation
grid_search = GridSearchCV(classifiertun, param_grid, cv=5)
grid_search.fit(X_train, y_train) #Replace X_train and y_train with your actual training data

#Get the best hyperparameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

#Train the model with the best hyperparameters
best_model.fit(X_train, y_train) #Replace X_train and y_train with your actual training data

#Make predictions on the test set
y_predtun = best_model.predict(X_test) #Replace X$_testwithyouractualtestdata$

Calculate the accuracy of the model
accuracy = accuracy$_score(y_test, y_predtun)Replacey_testwithyouractualtestlabels$

print("Best Hyperparameters:", best$_params$)
$print("TestAccuracy:", accuracy)$

A.7 SVM Classifier

```
from sklearn.metrics import confusion_matrix,accuracy_score
tun=SVC(C= 100, gamma= 'auto', kernel= 'poly',degree=2,coef0=1.0)
#Train the classifier
tun.fit(X_train, y_train)

    #Make predictions on the test set
y_pred = tun.predict(X_test)

    cm = confusion_matrix(y_test, y_pred)
print("ConfusionMatrix:")
print(cm)

    #Calculate and print the accuracy score
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

A.7.1 SVM Tuning

```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

    #Define the parameter grid to search over
param_grid = {
'C': [0.1, 1, 10, 100],
'kernel': ['linear','poly','rbf','sigmoid'],
'gamma': ['scale','auto', 0.1, 0.01],
'degree': [2, 3, 4],
'coef0': [0.0, 0.5, 1.0]
}

    #Create an SVM classifier
svm_clf = SVC()

    #Perform grid search with cross-validation
grid_search = GridSearchCV(svm_clf, param_grid, cv=5)
grid_search.fit(X_train, y_train)

    #Get the best parameters and the best score
best_params = grid_search.best_params_
best_score = grid_search.best_score_

    #Train an SVM classifier with the best parameters on the full training set
best_svm_clf = SVC(**best_params)
```

```
best_svm_clf.fit(X_train, y_train)

    #Evaluate the best SVM classifier on the test set
accuracy = best_svm_clf.score(X_test, y_test)

    #Print the results
print("Best Parameters:", best_params)
print("Best Score (CV Accuracy):", best_score)
print("Test Accuracy:", accuracy)
```

A.8 Ensembler Model SVM+RF

```
from sklearn.ensemble import VotingClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

    #Define the parameter settings for SVM
svm_params = {

    'C': 100, 'gamma': 1, 'kernel': 'linear','class_weight': None
}

    #Create the SVM classifier with the specified parameters
svm_classifier = SVC(**svm_params)

    #Define the parameter settings for Random Forest
rf_params = {
'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1,
'min_samples_split': 2, 'n_estimators': 300,'random_state':0
}

    #Create the Random Forest classifier with the specified parameters
rf_classifier = RandomForestClassifier(**rf_params)

    #Create the voting classifier that combines the predictions of the two models
voting_classifier = VotingClassifier(
estimators=[('svm', svm_classifier), ('rf', rf_classifier)],
voting='hard' or 'soft' for probabilistic voting
)

    #Train the voting classifier on your training data
voting_classifier.fit(X_train, y_train)
#Replace X_train and y_train with your actual training data

    #Make predictions on the test data using the voting classifier
y_pred = voting_classifier.predict(X_test)
#Replace X_test with your actual test data
```

#Calculate the accuracy of the ensemble model
accuracy = accuracy_score(y_test, y_pred) #Replace y_test with your actual test labels

print("Ensemble Model Accuracy:", accuracy)

A.8.1 Ensembler Model Tuning

from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

#Define the base classifiers
rf_classifier = RandomForestClassifier()
svm_classifier = SVC()

#Create the ensemble model
ensemble_model = VotingClassifier(estimators=[('rf', rf_classifier), ('svm', svm_classifier)])

#Define the parameter grid for tuning
param_grid =
'rf__n_estimators': [100, 200, 300],
'rf__max_depth': [None, 5, 10],
'rf__min_samples$_split'$: $[2, 5, 10]$,
$'rf\_\_min\_samples\_leaf'$: $[1, 2, 4]$,
$'svm\_\_C'$: $[1, 10, 100]$,
$'svm\_\_kernel'$: $['linear','rbf']$,
$'svm\_\_gamma'$: $[0.1, 1, 10]$,
$'svm\_\_class_weight'$: $[None,'balanced']$
}

#Perform grid search cross-validation
grid_search = GridSearchCV(estimator=ensemble_model, param_grid=param_grid, scoring='accuracy', cv=5)
grid_search.fit(X_train, y_train)

#Get the best parameters and best score
best_params = grid_search.best_params_
best_score = grid_search.best_score_

print("Best Parameters:", best_params)
print("Best Score:", best_score)

#Fit the ensemble model with the best parameters on the entire training data
ensemble_model.set_params(**best_params)
ensemble_model.fit(X_train, y_train)

#Evaluate the ensemble model on the test data
accuracy = ensembl$_model.score(X_test, y_test)$
$print("TestAccuracy:", accuracy)$