# DECODING INSTRUCTOR PERFORMANCE WITH DATA MINING AND MACHINE LEARNING

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**
**in**
**Computer Science & Engineering**
**School of Engineering & Sciences**

submitted by

**Namratha Addagada(AP21110010048)**

Under the Guidance of

**Dr. S. Sumalatha**

**Department of Computer Science & Engineering**
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2025

# DECLARATION

I undersigned hereby declare that the project report **Decoding Instructor Performance with Data Mining and Machine Learning** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology in the Computer Science & Engineering, SRM University-AP, is a bonafide work done by me under supervision of Dr. S. Sumalatha. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree of any other University.

| | | | | | |
|---|---|---|---|---|---|
| Place | : | SRM University AP | Date | : | June 4, 2025 |
| Name of student | : | Namratha Addagada | Signature | : | L. Sai Gopal .................................. |
| | | | | | A. Namratha |
| | | | | | A. Haasitha |

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SRM University-AP**

**Neerukonda, Mangalgiri, Guntur**

**Andhra Pradesh - 522 240**

## CERTIFICATE

This is to certify that the report entitled **Decoding Instructor Performance with Data Mining and Machine Learning** submitted by **Namratha Addagada,** to the SRM University-AP in partial fulfillment of the requirements for the award of the Degree of Master of Technology in in is a bonafide record of the project work carried out under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

| Project Guide | Head of Department |
|---|---|
| Name : Dr. S. Sumalatha | Name : Dr. Murali Krishna Enduri |
| Signature : S. Sumalatha | Signature : .......................... |

# ACKNOWLEDGMENT

I wish to record my indebtedness and thankfulness to all who helped me prepare this Project Report titled **Decoding Instructor Performance with Data Mining and Machine Learning** and present it satisfactorily.

I am especially thankful for my guide and supervisor Dr. S. Sumalatha in the Department of Computer Science & Engineering for giving me valuable suggestions and critical inputs in the preparation of this report. I am also thankful to Dr. Murali Krishna Enduri, Head of Department of Computer Science & Engineering for encouragement.

My teammates have been a constant source of encouragement and support, and I'm deeply grateful for their patience in listening to my presentations and offering valuable feedback.



Sai Gopal Lanka, Namratha Addagada, Haasitha Ambati

(Reg. No. AP21110010042, AP21110010048, AP21110010049)

B. Tech.

Department of Computer Science & Engineering

SRM University-AP

# ABSTRACT

Educational Data Mining is an emerging discipline with the objective of improving educational quality and problems like instructor performance prediction. Instructor performance is being measured in this research using three datasets: the Turkiye Student Evaluation Dataset, the Poland University Dataset, and a Synthetic Dataset. Feature selection techniques like Recursive Feature Elimination and Random Forest-based selection are used. Seven classification models—Decision Tree, Naive Bayes, Logistic Regression, Support Vector Machines (SVM), AdaBoost, Multi-Layer Perceptron (MLP) Classifier and XGBoost—are implemented on all the datasets, hence it is a 3×3 comparison. The best performing model, which is Decision Tree with SVM on the Synthetic Dataset, has an accuracy rate of 90.15%. A web application is also built with this model so that users can enter instructor information and get performance predictions with actionable recommendations.

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# INTRODUCTION TO THE PROJECT

Exploiting data mining and machine learning techniques to predict the instructor's performance is a reliable project where we can use various methodologies and techniques to experiment and fetch the best results. Firstly, let's understand the term Instructor's Performance and how it's been calculated, Instructor's Performance (IP) is the performance of any Instructor illuminating any concept for anyone. The Performance can be calculated using various methodologies where one can take feedback from the listeners of the Instructor teaching, or with the help of some standard teaching methods. The hardships while calculating the Instructor's performance are condensed by checking whether the Instructor is following few of the best methodologies such as modeling-based teaching (MBT), etc which are usually used in increasing the Instructor's Performance. The actual hardship is to find a dataset which consists of columns with various contrast methodologies that will be followed by the instructors and columns which are related to the student-teacher interaction, etc. Various real world datasets and synthetic datasets are taken into consideration where many hybrid machine learning algorithms and techniques are used to find out the best accurate solutions which are explained in detail moving forward.

The objective of this work is to predict the instructor performance with the help of student evaluation responses and their feedback of the corresponding instructors from various sources, which, in turn, will be useful in educational settings to assess and evaluate the quality of education, instructor's ability to deliver the course and overall perception of the students. In most instances, both course specific and instructor specific questions were included to involve all the factors that impact the course deliverability. Difficulty of the course, learner's engagement in course-related activities such as labs, projects, and so on, and alignment of the lessons with the syllabus given, were some of the questions that were associated with the course and its content. Alongside those, questions pertaining to the instructor such as educational qualifications, their preparedness for the class and fairness towards the students were also taken in consideration.

The literature review section covers the different approaches taken by researchers in tackling the instructor performance evaluation problem and also about the techniques that helped us understand the machine learning models and techniques to improve their performance. Additionally, some papers had information on synthetic datasets that helped in gaining insights on how different parameters affect the instructor's performance and generating a synthetic dataset. Few papers that were mentioned in the section helped us frame questions in such a way that different teaching techniques are evaluated at the same time; questions related to student-teacher interaction, question paper difficulty, punctuality, use of modern gadgets, etc are considered for evaluating the instructor's performance. Most Researchers have considered taking Student Evaluation of Teaching data from various educational institutions while some have scraped data from the internet especially from websites like ratemyprofessor.com where students get to write reviews on their instructors. The researchers have considered instructor specific aspects like age, experience, gender, punctuality, availability after hours and course specific aspects. like difficulty of the course, whether the course outcomes have been met or not etc for their datasets. Some have tried boosting approaches while others have experimented on the effect of feature selection methods on accuracy when applied with different classifiers. The dataset section gives an insight on the three different datasets considered: 1. Poland Dataset 2. Turkiye Student Evaluation Dataset 3. Synthetic Dataset

The Poland dataset with its 5800 records consists of average student evaluation of teaching results for each instructor and has instructor specific data like their gender, total number of years of experience, the number of degrees they hold etc. Turkey dataset from Gazi University, Turkey has 5820 records with each record consisting of student response to a student evaluation of a teaching questionnaire with 28 questions regarding the course and the instructor rated between 1 and 5. The third dataset is a synthetic dataset generated with 11 questions rated by students from 1-5 regarding the course and the instructor. The different approaches taken to clean the three different datasets, be it feature reduction using correlation, tackling null values, attributes with null values and dataset imbalance has been discussed in the data preprocessing section. Finally the methodology and results sections discuss the various approaches taken for training the models, feature selection methods and the respective accuracies. Some papers were explored in order to understand the complex machine learning techniques such as boosting approach in ensembling techniques, recursive-feature elimination, SMOTE etc.

To make our project more meaningful and closer to a real-world application, we expanded our experiments by applying each machine learning method individually to all three datasets—two of which are publicly available (Turkiye and Poland), and one that we generated ourselves. This gave us a complete 3×3 model-dataset comparison, helping us evaluate how different models perform across different data sources. Among all the combinations, the fusion of Decision Tree and Support Vector Machine applied on our synthetic dataset gave the highest accuracy. Since the synthetic dataset was designed based on real student feedback patterns and practical teaching parameters, we chose it as the foundation to build a web application. The website allows users to enter an instructor's name or ID and view predicted performance scores along with detailed evaluation metrics—making our research not just theoretical, but also useful in real-world educational settings.

# Chapter 2

# MOTIVATION AND REAL-WORLD RELEVANCE

In this chapter, we discuss the driving factors behind selecting a project that explores data mining and machine learning techniques to predict instructor performance. This section articulates the broader reasons why such a project is not only pivotal for final-year engineering students but also valuable in contributing to the quality of teaching and learning processes.

## 2.1 SIGNIFICANCE OF DATA-DRIVEN INSIGHTS IN EDUCATION

The field of education is undergoing a rapid transformation: traditional teaching methods are being supplemented or replaced by innovative, technology-driven approaches. As data accumulates in educational institutions – from student feedback forms to academic performance metrics – leveraging this information becomes essential. Data mining and machine learning techniques offer sophisticated means to extract insights from large, often complex datasets. By focusing on instructor performance, students can tackle a real-world challenge with immediate relevance and impact:

- **Enhancement of Teaching Quality:** Analyzing patterns in feedback and evaluations enables administrators and instructors to pinpoint areas of improvement, making teaching more effective and engaging.
- **Evidence-Based Decision Making:** Educational institutions can adopt

data-driven policies, such as allocating resources to training programs or restructuring course delivery, guided by patterns in instructor evaluation data.

- **Feedback Loop for Continuous Improvement:** Capturing and interpreting student evaluations throughout the semester can create a regular feedback mechanism, helping instructors to refine their course content and methodology in near real-time.

## 2.2 WHY FINAL-YEAR ENGINEERING PROJECTS BENEFIT FROM REAL-WORLD DATA ANALYSIS

Final-year engineering projects often serve as a bridge between classroom theory and professional practice. Data analytics and machine learning have emerged as some of the most sought-after skill sets in the job market. By applying these competencies in a domain like education, students gain both technical experience and domain-specific insight:

- **Industry Alignment:** Modern industries, from healthcare to finance, rely on data science. A project that harnesses machine learning models on educational data not only builds advanced skills but also demonstrates versatility and adaptability.
- **Cross-Disciplinary Learning:** Working on data related to instructor performance demands knowledge of statistics, coding, domain understanding, and communication skills – all of which are integral to holistic engineering education.
- **Practical Exposure:** By treating real or realistic educational datasets, students handle the kinds of unstructured or messy data encountered

outside of academia. This practical immersion better prepares them for future career challenges.

## 2.3 TURNING IDEAS INTO TANGIBLE SOLUTIONS

One of the key motivations behind this project is showing how abstract knowledge in data science can be translated into concrete tools or frameworks that stakeholders in academia can use:

### 2.3.1 Bridging the Gap Between Theory and Practice

Data mining and machine learning concepts are typically covered in theory-based courses. Through a project on instructor performance prediction, students apply and see these methods in action – from handling incomplete data to tuning classifiers for higher accuracy. They also learn to cope with real-world complexities like unbalanced class distributions or subjective feedback scales.

### 2.3.2 Diverse Research Opportunities

As part of designing robust instructor-evaluation systems, students explore questions such as:

- Which features (e.g., clarity, preparedness, empathy) most affect instructor performance?
- How do different machine learning algorithms (e.g., ensemble methods, neural networks) compare in terms of predictive accuracy?
- What are the potential biases and ethical considerations in evaluating human performance through algorithmic means?

These inquiries foster deeper understanding and encourage innovative problem-solving. In some cases, they can lead to publishable findings or new research avenues.

### 2.3.3 From Evaluation to Real-World Implementation

While most academic projects conclude with model evaluation, we aimed to go a step further by turning our best-performing approach into a usable product. After comparing all model-dataset combinations, we found that the Decision Tree + SVM combination on our own synthetic dataset yielded the highest accuracy. Since the synthetic dataset was carefully designed to simulate real-world feedback patterns, we considered it suitable for practical application.

As a result, we developed a web-based platform where users can input an instructor's name or ID and receive detailed predictions and insights on their teaching performance. This transition from theoretical modeling to a functional tool demonstrates how academic research can be translated into solutions that benefit educational stakeholders in real time.

## 2.4 ENSURING THOROUGH DOCUMENTATION AND PRESENTATION

As with any engineering project, a thorough documentation process enables others to replicate, extend, or audit the work. In the context of predicting instructor performance:

- **Clarity of Methodology:** Outlining data cleaning steps, feature engineering processes, and modeling approaches ensures transparency and scientific integrity.

- **Usability for Stakeholders:** Presenting the results in a clear, user-friendly format (e.g., dashboards, reports) can directly inform institutional strategies for improving teaching standards.
- **Long-Term Value:** A well-documented solution is easier to maintain, refine, and update as new data becomes available, or as teaching evaluation criteria evolve.

## 2.5 EFFECTIVE PLANNING AND PROJECT MANAGEMENT

To maximize the educational benefits, the project enforces a structured workflow that hones project management capabilities:

- **Early Prototyping:** Determining feasibility with small test datasets and analyzing whether chosen algorithms align with the final goal.
- **Milestone Tracking:** Adhering to a timeline that includes data collection, preprocessing, model development, accuracy improvement, and final deployment steps.
- **Team Collaboration:** Many final-year projects involve teamwork. Balancing roles in data handling, model development, and results interpretation helps improve communication and leadership skills.

## 2.6 A PLATFORM FOR PROFESSIONAL AND PERSONAL GROWTH

Working on a forward-looking project that leverages machine learning to predict instructor effectiveness provides multiple growth opportunities:

### 2.6.1 Technical Competency

Students learn to navigate popular data science libraries, build and test multiple models, and interpret quantitative metrics like precision, recall, and accuracy. These are indispensable proficiencies for a career in analytics or software development.

### 2.6.2 Ethical and Social Perspectives

Evaluating human performance is inherently sensitive. Reflecting on fairness, privacy, and transparency encourages the adoption of ethical data-mining practices. It teaches students to handle real-world consequences—how might stakeholders react to automated evaluations or the visibility of negative feedback?

### 2.6.3 Career Readiness

The final-year project yields a tangible outcome in a rapidly growing field. This project experience signals to prospective employers that graduates can tackle end-to-end solutions—from problem formulation to model deployment—while adhering to professional and ethical standards.

## 2.7 CONCLUDING REMARKS ON THE IMPORTANCE OF MOTIVATION

Undertaking a final-year engineering project that incorporates data mining and machine learning to predict instructor performance holds significance beyond academic requirements. It merges interdisciplinary knowledge, addresses authentic problems within educational contexts, and yields insights that can meaningfully enhance the teaching and learning experience. By immersing in robust analytical processes, students reinforce their foundation in engineering and graduate with heightened readiness to face the data-driven challenges of modern industries.

# Chapter 3

# LITERATURE SURVEY

The authors in Paper [1] applied two well known data mining techniques; stepwise regression and decision tree techniques. For regression analysis the stepwise regression method and for decision trees CHAID and CART algorithms are applied. The stepwise regression procedure is one of the variable selection methods in regression analysis. The dataset includes all course sections from Fall 2004 to Summer 2009. The data collected is based on 12 questions: To reduce the number of independent variables and to understand the patterns of relationship among them the factor analysis was applied. Instructors, who have well prepared course outlines, use satisfactory materials, help the students outside the lectures, grade exams fairly and on time receive higher evaluations. An additional instructor characteristic; the employment status of the instructor that is not included in the questionnaire is found to be significant.

Paper [2] considered a dataset consisting of student evaluation of teaching (SET) survey collected from Gazi University, Turkey. The dataset consisted of 33 attributes. The authors have used Weka Ranker to rank the attributes. The 10 least important attributes have been dropped and the remaining attributes—which were the top 24—were considered for the next step. The authors used four different classification approaches, and for each type of classifier, they trained two models: one model with the top 24 attributes and the second trained and tested on all the attributes. The highest accuracy seen by the authors was 85increase in accuracy in the best case when the aforementioned procedure was carried out on the classifiers.

Paper [3] considered a dataset consisting of student evaluation of teaching (SET) survey collected from Gazi University, Turkey. The dataset consisted of 33 attributes. The authors have used Weka Ranker to rank the attributes. The 10 least important attributes have been dropped and the remaining attributes—which were the top 24—were considered for the next step. The authors used four different classification approaches, and for each type of classifier, they trained two models: one model with the top 24 attributes and the second was trained and tested on all the attributes. The highest accuracy seen by the authors was 85%. They concluded that removing the worst attributes had very little impact on improving the accuracy as there was only a 2% increase in accuracy in the best case.

The authors in [4] used a system consisting of two layers with an MLP and a decision tree. They tried two different approaches with two different decision trees namely ID3 and C4.5. The data used in this study was collected from the Department of Academics of a university in Ondo state, South West Nigeria. The data was collected over a span of 6 years from 2010 to 2015. It consisted of information about the instructor like their ID, age, rank, experience, qualification. Along with this, the data also included course-related questions like course ID and student satisfaction. Attribute selection was first carried out followed by training and testing. It was concluded that C4.5 outperformed the ID3 decision tree with an accuracy of 83The authors also concluded that attributes like experience and rank had a major impact on the performance of the instructor, while professional qualification had little impact on their performance.

In paper [5], the authors used the surge in demand for instructors at the onset of COVID-19 as motivation to develop a model to help predict the performance of instructors, which can assist the educational institution's management in hiring and for existing instructors to improve their performance. A dataset with over 10,000 records was created using SET surveys spanning over 4 semesters. It considered courtesy, punctuality, grooming, attendance, and the ability of the instructor to explain the concept clearly as attributes in the dataset. The target variable consisted of the performance ranging from outstanding, very satisfactory, satisfactory, unsatisfactory, to poor. The authors used a decision tree to predict the performance of the instructor.

The main focus of the work described in [6] is on predicting the instructor performance and investigating the factors that affect students' achievements to improve the education system quality. The most important attributes of the Turkey Student Evaluation records dataset were selected, and 4 different classifications (J48 Decision Tree, Multilayer Perceptron, Naïve Bayes, and Sequential Minimal Optimization) were applied. Selection of attributes impacted the performance of classifiers, and varying results were obtained in different cases. It was also observed that the performance of an instructor is mainly affected by the number of courses that are taught.

The study "Predicting Instructor Performance Using Data Mining Techniques in Higher Education" in [7] takes an unusual approach to predicting instructor performance in higher education using data mining techniques. The study is significant because it addresses the difficult task of evaluating instructor performance, which is frequently subjective and influenced by a variety of factors such as student biases, course content, and teaching style. The authors predicted four metrics of instructor performance using a dataset of 47 instructors and 1,676 student evaluations: overall instructor rating, course quality, course difficulty, and student learning. They then used three distinct data mining techniques to predict instructor performance on the dataset: decision tree, random forest, artificial neural network, and discriminant analysis, comparing their performances. C5.0 classifier performed the best in terms of accuracy (92.3(94.4

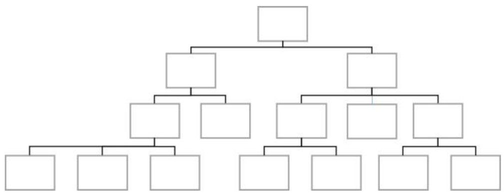


Figure 3.1: Decision Tree diagram from [7]

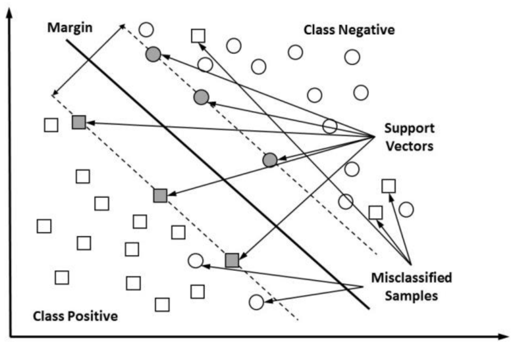



Figure 3.2: Support Vector Machine diagram from [7]

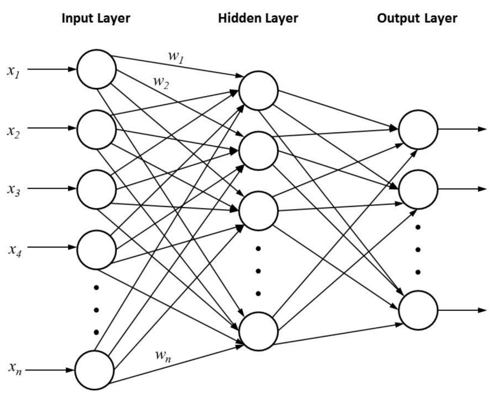



Figure 3.3: Artificial Neural Network diagram from [7]

Furthermore, the authors discovered that certain characteristics, such as teacher experience, student attendance, and course grade distribution, had a substantial impact on instructor performance. This approach, according to the authors, can be used to identify potential areas of improvement in teacher performance and to build focused interventions to improve teaching quality. For example, if the data mining analysis indicates that course grade distribution is a key influence in instructor performance, course design and grading standards could be modified to improve student learning outcomes and instructor evaluations. Overall, this study demonstrates the ability of data mining approaches to provide insights into instructor performance in higher education and to aid ongoing efforts to improve teaching quality.

The authors of the paper "Predicting Instructor Performance in Online Courses: A Multi-Criteria Approach Using Data Mining Techniques" [8] used data mining techniques to predict online instructors' performance based on several factors. They gathered information from a big online learning platform, such as course ratings, teacher ratings, student involvement, instructor experience, and course complexity. They then developed a model to predict instructor performance using a variety of data mining approaches such as decision trees, random forests, and support vector machines.

The accuracy, precision, recall, and F1 score were among the measures used by the authors to assess the model's performance. The results showed that the random forest model had the highest accuracy of 86.6%, demonstrating that it can accurately predict instructor performance. The authors also ran a sensitivity study to see how other criteria affected the model's performance. They discovered that the most relevant criteria for predicting instructor performance were course ratings and instructor ratings, followed by student involvement and course difficulty. Overall, this study indicates the potential of data mining approaches for predicting instructor success in online education and emphasizes the significance of evaluating instructor performance using several factors. The model's high accuracy values suggest its potential to improve the quality of online education and students' learning experiences.

Eighteen ML algorithms' performance were considered by Basem S. Abunasser et al. in [9], out of which Extra Trees Regressor turned out to be the best machine learning model in terms of accuracy, precision, recall, and F1 score over a dataset obtained from the UCI repository. In addition, the authors proposed a Deep Learning model consisting of seven Dense layers: one input layer (33 features), five hidden layers (256, 128, 64, 32, and 16 neurons), and one output layer with five classes (softmax). This DL model yielded better results compared to the rest of the 18 algorithms in terms of accuracy, precision, recall, and F1 score (98.92

It is a known fact that features play a key role in increasing the accuracy of a dataset when any machine learning model is applied. Also, for calculating the grades/ total performance, we need quality features / questionnaires / best teaching practices where we can take reference and prepare our questions/ features in a synthetic dataset. In paper [10], "Development and validation of an observation protocol for measuring science teachers' modeling-based teaching performance" details the development and testing of a tool to assess the effectiveness of science teachers' modeling-based teaching (MBT) practices. Modeling-based teaching practices typically consist of the 5E's—Engage, Explore, Explain, Elaborate, and Evaluate. This basically means that the teacher should bring a problem that will truly engage the students to explore, followed by explaining their understanding of the problem to each other, elaborate their understanding by applying that problem to new situations, connecting those understandings with real-world scenarios, and finally, the teacher should evaluate their understanding. The study involves creating an observation protocol based on a review of the literature and interviews with science education professionals. The methodology was tested in real-world classroom situations and proven to be valid and reliable for evaluating MBT practices. According to the authors, this instrument can be used to improve scientific teacher training and professional growth. Similarly, in our synthetic dataset, we considered the questions that would yield better results in predicting the performance of the Instructor. We randomly filled the data for each entry and developed a hybrid machine learning framework that was reviewed by Dr. Saleti Sumalatha. The next step after review is data preprocessing and implementation of our protocol/ framework.

In paper [11], "Addressing the feasibility of the teacher performance rate and accuracy scale as a treatment integrity tool" examines the effectiveness of a tool designed to measure the fidelity of implementation of a behavioral intervention by teachers. The tool in question is the Teacher Performance Rate and Accuracy Scale (TPRAS), developed to assess how accurately and consistently teachers implement a behavioral intervention. The study found TPRAS feasible and effective for measuring treatment integrity, ensuring quality of implementation, and supporting teacher professional development. Simply, it shows that the TPRA scale is useful to supervisors of classroom instruction because it provides objective feedback on teacher performance, tracks skill acquisition, and offers a permanent product of both the student's and teacher's progress. To summarize, the authors basically employed a methodology in which a group of individuals visits the classrooms, gathers student feedback, and evaluates the instructor's performance. Similarly, in our synthetic dataset, we considered the questions from the student portal of SRMAP, where students give feedback on their professors. The questions are framed by a group of non-teaching professionals in the academic department who monitor classroom progress, then gather feedback from students. The data is generated randomly and is trained and tested using a machine learning framework (a fusion of Boosting and Recursive Feature Elimination).

On the roadmap to finding better features for preparing a synthetic dataset, [12] shows best practices such that feedback on any instructor is maximized. In [12], ”Supporting students’ meaningful engagement in scientific modeling through epistemological messages: A case study of contrasting teaching approaches” looks at how epistemological messages can help students engage in scientific modeling. The study contrasts two teaching styles—one emphasizing epistemological messages and one that does not—observing their impact on student participation in modeling practices. It found that students who received epistemological lessons had stronger engagement and knowledge of scientific modeling than those who did not. The authors believe this approach can help students build scientific modeling skills and that it can be included in science education to improve the quality of scientific learning. Paper [12] basically helped us discover effective questions to evaluate instructors’ performance at an initial level. It highlighted the need for at least two contrasting levels of teaching techniques and teacher–student interaction questions (such as “Is the teacher fair and unbiased?”) to evaluate an instructor’s performance more thoroughly.

# Chapter 4

# DESIGN AND METHODOLOGY

## 4.1 DESIGN

### 4.1.1 Cross-Dataset Methodology Workflow

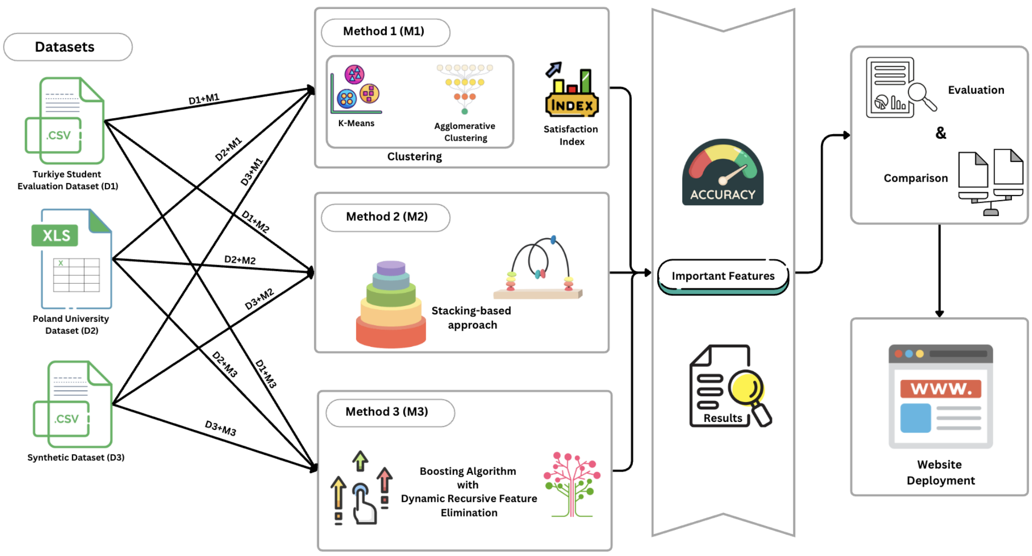



Figure 4.1: Overview of the cross-dataset methodology design

Each dataset is evaluated using three different methods — clustering and satisfaction index (M1), stacking-based approach (M2), and boosting with dynamic RFE (M3). The process leads to accuracy evaluation, important feature extraction, comparative analysis, and final deployment on a website.

## 4.2 METHODOLOGY

### 4.2.1 Dataset Description

Prediction of Instructor Performance is the project where we obtain a dataset of the feedback forms sent to the students about their instructor, and we pick a dependent variable to decide whether the instructor's performance is up to the mark (or not) using various Machine Learning Models.

In this work, we used **three** datasets:

1. **Turkiye Student Evaluation Dataset [16]**

   The dataset has been obtained from the UC Irvine Machine Learning Repository. It is a real-world dataset that was donated on 2013-09-01. It contains 5820 records and 33 columns, 28 of which are critical questions relevant to the effectiveness of the instructor in teaching the course.

   **Dataset columns include:**

   - `instr`: Instructor's identifier; values from {1,2,3}
   - `class`: Course code (descriptor); values from {1–13}
   - `repeat`: Number of times the student is taking this course; values from {0,1,2,3,. . . }
   - `attendance`: Code of the level of attendance; values from {0,1,2,3,4}
   - `difficulty`: Level of difficulty of the course (1–5)
   - `Q1–Q28`: Survey questions about various aspects of the instructor's teaching, preparedness, etc.

   The datatype of all attributes is `int`, and there were no missing values.

2. **Poland University Dataset**

   This dataset consists of Student Evaluation of Teaching (SET) ratings provided by students from a university in Poland at the end of the winter semester of the 2020/2021 academic year. It has 8015 records in total. The survey has 9 main questions (Q1–Q9), each answered on a 1–5 scale (Strongly disagree to Strongly agree).

   **Questions include:**

   - Q1. I learnt a lot during the course.
   - Q2. The knowledge acquired is very useful.
   - Q3. The professor used activities to make the class more engaging.
   - Q4. I would enroll again in a course taught by this lecturer.
   - Q5. The classes started on time.
   - Q6. The lecturer always used time efficiently.
   - Q7. The lecturer delivered the class content in an understandable way.
   - Q8. The lecturer was available when we had doubts.
   - Q9. The lecturer treated all students equally, regardless of race/ethnicity/backgrounc

   Additional attributes include: seniority of the instructor, gender, average SET score, number of degrees, student grade average, and the percentage of students who passed the course.

3. **Synthetic Dataset**

   A synthetic dataset was created with existing knowledge from reference papers [10], [11], [12]. In this dataset, rows were randomly generated for all the questions/columns, and stored for further analysis and machine learning implementation.

The dataset has **11 columns** (11 questions), such as:

(a) Is fully prepared and completes all syllabus topics in detail.

(b) Encourages participation and discussion in the class.

(c) Uses modern teaching aids/gadgets (animation, video, simulation, etc.) in the class and shares handouts/references/PPT/web-resources.

(d) Relates the course content with real-world scenarios to improve interest and understanding.

(e) Inspires me by his/her teaching competency (making the class interesting).

(f) Communicates the desired knowledge by adopting a learner-centric approach.

(g) Is dedicated to teaching and punctual in class.

(h) Is a role model and life-long mentor.

(i) Sets an effective question paper using standard principles of honesty, dedication, and integrity.

(j) Conducted periodical assessments as per schedule (quiz, seminars, assignments, etc.).

(k) Is fair and unbiased throughout the evaluation process.

We let a user specify a number $n$, after which $3 \times n$ entries were generated (we set $n = 3000$ for a total of 9000 records).

# Chapter 5

# IMPLEMENTATION

## 5.1 DATA PREPROCESSING

**1. Turkiye Student Evaluation Dataset:** The data cleaning process was started off by a checking for any null values and outliers. Boxplot visualization of the data has been done to understand the range of values and to identify outliers for all the attributes and deal with them by applying thresholds.

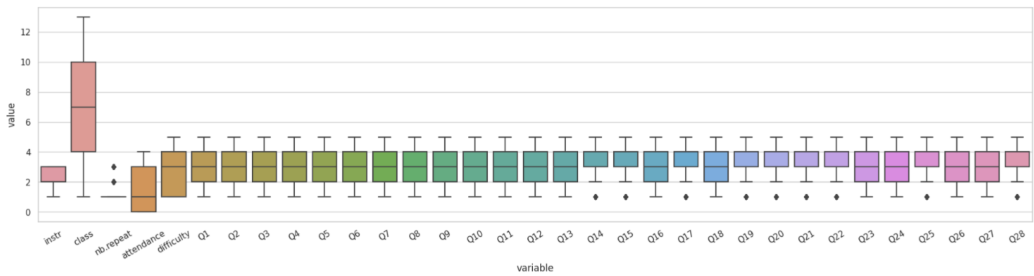



Figure 5.1: Boxplot for Data Visualization

Upon dealing with the outliers and normalizing the values it was discovered that more than 1000 datapoints were lost. Therefore, data before removing outliers was used for further steps and investigations. Moving on, one attribute "nb.repeat" has been dropped as its values were mostly constant throughout the data. Analysis of this data stems into two approaches from this step onwards.

**Approach 1 (Satisfaction Index):** `Satisfaction_Index` was introduced as a new attribute and calculated as the average of Q9 and Q10. This approach to compute the target variable was inspired by the work in [13]. The resulting values were then segregated into three classes:

- `Satisfaction_Index` $< 3 \rightarrow$ Class 0
- `Satisfaction_Index` $= 3 \rightarrow$ Class 1
- `Satisfaction_Index` $> 3 \rightarrow$ Class 2

A new column named `Recommend_Instructor` was created to store the class value for each datapoint. After dropping duplicate entries, the shape of the dataframe became (3717, 34). This step ensures prevention of data leakage into the test set and helps evaluate the model fairly on unseen data.

To address class imbalance, oversampling was performed using the SMOTE method at the end of this step.

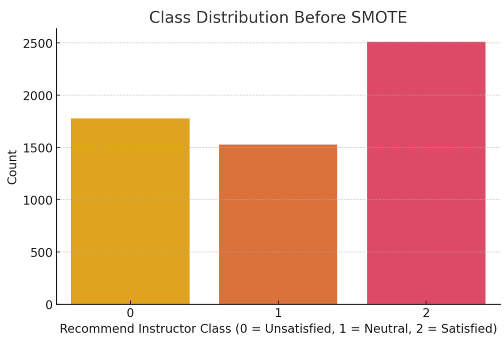



Figure 5.2: Class Distribution Before SMOTE

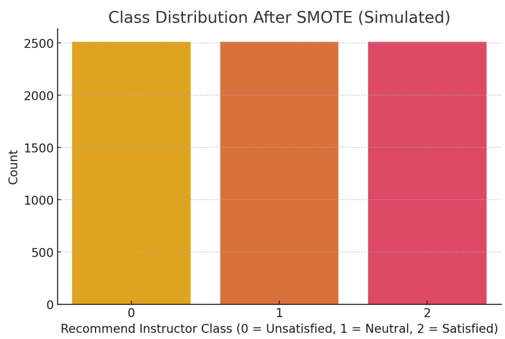



Figure 5.3: Class Distribution After SMOTE (Simulated)

**Approach 2 (Clustering):** This will be introduced in the next subsection (Machine Learning models applied).

**2. Poland University Dataset:** Missing values have been updated with the mean value of its respective attributes. A correlation heatmap for all the attributes has been generated and attributes with strong positive and negative correlation have been dropped since it reduces redundancy and can improve the accuracy.

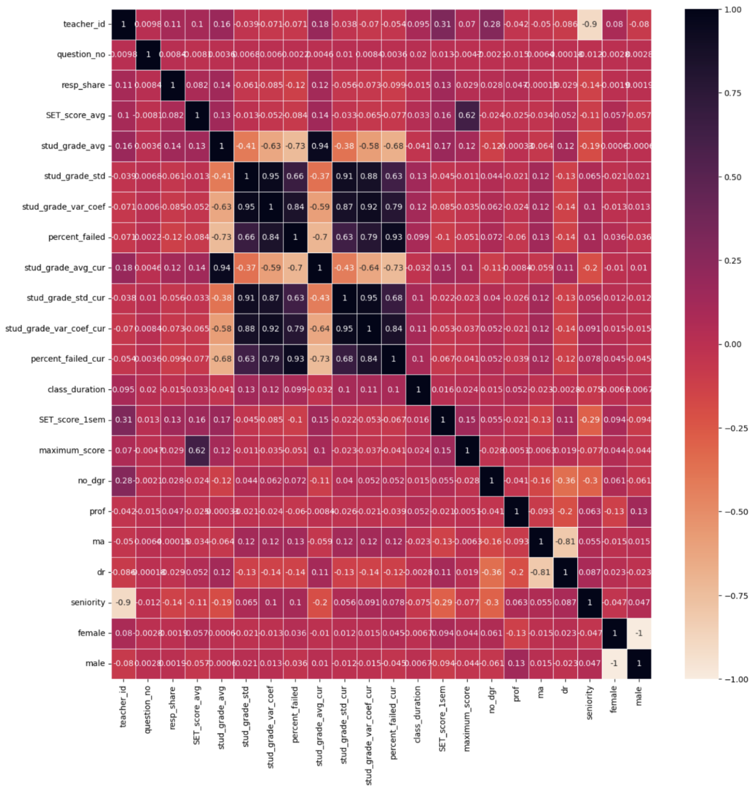


Figure 5.4: Correlation heatmap

There are two different attributes for gender male and female which have been dropped and a single attribute called gender has been introduced. A new attribute was created called `Result` which is assigned to 1 signifying that the performance of the instructor is satisfactory and it is assigned to 0 when the performance is unsatisfactory. The values for the `Result` attribute is set to 1 if the average SET score for the instructor is greater than the mean value of average SET score, it is set to 0 otherwise. An imbalance in the target variables has been identified. To rectify this problem, SMOTE has been used to balance out the dataset.

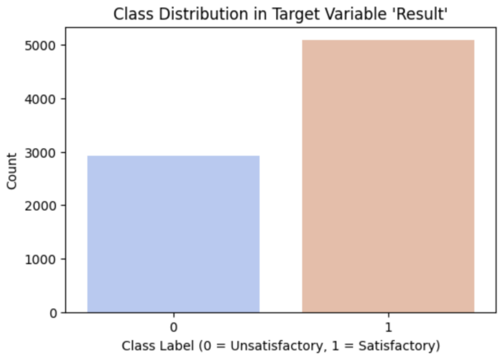



Figure 5.5: Class Distribution in Target Variable 'Result'

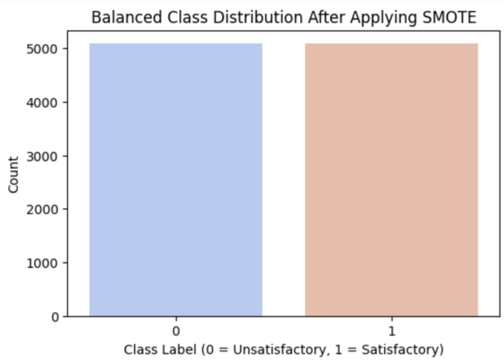



Figure 5.6: Balanced Class Distribution After Applying SMOTE

**3. Synthetic Dataset:** Firstly, as said, 3×n number of data entries will be generated randomly, where the final dataset consists of an equal number of higher ratings, lower ratings and mixed ratings. Then, the dataset is shuffled such that any of the Machine Learning Models won't easily predict the values. "Is a role model and life-long mentor for me" is the initial target variable, and if the value in any row is greater than the column mean, then we update the target variable with 1, else 0.

## 5.2 MACHINE LEARNING MODELS APPLIED

**1. Turkiye Student Evaluation Dataset**

**Approach 1 (Satisfaction Index):** Furthermore, the attributes Satisfaction_Index, Q9, and Q10 were dropped from the final dataframe, and Recommend_Instructor was chosen as the target variable. The data was split into train and test datasets using sklearn.train_test_split. Multiple models such as Logistic Regression, Decision Tree Classifier, Bagging Classifier, Random Forest Classifier, AdaBoost Classifier, XGBoost Classifier, Voting Classifier (Hard and Soft), and Support Vector Machine (SVM) were applied to the training data and tested against the test dataset for easier and unbiased comparison.

Upon individual training, the maximum accuracy was achieved by the AdaBoost Classifier with **84.4%**, while the lowest accuracy was observed for Soft Voting Classification at **77.4%**.

In addition, Recursive Feature Elimination (RFE) using Logistic Regression as the estimator, combined with Stratified K-Fold Cross-Validation, was applied to the above-mentioned models to observe changes in accuracy. The hyperparameters used for RFE were n_features_to_select = 25, and for Stratified K-Fold Cross-Validation, n_splits = 10, n_repeats = 3. The highest and lowest accuracies in this iteration were obtained with the Random Forest Classifier and Decision Tree Classifier at **86.9%** and **80.6%**, respectively.

**Approach 2 (Clustering):** This approach was inspired by the work in [2]. K-means clustering and agglomerative clustering algorithms were applied on the raw data, after the duplicates were dropped. In order to determine the number of clusters, the Elbow method and Silhouette scores method were used.

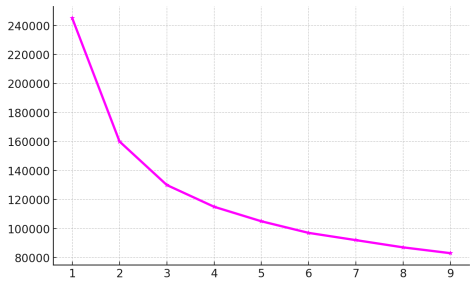



Figure 5.7: Elbow method

```
----------------------------------------
KMeans(n_clusters=2, random_state=42)
Silhouette score: 0.2619927902351312
----------------------------------------
KMeans(n_clusters=3, random_state=42)
Silhouette score: 0.2080007244242248
----------------------------------------
KMeans(n_clusters=4, random_state=42)
Silhouette score: 0.2101006490944269
----------------------------------------
KMeans(n_clusters=5, random_state=42)
Silhouette score: 0.1901365158081606
----------------------------------------
KMeans(n_clusters=6, random_state=42)
Silhouette score: 0.1893246483518626
----------------------------------------
KMeans(n_clusters=7, random_state=42)
Silhouette score: 0.1757740134170610
----------------------------------------
KMeans(n_clusters=8, random_state=42)
Silhouette score: 0.1633859215384840
----------------------------------------
KMeans(n_clusters=9, random_state=42)
Silhouette score: 0.1519589925472606
----------------------------------------
```

Figure 5.8: Silhouette scores

It was inferred from the plots and Silhouette scores that 2 clusters are to be applied for both K-means clustering and agglomerative clustering models. The lists of predicted cluster labels were added as new attributes to their respective data frames. The data was split into train and test sets, with “cluster_label” as the target variable, followed by fitting numerous classification models such as Logistic Regression, Decision Tree Classifier, Bagging Classifier, Random Forest Classifier, AdaBoost Classifier, XGBoost Classifier, Voting Classifier (Hard and Soft) and SVM individually, with the training dataset. Furthermore, these individual models were added to a Pipeline that consisted of Recursive Feature Elimination followed by Stratified k fold cross-validation. The results of these experiments will be presented in the Discussions section.

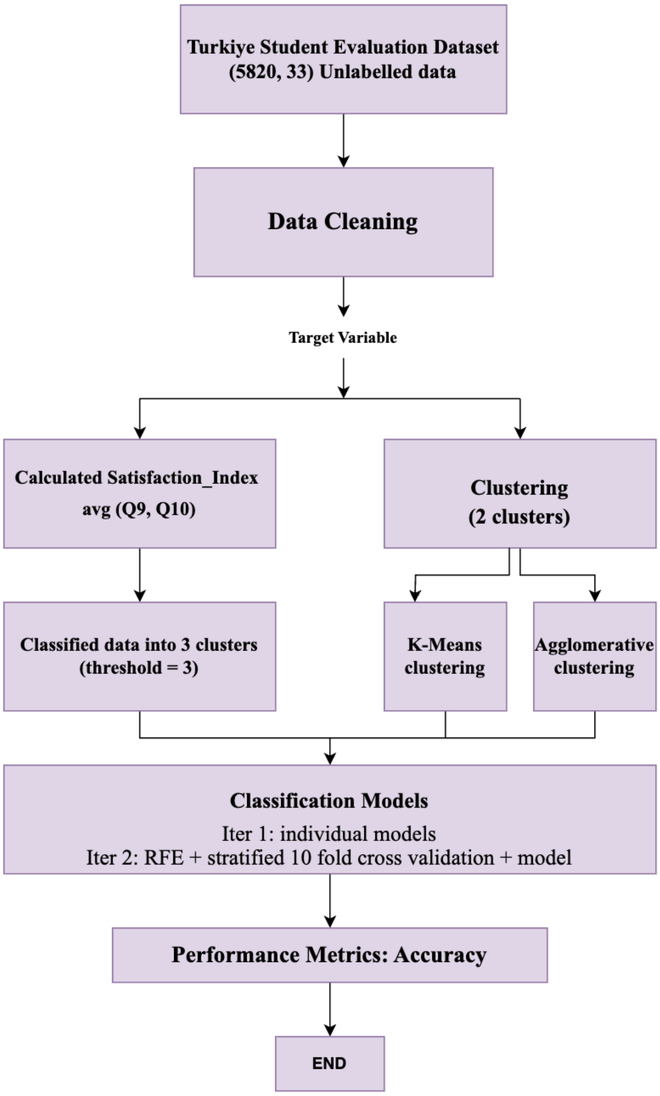



Figure 5.9: Flowchart for Turkiye Student Evaluation dataset all approaches

**2. Poland University Dataset:** Attributes including student average grade over the last 6 semesters, Student grade Standard Deviation over the last 6 semesters, Percentage of students failed over the last 6 semesters, student average grade last semester, Student grade Standard Deviation past semester, Percentage of students failed last semester, number of degrees the instructor has, whether the instructor has professorship, the instructor has a masters degree, the instructor has a doctorate degree, total years of experience in teaching and the gender are taken as dependant variables and Result taken as the target variable. Different classification models have been used like logistic regression, decision tree classification, Naive Bayes Classification, Random Forest Classification were applied on the dataset. The highest accuracy for individual classifiers observed was 66% with random forest classifiers with recursive feature elimination. A stacking based approach was used with the first layer consisting of Decision Tree Classifier, Random Forest Classifier, KNN and XGBoost and the second layer consisting of logistic regression which yielded the highest accuracy of 78.5% among the different approaches applied.

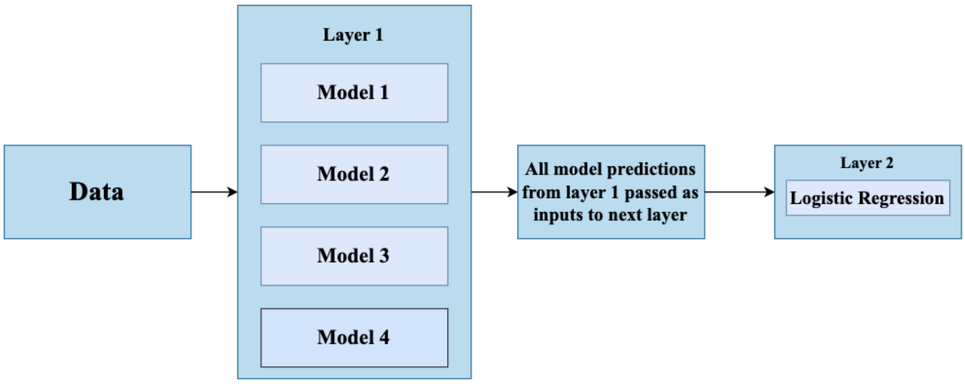



Figure 5.10: Stacking-based approach flowchart used in Poland dataset

**3. Synthetic Dataset:** A new Hybrid Model is introduced by making the fusion of Boosting approach and recursive feature elimination method where, First, a Machine learning Model is applied to the dataset and then by including the predicted values of the first model, we add a new feature "First_Prediction" into the dataset and apply second machine learning Model, This is how the boosting approach works as said in [14]. Accuracies are calculated after applying both the Machine Learning Models. For each model applied, the best accuracy is taken into consideration as we are doing recursive feature elimination to gain the best accuracy. Recursive Feature

Elimination is done with the help of applying Linear Regression between each Independent variable to the dependent variable of the dataset and R-square values are calculated for each regression. Also, features with the least R-squared values are eliminated iteratively and this process is stopped if the current accuracy after removing any feature is less than the previous accuracy of the model applied or 100% accuracy is gained as explained in [15]. The Recursive Feature Elimination is applied for both the machine learning models applied.

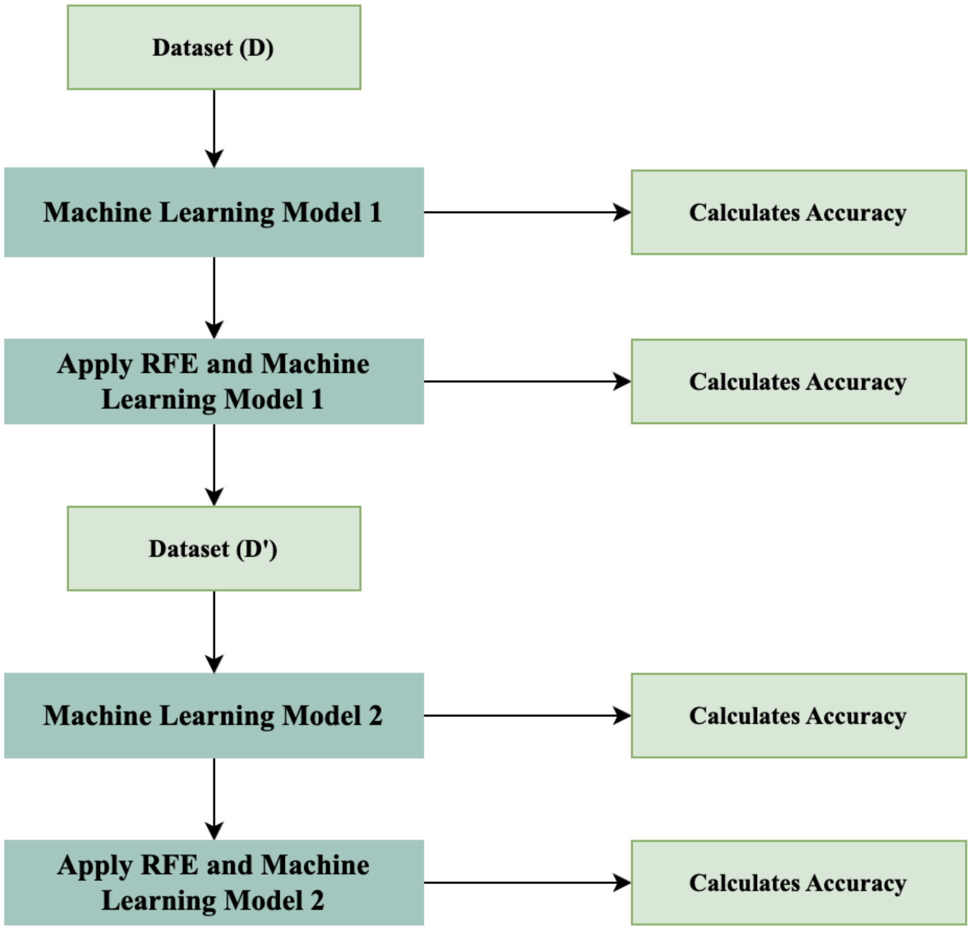



Figure 5.11: Boosting Algorithm with Dynamic Recursive Feature Elimination

The Machine Learning Models used -

- Decision Tree
- Logistic Regression
- Support Vector Machine
- AdaBoost Classifier
- Multi-Layer Perceptron (MLP) Classifier
- Gradient Boosting Classifier
- XGBoost Classifier

We get 49 combinations using the 7 Machine Learning Models above mentioned and it is observed to have the highest accuracy as 90.15%.

# Chapter 6

# HARDWARE/ SOFTWARE TOOLS USED

Table 6.1: Software Tools Used

| Category | Tools and Purpose |
|---|---|
| Programming Languages | • Python – ML model implementation and preprocessing<br>• JavaScript – Front-end development<br>• Node.js – Backend logic and API handling<br>• HTML & CSS – Structuring and styling the web interface |
| Libraries & Frameworks | • scikit-learn – Model training, evaluation, RFE<br>• pandas, NumPy – Data handling and manipulation<br>• matplotlib, seaborn – Data visualization<br>• XGBoost – Advanced boosting classifier<br>• SMOTE – Class imbalance handling<br>• Express.js – Node.js API framework<br>• Mongoose – MongoDB connection |
| Database | • MongoDB – Stores user data and prediction results |
| Development Tools | • Visual Studio Code – Code editor<br>• Google Colab – ML model execution<br>• Postman – API testing<br>• Git & GitHub – Version control and collaboration<br>• CSV Handling – Dataset storage and loading<br>• Word / LaTeX – Report writing and formatting |
| Authentication | • Google OAuth – Google account login integration |

Table 6.2: Hardware Tools Used

| Component | Description |
|---|---|
| Development Machine | Laptop/PC with Intel i5/i7 or AMD Ryzen 5+, SSD, minimum 8 GB RAM |
| Operating System | macOS, Windows, or Linux – used as development environments |
| Cloud Platform | Google Colab – Used for cloud-based Python execution and model training |
| External Tools | • External Storage – For backups or large datasets<br>• Internet Connection – Required for APIs, cloud tools, and deployment |

# Chapter 7

# RESULTS & DISCUSSION

## 1. Turkiye Student Evaluation Dataset:

Table 7.1: Results for Turkiye Student Evaluation Dataset

| Classifier | Satisfaction Index | | K-means Clustering | | Agglomerative Clustering | |
|---|---|---|---|---|---|---|
| | Individual Model | RFE + CV | Individual Model | RFE + CV | Individual Model | RFE + CV |
| Logistic Regression | 81.7 | 81.5 | **99.4** | 98.5 | 95.4 | 98.5 |
| Decision Tree Classifier | 77.4 | 80.6 | 94.8 | 95.1 | 92.5 | 95.1 |
| Bagging Classifier | 80.7 | 84.7 | 96.8 | 96.5 | 95.1 | 96.5 |
| Random Forest Classifier | 77.4 | **86.9** | 94.8 | 97.7 | 92.5 | 97.7 |
| AdaBoost Classifier | **84.4** | 83.0 | 97.4 | 97.8 | 94.8 | 97.8 |
| XGBoost Classifier | 83.6 | 86.1 | 98.1 | 97.6 | 96.4 | 97.6 |
| SVM | 82.9 | 82.2 | 99.0 | 98.4 | 95.4 | 98.4 |
| Voting Classifier (Soft) | 83.7 | – | 98.8 | – | 96.1 | – |
| Voting Classifier (Hard) | 82.6 | – | 98.9 | – | 96.5 | – |

**Note:** The bold numbers in the table highlight the model with the highest accuracy for a given iteration.

**Hyperparameters used to obtain the results in the above table:**

- Recursive Feature Elimination (RFE): `n_features_to_select = 25`
- Stratified K-Fold Cross Validation: `n_splits = 10`, `n_repeats = 3`

**Approach 1 (Satisfaction Index):** The best performing individual model was **AdaBoost Classifier** with an accuracy of **84.4%**. Upon applying RFE and Stratified K-Fold Cross Validation, the **Random Forest Classifier** achieved the highest accuracy of **86.9%**, which is comparatively better than the rest.

**Approach 2 (Clustering):**

- **K-Means Clustering:** The highest accuracy for both the individual model and the RFE + CV approach was achieved by **Logistic Regression**, with accuracies of **99.4%** and **98.5%**, respectively.

- **Agglomerative Clustering:** The **Voting Classifier (Hard Voting)** achieved the highest accuracy at **96.5%**, which was just **0.1%** higher than the **XGBoost Classifier** (96.4%). Meanwhile, the RFE + CV + Logistic Regression pipeline also performed strongly, with an accuracy of **98.5%**.

Although the use of RFE and Stratified K-Fold Cross Validation increased the accuracy of most models, there were cases where the individual models performed better than the pipeline.

Several additional experiments were performed to observe the change in accuracy scores by tweaking the hyperparameters: `n_features_to_select`, `n_splits`, and `n_repeats`. It was observed that:

- Decreasing `n_features_to_select` led to a drop in accuracy.
- Changing `n_splits` and `n_repeats` had no significant impact on accuracy.
- However, increasing these values significantly increased the total execution time.

Therefore, the values `n_splits = 10` and `n_repeats = 3` were retained for all experiments.

**Conclusion:** It can be clearly ascertained from the table that **K-Means Clustering with two clusters** outperforms **Agglomerative Clustering with two clusters** in terms of accuracy for the Turkiye Student Evaluation Dataset.

**2. Poland University Dataset:**

Table 7.2: Results for Poland University Dataset

| **Classifier** | **Performance (Accuracy)** | |
|---|---|---|
| | `N_splits = 10` | `n_repeats = 3` |
| Logistic Regression | 56 | 61 |
| Decision Tree Classifier | 63 | 64 |
| Bagging Classifier | **67** | 66 |
| Random Forest Classifier | 66.5 | **66.6** |
| AdaBoost Classifier | 58 | 61 |
| XGBoost Classifier | 54 | 66 |
| SVM | 54 | 59 |
| Voting Classifier | 64 | – |
| Stacking Classifier | **78.5** | – |

The highest accuracy for this dataset was observed for Bagging Classifier with 67% accuracy. Random forest classifier combined with recursive feature elimination and stratified k fold cross validation yielded 66.6% accuracy. Apart from these models, the model that has been introduced in the previous sections, a stacking based approach was developed in which the first layer consisted of Decision Tree Classifier, Random Forest Classifier, KNN Classifier and XGBoost Classifier. And their predicted values were taken as columns in a new dataset. A logistic regression model was trained on all the outputs of the first layer with results as the target variable. This yielded an accuracy of 78.5%.

**3. Synthetic Dataset:** For the synthetic dataset, the main aim was to collect the dataset of the responses that are given by the students in any university every semester for their professors. Unfortunately, due to some privacy concerns from the universities, the feedback given by students in their universities upon their lecturers couldn't be shared, a dataset with random values is generated where 33% of the data generates a positive result, 33% of the data generates a negative result and 33% of the data generates a mixed results about the professors. While working to fix the Target variable for the dataset, the initial thought is to proceed by assigning a weight for each dependent variable (question) and generate the target variable by doing weighted average of each entry and if the weighted average is greater than the mean of the target variable, then, it's a Yes, else no. For any machine learning model applied, the generated dataset resulted in producing very high accuracy values ( 99% accuracy) which was suspicious and doubted whether the Machine Learning Model found it very easy to predict the Independent feature and decided to change the target variable. In the process, revised many articles, papers, etc, the papers [10][11],[12] helped in understanding the problem in a better way and even understood how Instructor performance is being evaluated in the real world scenarios and found some matching qualities in the feedback form given in SRM University, AP and then decided to move on with the questions present in the feedback form and after much discussion among the group, "is a role model and life-long mentor for me" is considered as the target variable. Now, while implementing the boosting approach with RFE using SKLearn for synthetic dataset generated above, it was found that the Highest accuracies are jumping around 72% where machine learning models like Decision Tree with RFE, Logistic Regression with RFE, Support Vector Machines with RFE, Adaboost Classifier with RFE Gradient Boosting Classifier with RFE are being used. Below here is a small picture depicting the graph which shows the accuracy with different combinations of Machine Learning Models.

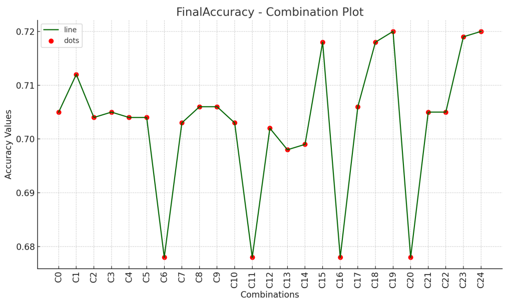



Figure 7.1: Graph depicting Accuracies after applying Boosting Approach with Recursive Feature Elimination using Sklearn on the Synthetic Dataset

```
Combination 1  –  Decision Tree and Decision Tree
Combination 2  –  Decision Tree and Logistic Regression
Combination 3  –  Decision Tree and Support Vector Machines
Combination 4  –  Decision Tree and Adaboost Classifier
Combination 5  –  Decision Tree and Gradient Boosting Classifier
Combination 6  –  Logistic Regression and Decision Tree
Combination 7  –  Logistic Regression and Logistic Regression
Combination 8  –  Logistic Regression and Support Vector Machines
Combination 9  –  Logistic Regression and Adaboost Classifier
Combination 10 –  Logistic Regression and Gradient Boosting Classifier
Combination 11 –  Support Vector Machines and Decision Tree
Combination 12 –  Support Vector Machines and Logistic Regression
Combination 13 –  Support Vector Machines and Support Vector Machines
Combination 14 –  Support Vector Machines and Adaboost Classifier
Combination 15 –  Support Vector Machines and Gradient Boosting Classifier
Combination 16 –  Adaboost Classifier and Decision Tree
Combination 17 –  Adaboost Classifier and Logistic Regression
Combination 18 –  Adaboost Classifier and Support Vector Machines
Combination 19 –  Adaboost Classifier and Adaboost Classifier
Combination 20 –  Adaboost Classifier and Gradient Boosting Classifier
Combination 21 –  Gradient Boosting Classifier and Decision Tree
Combination 22 –  Gradient Boosting Classifier and Logistic Regression
Combination 23 –  Gradient Boosting Classifier and Support Vector Machines
Combination 24 –  Gradient Boosting Classifier and Adaboost Classifier
Combination 25 –  Gradient Boosting Classifier and Gradient Boosting Classifier
```

Figure 7.2: Combinations of Machine Learning Models involved in Boosting Approach with Recursive Feature Elimination using Sklearn

On exploring the ways to increase the accuracy, implemented Dynamic Recursive Feature Elimination with the help of Linear Regression in an analytical approach. Basically, Linear Regression is applied on each column with respect to Target Variable and started eliminating the columns with least accuracies. If the accuracy of the dataset after eliminating any feature is less than the previous accuracy observed, then, recursive feature elimination will be terminated. Recursive feature elimination is even stopped if the accuracy is 100% and even if there are no columns to get eliminated. From Table 3, it is observed that, initially, the accuracies are around 67.12% - 72% in all the combinations. By applying the boosting approach with dynamic implementation of RFE using analytical approach as explained above, the accuracy is increased to 90.15%. The least accuracy in any combination is 72.12% which is also greater than the initial accuracy observed.

All Combinations:-

Table 7.3: Results for Synthetic Dataset

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 0 | Decision Tree | 67.12 | 67.16 | Decision Tree | 80.71 | 80.71 |
| 1 | Decision Tree | 67.59 | 67.59 | Logistic Regression | 89.85 | 89.85 |
| **2** | **Decision Tree** | **67.25** | **67.73** | **Support Vector Machines** | **90.15** | **90.15** |
| 3 | Decision Tree | 67.46 | 67.66 | Adaboost Classifier | 90.04 | 90.04 |
| 4 | Decision Tree | 67.23 | 67.37 | MLP Classifier | 89.90 | 89.90 |
| 5 | Decision Tree | 67.37 | 67.73 | Gradient Boosting Classifier | 89.69 | 89.69 |
| 6 | Decision Tree | 67.84 | 67.84 | XG Boost | 88.38 | 88.38 |
| 7 | Logistic Regression | 72.78 | 72.78 | Decision Tree | 66.32 | 72.78 |
| 8 | Logistic Regression | 72.78 | 72.78 | Logistic Regression | 73.57 | 73.57 |
| 9 | Logistic Regression | 72.78 | 72.78 | Support Vector Machines | 73.69 | 73.69 |
| 10 | Logistic Regression | 72.78 | 72.78 | Adaboost Classifier | 72.41 | 72.78 |
| 11 | Logistic Regression | 72.78 | 72.78 | MLP Classifier | 72.89 | 72.89 |
| 12 | Logistic Regression | 72.78 | 72.78 | Gradient Boosting Classifier | 73.94 | 73.94 |
| 13 | Logistic Regression | 72.78 | 72.78 | XG Boost | 71.89 | 72.78 |
| 14 | Support Vector Machines | 72.19 | 72.66 | Decision Tree | 67.55 | 72.66 |
| 15 | Support Vector Machines | 72.19 | 72.66 | Logistic Regression | 73.53 | 73.53 |
| 16 | Support Vector Machines | 72.19 | 72.66 | Support Vector Machines | 73.64 | 73.64 |
| 17 | Support Vector Machines | 72.19 | 72.66 | Adaboost Classifier | 73.76 | 73.76 |
| 18 | Support Vector Machines | 72.19 | 72.66 | MLP Classifier | 72.21 | 72.66 |
| 19 | Support Vector Machines | 72.19 | 72.66 | Gradient Boosting Classifier | 72.57 | 72.66 |
| 20 | Support Vector Machines | 72.19 | 72.66 | XG Boost | 71.39 | 72.66 |
| 21 | Adaboost Classifier | 72.05 | 72.12 | Decision Tree | 67.41 | 72.12 |
| 22 | Adaboost Classifier | 72.05 | 72.12 | Logistic Regression | 73.35 | 73.35 |

Continued on next page

Table 7.3 – continued from previous page

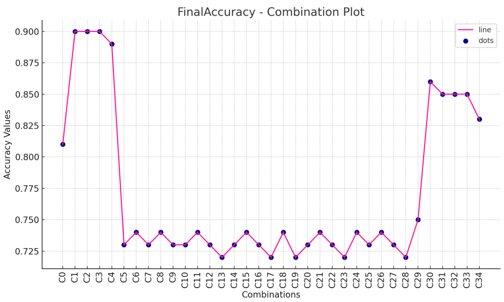
| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 23 | Adaboost Classifier | 72.05 | 72.12 | Support Vector Machines | 73.89 | 73.89 |
| 24 | Adaboost Classifier | 72.05 | 72.12 | Adaboost Classifier | 72.46 | 72.46 |
| 25 | Adaboost Classifier | 72.05 | 72.12 | MLP Classifier | 71.91 | 72.12 |
| 26 | Adaboost Classifier | 72.05 | 72.12 | Gradient Boosting Classifier | 72.94 | 72.94 |
| 27 | Adaboost Classifier | 72.05 | 72.12 | XG Boost | 71.87 | 72.12 |
| 28 | MLP Classifier | 70.94 | 72.37 | Decision Tree | 71.30 | 72.37 |
| 29 | MLP Classifier | 72.14 | 72.14 | Logistic Regression | 73.26 | 73.26 |
| 30 | MLP Classifier | 71.46 | 72.51 | Support Vector Machines | 74.62 | 74.62 |
| 31 | MLP Classifier | 70.60 | 71.91 | Adaboost Classifier | 73.19 | 73.19 |
| 32 | MLP Classifier | 71.69 | 72.14 | MLP Classifier | 73.39 | 73.39 |
| 33 | MLP Classifier | 71.14 | 72.32 | Gradient Boosting Classifier | 73.41 | 73.41 |
| 34 | MLP Classifier | 71.21 | 72.30 | XG Boost | 73.03 | 73.03 |
| 35 | Gradient Boosting Classifier | 72.01 | 72.01 | Decision Tree | 68.19 | 72.01 |
| 36 | Gradient Boosting Classifier | 72.01 | 72.01 | Logistic Regression | 73.44 | 73.44 |
| 37 | Gradient Boosting Classifier | 72.01 | 72.01 | Support Vector Machines | 74.12 | 74.12 |
| 38 | Gradient Boosting Classifier | 72.01 | 72.01 | Adaboost Classifier | 74.03 | 74.03 |
| 39 | Gradient Boosting Classifier | 72.01 | 72.01 | MLP Classifier | 73.53 | 73.53 |
| 40 | Gradient Boosting Classifier | 72.01 | 72.01 | Gradient Boosting Classifier | 72.82 | 72.82 |
| 41 | *Gradient Boosting Classifier* | 72.01 | 72.01 | *XG Boost* | 70.87 | 72.01 |
| 42 | XG Boost | 71.89 | 71.96 | Decision Tree | 75.17 | 75.17 |
| 43 | XG Boost | 71.89 | 71.96 | Logistic Regression | 85.53 | 85.53 |
| 44 | XG Boost | 71.89 | 71.96 | Support Vector Machines | 84.42 | 84.42 |
| 45 | XG Boost | 71.89 | 71.96 | Adaboost Classifier | 84.37 | 84.37 |
| 46 | XG Boost | 71.89 | 71.96 | MLP Classifier | 84.58 | 84.58 |
| 47 | XG Boost | 71.89 | 71.96 | Gradient Boosting Classifier | 84.81 | 84.81 |

Continued on next page

Table 7.3 – continued from previous page

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 48 | XG Boost | 71.89 | 71.96 | XG Boost | 82.90 | 82.90 |

**Note:** The **bold numbers** in the above table highlight the highest accuracies, and the *italicized numbers* highlight some of the lowest accuracies.




(a) Graph depicting Accuracies after applying Boosting Approach with Dynamic Recursive Feature Elimination using analytical approach on the Synthetic Dataset

```
Combination 1  - Decision Tree and Decision Tree
Combination 2  - Decision Tree and Logistic Regression
Combination 3  - Decision Tree and Support Vector Machines
Combination 4  - Decision Tree and Adaboost Classifier
Combination 5  - Decision Tree and MLP Classifier
Combination 6  - Decision Tree and Gradient Boosting Classifier
Combination 7  - Decision Tree and XG Boost
Combination 8  - Logistic Regression and Decision Tree
Combination 9  - Logistic Regression and Logistic Regression
Combination 10 - Logistic Regression and Support Vector Machines
Combination 11 - Logistic Regression and Adaboost Classifier
Combination 12 - Logistic Regression and MLP Classifier
Combination 13 - Logistic Regression and Gradient Boosting Classifier
Combination 14 - Logistic Regression and XG Boost
Combination 15 - Support Vector Machines and Decision Tree
Combination 16 - Support Vector Machines and Logistic Regression
Combination 17 - Support Vector Machines and Support Vector Machines
Combination 18 - Support Vector Machines and Adaboost Classifier
Combination 19 - Support Vector Machines and MLP Classifier
Combination 20 - Support Vector Machines and Gradient Boosting Classifier
Combination 21 - Support Vector Machines and XG Boost
Combination 22 - Adaboost Classifier and Decision Tree
Combination 23 - Adaboost Classifier and Logistic Regression
Combination 24 - Adaboost Classifier and Support Vector Machines
Combination 25 - Adaboost Classifier and Adaboost Classifier
Combination 26 - Adaboost Classifier and MLP Classifier
Combination 27 - Adaboost Classifier and Gradient Boosting Classifier
Combination 28 - Adaboost Classifier and XG Boost
Combination 29 - MLP Classifier and Decision Tree
Combination 30 - MLP Classifier and Logistic Regression
Combination 31 - MLP Classifier and Support Vector Machines
Combination 32 - MLP Classifier and Adaboost Classifier
Combination 33 - MLP Classifier and MLP Classifier
Combination 34 - MLP Classifier and Gradient Boosting Classifier
Combination 35 - MLP Classifier and XG Boost
Combination 36 - Gradient Boosting Classifier and Decision Tree
Combination 37 - Gradient Boosting Classifier and Logistic Regression
Combination 38 - Gradient Boosting Classifier and Support Vector Machines
Combination 39 - Gradient Boosting Classifier and Adaboost Classifier
Combination 40 - Gradient Boosting Classifier and MLP Classifier
Combination 41 - Gradient Boosting Classifier and Gradient Boosting Classifier
Combination 42 - Gradient Boosting Classifier and XG Boost
Combination 43 - XG Boost and Decision Tree
Combination 44 - XG Boost and Logistic Regression
Combination 45 - XG Boost and Support Vector Machines
Combination 46 - XG Boost and Adaboost Classifier
Combination 47 - XG Boost and MLP Classifier
Combination 48 - XG Boost and Gradient Boosting Classifier
Combination 49 - XG Boost and XG Boost
```

(b) Combinations of Machine Learning Models involved in Boosting Approach with Dynamic Recursive Feature Elimination using analytical approach

Figure 7.3: Boosting approach evaluation on Synthetic Dataset

## 7.1 CROSS-DATASET METHODOLOGY EVALUATION

This chapter presents the outcomes of applying machine learning methodologies developed for each dataset (Synthetic, Turkiye, and Poland) onto the other datasets. The objective is to evaluate the generalizability, consistency, and adaptability of each methodology across different educational data sources. Performance metrics such as accuracy (before and after Dynamic Recursive Feature Elimination) are used to measure the effectiveness of each model combination.

Table 7.4: Accuracy Results on Turkiye Dataset using Synthetic Dataset Methodology

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 0 | Decision Tree | 100.00 | 100.00 | Decision Tree | 100.00 | 100.00 |
| 1 | Decision Tree | 100.00 | 100.00 | Logistic Regression | 100.00 | 100.00 |
| 2 | Decision Tree | 100.00 | 100.00 | Support Vector Machines | 100.00 | 100.00 |
| 3 | Decision Tree | 100.00 | 100.00 | Adaboost Classifier | 100.00 | 100.00 |
| 4 | Decision Tree | 100.00 | 100.00 | MLP Classifier | 100.00 | 100.00 |
| 5 | Decision Tree | 100.00 | 100.00 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 6 | Decision Tree | 100.00 | 100.00 | XG Boost | 100.00 | 100.00 |
| 7 | Logistic Regression | 94.46 | 94.46 | Decision Tree | 100.00 | 94.46 |
| 8 | Logistic Regression | 94.46 | 94.46 | Logistic Regression | 94.46 | 94.46 |
| 9 | Logistic Regression | 94.46 | 94.46 | Support Vector Machines | 94.46 | 94.46 |
| 10 | Logistic Regression | 94.46 | 94.46 | Adaboost Classifier | 100.00 | 94.46 |
| 11 | Logistic Regression | 94.46 | 94.46 | MLP Classifier | 94.46 | 94.46 |
| 12 | Logistic Regression | 94.46 | 94.46 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 13 | Logistic Regression | 94.46 | 94.46 | XG Boost | 100.00 | 100.00 |
| 14 | Support Vector Machines | 94.25 | 94.25 | Decision Tree | 100.00 | 94.25 |
| 15 | Support Vector Machines | 94.25 | 94.25 | Logistic Regression | 94.25 | 94.25 |

Continued on next page

Table 7.4 – continued from previous page

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 16 | Support Vector Machines | 94.25 | 94.25 | Support Vector Machines | 94.25 | 94.25 |
| 17 | Support Vector Machines | 94.25 | 94.25 | Adaboost Classifier | 100.00 | 94.25 |
| 18 | Support Vector Machines | 94.25 | 94.25 | MLP Classifier | 94.25 | 94.25 |
| 19 | Support Vector Machines | 94.25 | 94.25 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 20 | Support Vector Machines | 94.25 | 94.25 | XG Boost | 100.00 | 100.00 |
| 21 | Adaboost Classifier | 100.00 | 100.00 | Decision Tree | 100.00 | 100.00 |
| 22 | Adaboost Classifier | 100.00 | 100.00 | Logistic Regression | 100.00 | 100.00 |
| 23 | Adaboost Classifier | 100.00 | 100.00 | Support Vector Machines | 100.00 | 100.00 |
| 24 | Adaboost Classifier | 100.00 | 100.00 | Adaboost Classifier | 100.00 | 100.00 |
| 25 | Adaboost Classifier | 100.00 | 100.00 | MLP Classifier | 100.00 | 100.00 |
| 26 | Adaboost Classifier | 100.00 | 100.00 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 27 | Adaboost Classifier | 100.00 | 100.00 | XG Boost | 100.00 | 100.00 |
| 28 | MLP Classifier | 95.30 | 95.30 | Decision Tree | 100.00 | 100.00 |
| 29 | MLP Classifier | 95.30 | 95.30 | Logistic Regression | 96.00 | 96.00 |
| 30 | MLP Classifier | 95.30 | 95.30 | Support Vector Machines | 94.95 | 94.88 |
| 31 | MLP Classifier | 95.30 | 95.30 | Adaboost Classifier | 100.00 | 100.00 |
| 32 | MLP Classifier | 95.30 | 95.30 | MLP Classifier | 95.30 | 95.30 |
| 33 | MLP Classifier | 95.30 | 95.30 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 34 | MLP Classifier | 95.30 | 95.30 | XG Boost | 100.00 | 100.00 |
| 35 | Gradient Boosting Classifier | 100.00 | 100.00 | Decision Tree | 100.00 | 100.00 |
| 36 | Gradient Boosting Classifier | 100.00 | 100.00 | Logistic Regression | 100.00 | 100.00 |
| 37 | Gradient Boosting Classifier | 100.00 | 100.00 | Support Vector Machines | 100.00 | 100.00 |
| 38 | Gradient Boosting Classifier | 100.00 | 100.00 | Adaboost Classifier | 100.00 | 100.00 |
| 39 | Gradient Boosting Classifier | 100.00 | 100.00 | MLP Classifier | 100.00 | 100.00 |

Continued on next page

Table 7.4 – continued from previous page

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 40 | Gradient Boosting Classifier | 100.00 | 100.00 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 41 | Gradient Boosting Classifier | 100.00 | 100.00 | XG Boost | 100.00 | 100.00 |
| 42 | XG Boost | 100.00 | 100.00 | Decision Tree | 100.00 | 100.00 |
| 43 | XG Boost | 100.00 | 100.00 | Logistic Regression | 100.00 | 100.00 |
| 44 | XG Boost | 100.00 | 100.00 | Support Vector Machines | 100.00 | 100.00 |
| 45 | XG Boost | 100.00 | 100.00 | Adaboost Classifier | 100.00 | 100.00 |
| 46 | XG Boost | 100.00 | 100.00 | MLP Classifier | 100.00 | 100.00 |
| 47 | XG Boost | 100.00 | 100.00 | Gradient Boosting Classifier | 100.00 | 100.00 |
| 48 | XG Boost | 100.00 | 100.00 | XG Boost | 100.00 | 100.00 |

Table 7.5: Accuracy Results on Poland Dataset using Methodologies Derived from Synthetic Dataset

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 0 | Decision Tree | 99.29 | 99.21 | Decision Tree | 99.29 | 99.29 |
| 1 | Decision Tree | 99.29 | 99.21 | Logistic Regression | 99.29 | 99.29 |
| 2 | Decision Tree | 99.29 | 99.21 | Support Vector Machines | 99.29 | 99.29 |
| 3 | Decision Tree | 99.29 | 99.21 | Adaboost Classifier | 99.29 | 99.29 |
| 4 | Decision Tree | 99.29 | 99.21 | MLP Classifier | 99.33 | – |
| 5 | Decision Tree | 99.29 | 99.21 | Gradient Boosting Classifier | 99.29 | 99.29 |
| 6 | Decision Tree | 99.29 | 99.21 | XG Boost | 99.29 | 99.29 |
| 7 | Logistic Regression | 62.87 | 62.74 | Decision Tree | 99.29 | 99.21 |
| 8 | Logistic Regression | 62.87 | 62.74 | Logistic Regression | 62.12 | 61.29 |
| 9 | Logistic Regression | 62.87 | 62.74 | Support Vector Machines | 62.87 | 62.87 |
| 10 | Logistic Regression | 62.87 | 62.74 | Adaboost Classifier | 67.73 | 67.73 |
| 11 | Logistic Regression | 62.87 | 62.74 | MLP Classifier | 82.04 | – |
| 12 | Logistic Regression | 62.87 | 62.74 | Gradient Boosting Classifier | 77.34 | 77.34 |
| 13 | Logistic Regression | 62.87 | 62.74 | XG Boost | 98.34 | 98.71 |

Continued on next page

Table 7.5 – continued from previous page

| S.No | First Machine Learning Model | Acc. 1 (%) | Acc. 1 + RFE (%) | Second Machine Learning Model | Acc. 2 (%) | Acc. 2 + RFE (%) |
|---|---|---|---|---|---|---|
| 14 | Support Vector Machines | 63.58 | 63.28 | Decision Tree | 99.50 | 99.21 |
| 15 | Support Vector Machines | 63.58 | 63.28 | Logistic Regression | 62.33 | 62.54 |
| 16 | Support Vector Machines | 63.58 | 63.28 | Support Vector Machines | 63.58 | 63.58 |
| 17 | Support Vector Machines | 63.58 | 63.28 | Adaboost Classifier | 66.57 | 66.57 |
| 18 | Support Vector Machines | 63.58 | 63.28 | MLP Classifier | 84.45 | – |
| 19 | Support Vector Machines | 63.58 | 63.28 | Gradient Boosting Classifier | 77.55 | 77.75 |
| 20 | Support Vector Machines | 63.58 | 63.28 | XG Boost | 97.71 | 97.75 |
| 21 | Adaboost Classifier | 67.73 | 67.73 | Decision Tree | 99.29 | 99.25 |
| 22 | Adaboost Classifier | 67.73 | 67.73 | Logistic Regression | 67.23 | 67.32 |
| 23 | Adaboost Classifier | 67.73 | 67.73 | Support Vector Machines | 67.73 | 67.73 |
| 24 | Adaboost Classifier | 67.73 | 67.73 | Adaboost Classifier | 67.86 | 67.86 |
| 25 | Adaboost Classifier | 67.73 | 67.73 | MLP Classifier | 82.91 | – |
| 26 | Adaboost Classifier | 67.73 | 67.73 | Gradient Boosting Classifier | 72.97 | 73.26 |
| 27 | Adaboost Classifier | 67.73 | 67.73 | XG Boost | 97.96 | 97.96 |
| 28 | MLP Classifier | 83.37 | – | Decision Tree | 99.33 | 99.29 |
| 29 | MLP Classifier | 83.37 | – | Logistic Regression | 83.37 | 83.37 |
| 30 | MLP Classifier | 83.37 | – | Support Vector Machines | 83.37 | 83.37 |
| 31 | MLP Classifier | 83.37 | – | Adaboost Classifier | 83.37 | 83.37 |
| 32 | MLP Classifier | 83.37 | – | MLP Classifier | 85.20 | – |
| 33 | MLP Classifier | 83.37 | – | Gradient Boosting Classifier | 83.78 | 83.74 |
| 34 | MLP Classifier | 83.37 | – | XG Boost | 98.67 | 98.05 |
| 35 | Gradient Boosting Classifier | 78.00 | 77.34 | Decision Tree | 99.29 | 98.50 |
| 36 | Gradient Boosting Classifier | 78.00 | 77.34 | Logistic Regression | 78.00 | 78.00 |
| 37 | Gradient Boosting Classifier | 78.00 | 77.34 | Support Vector Machines | 78.00 | 78.00 |

Continued on next page

Table 7.5 – continued from previous page

| **S.No** | **First Machine Learning Model** | **Acc. 1 (%)** | **Acc. 1 + RFE (%)** | **Second Machine Learning Model** | **Acc. 2 (%)** | **Acc. 2 + RFE (%)** |
|---|---|---|---|---|---|---|
| 38 | Gradient Boosting Classifier | 78.00 | 77.34 | Adaboost Classifier | 78.00 | 78.00 |
| 39 | Gradient Boosting Classifier | 78.00 | 77.34 | MLP Classifier | 86.49 | – |
| 40 | Gradient Boosting Classifier | 78.00 | 77.34 | Gradient Boosting Classifier | 78.71 | 78.63 |
| 41 | Gradient Boosting Classifier | 78.00 | 77.34 | XG Boost | 97.30 | 96.59 |
| 42 | XG Boost | 98.21 | 98.71 | Decision Tree | 99.29 | 99.29 |
| 43 | XG Boost | 98.21 | 98.71 | Logistic Regression | 98.21 | 98.21 |
| 44 | XG Boost | 98.21 | 98.71 | Support Vector Machines | 98.21 | 98.21 |
| 45 | XG Boost | 98.21 | 98.71 | Adaboost Classifier | 98.21 | 98.21 |
| 46 | XG Boost | 98.21 | 98.71 | MLP Classifier | 98.59 | – |
| 47 | XG Boost | 98.21 | 98.71 | Gradient Boosting Classifier | 98.21 | 98.21 |
| 48 | XG Boost | 98.21 | 98.71 | XG Boost | 99.04 | 98.67 |

## Table 7.6: Accuracy Results on Synthetic Dataset using Methodologies Derived from Turkiye Dataset

| **Classifier** | **Satisfaction Index** | | **K-means Clustering** | | **Agglomerative Clustering** | |
|---|---|---|---|---|---|---|
| | Individual Model | RFE + K-Fold + Model | Individual Model | RFE + K-Fold + Model | Individual Model | RFE + K-Fold + Model |
| Logistic Regression | 81.7 | 81.5 | 99.4 | 98.5 | 95.4 | 98.5 |
| Decision Tree Classifier | 77.4 | 80.6 | 94.8 | 95.1 | 92.5 | 95.1 |
| Bagging Classifier | 80.7 | 84.7 | 96.8 | 96.5 | 95.1 | 96.5 |
| Random Forest Classifier | 77.4 | **86.9** | 94.8 | 97.7 | 92.5 | 97.7 |
| Adaboost Classifier | **84.4** | 83.0 | 97.4 | 97.8 | 94.8 | 97.8 |
| XGBoost Classifier | 83.6 | 86.1 | 98.1 | 97.6 | 96.4 | 97.6 |
| SVM | 82.9 | 82.2 | 99.0 | 98.4 | 95.4 | 98.4 |
| Voting Classifier (Soft Voting Score) | 83.7 | - | 98.8 | - | 96.1 | - |
| Voting Classifier (Hard Voting Score) | 82.6 | - | 98.9 | - | **96.5** | - |

Table 7.7: Accuracy Results on Synthetic Dataset using Methodologies Derived from Poland University Dataset

| **Classifier** | **Model Accuracy (%)** |
|---|---|
| Logistic Regression | 50.63 |
| Decision Tree Classifier | 50.03 |
| Random Forest Classifier | 48.93 |
| Bagging Classifier | 49.43 |
| Adaboost Classifier | 49.40 |
| XGBoost Classifier | 48.67 |
| SVM | 50.97 |
| Voting Classifier (Soft Voting) | 47.93 |
| Voting Classifier (Hard Voting) | 48.53 |
| Stacking Classifier | 51.10 |

Table 7.8: Accuracy Results on Poland Dataset using Methodologies Derived from Türkiye Dataset

| **Classifier** | **Satisfaction Index** | | **K-means Clustering** | | **Agglomerative Clustering** | |
|---|---|---|---|---|---|---|
| | **Individual Model** | **RFE + CV** | **Individual Model** | **RFE + CV** | **Individual Model** | **RFE + CV** |
| Logistic Regression | 71.31 | 71.39 | 98.71 | 98.33 | 94.89 | 94.12 |
| Decision Tree Classifier | 97.30 | 97.98 | 100.00 | 100.00 | 100.00 | 100.00 |
| Bagging Classifier | 98.00 | 98.48 | 100.00 | 100.00 | 99.83 | 100.00 |
| Random Forest Classifier | 98.96 | 99.14 | 100.00 | 100.00 | 100.00 | 100.00 |
| Adaboost Classifier | 76.55 | 74.61 | 98.46 | 98.85 | 94.18 | 94.93 |
| XGBoost Classifier | 98.79 | 98.43 | 100.00 | 100.00 | 99.83 | 100.00 |
| SVM | 54.93 | 65.56 | 98.09 | 97.98 | 96.17 | 96.31 |
| Voting Classifier (Soft) | 98.25 | 98.59 | 100.00 | 100.00 | 99.83 | 99.98 |

Table 7.9: Accuracy Results on Turkiye Dataset using Methodologies Derived from Poland Dataset

| **Classifier** | **Model Accuracy (%)** |
|---|---|
| Decision Tree Classifier | 100.00 |
| XGBoost Classifier | 100.00 |
| Bagging Classifier | 100.00 |
| Random Forest Classifier | 100.00 |
| Adaboost Classifier | 100.00 |
| Stacking Classifier | 100.00 |
| Voting Classifier | 100.00 |
| Logistic Regression | 98.92 |
| SVM | 98.38 |

## 7.2 WEB APPLICATION: KNOWMYPROF

As a value-added component to this project, a web-based interface was developed to present instructor information in an intuitive and interactive format. The platform allows users to view each professor's profile, including their photograph, average rating, and key achievements such as research contributions, years of experience, and academic involvement. The data displayed is sourced directly from the synthetic dataset used in the project and stored in a structured format within MongoDB. By integrating technologies such as HTML, CSS, JavaScript, Node.js, and Express, the website enhances user engagement and serves as a practical demonstration of how data-driven academic insights can be made accessible to a broader audience. This implementation reinforces the project's focus on real-world relevance and supports informed decision-making in educational environments.

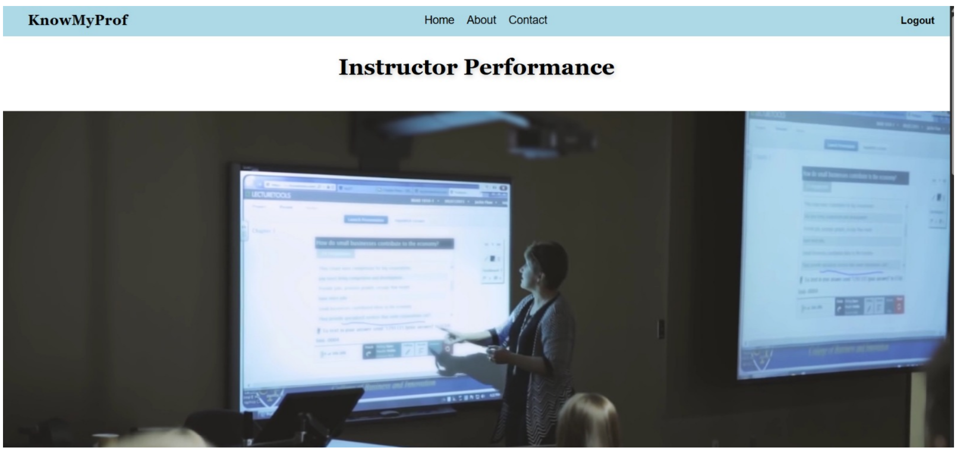



Figure 7.4: KnowMyProf Homepage

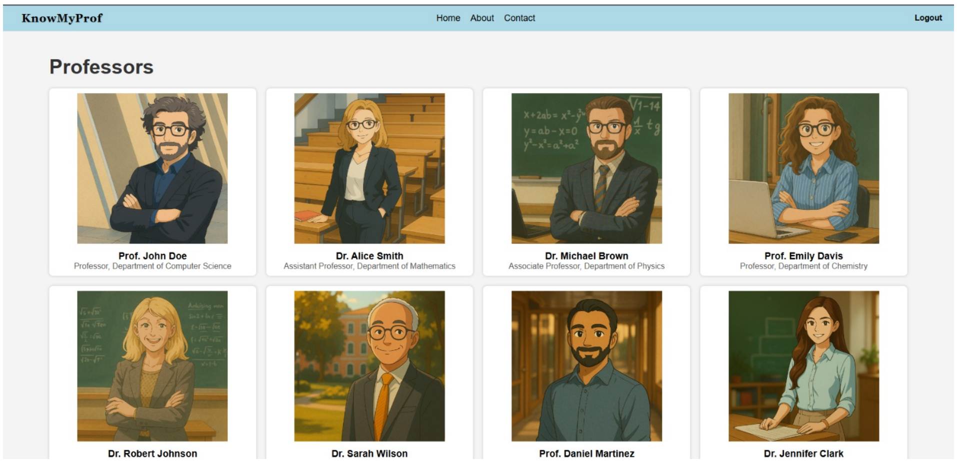



Figure 7.5: Professors Listing Page

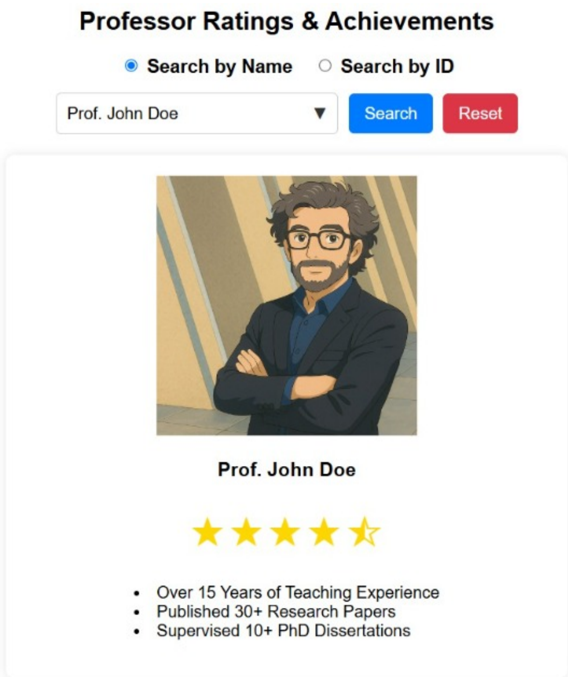



Figure 7.6: Professor Rating View

# Chapter 8

# CONCLUSION

## CONCLUDING REMARKS

Providing quality education is crucial, and this study contributes to that goal by leveraging data collected from students about their courses and instructors to predict instructor performance. Since instructors play a pivotal role in shaping learning outcomes, understanding and improving their effectiveness has a direct impact on education quality.

Unlike many studies that focus solely on student performance metrics such as CGPA or internal assessments, this work takes a different approach—shifting the focus toward the educators themselves. It evaluates both instructor-specific factors (like experience and preparedness) and course-specific factors (like difficulty and structure), recognizing that these dimensions are equally important in understanding overall teaching effectiveness.

In this work, we examined three datasets: the Poland University Dataset, the Turkiye Student Evaluation Dataset, and a synthetically generated dataset. Each dataset brings unique attributes and structures, providing a diverse base for experimentation. Using various machine learning models and feature selection techniques, we analyzed and predicted instructor performance with notable accuracies—78% for the Poland dataset, 99% for the Turkiye dataset, and 90.15% for the synthetic dataset.

To further strengthen our contribution, we applied every machine learning methodology to each of the three datasets individually—producing a comprehensive 3×3 matrix of model-dataset results. From this, the Decision Tree + Support Vector Machine combination on the synthetic dataset yielded the highest accuracy. Using this outcome, we developed a web-based application where users can input an instructor's name or ID and receive performance insights—making our work applicable beyond the academic setting.

Overall, this project demonstrates how data mining and machine learning can be practically applied to improve teaching quality and support data-informed decision-making in educational institutions.

## FUTURE WORK

Additional factors like students' interest towards a course, improvement over the period of the course, student attendance, and instructor-specific factors such as non-verbal cues, communication style, and body language can be explored further to understand their influence on instructor performance. These human-centric and behavioral aspects are often harder to quantify but have a significant impact on student learning and engagement.

More exploration can also be carried out to improve the prediction accuracies beyond the currently stated values by experimenting with larger datasets, dynamic feature sets, and real-time feedback inputs.

It is certain that the quality and relevance of the questions in a dataset play a crucial role in model performance. Therefore, pursuing more informative, structured, and diverse datasets—possibly crowdsourced or collected across various institutions—can enable broader experimentation and more reliable conclusions. Collaborating with academic departments to access anonymized real-world feedback data could provide additional depth and authenticity to future datasets.

Apart from the models discussed in this report, there is potential to explore a wide range of advanced machine learning and deep learning algorithms such as LightGBM, CatBoost, Gradient Boosting Machines, and neural networks. Ensemble techniques like stacking, bagging, and boosting can be fine-tuned and compared under different scenarios. Automated machine learning (AutoML) platforms can also be tested to evaluate their efficiency in handling educational datasets.

Feature importance is another key direction to pursue. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) can be used to explain model predictions and help in identifying the most influential attributes. This not only improves model interpretability but also allows institutions to focus on actionable areas during instructor evaluations.

Currently, in the synthetic dataset model, we manually defined fixed parameters like the target variable, train-test split ratio, and feature weights. In the future, we aim to enhance this system by turning it into a modular and reusable framework that allows users to define these parameters dynamically. By creating a flexible user interface or configuration setup, the model can be applied to different datasets and user needs without manual code changes—making it more scalable and user-friendly.

Additionally, a key area of interest is model stability and failure analysis. We plan to investigate scenarios where models underperform—such as imbalanced classes, noisy data, or biased inputs—and design mechanisms to handle such challenges. Introducing techniques like cross-validation with multiple sampling strategies, or implementing robust regularization, can make the model more resilient to various data conditions.

Finally, we intend to expand our existing web application to support dataset uploads, model customization, and downloadable results. Incorporating visualization dashboards and exporting tools would help stakeholders analyze and interpret model outcomes more effectively, bridging the gap between technical output and user comprehension.

# REFERENCES


[1] **Mardikyan, S., & Badur, B.** (2011). Analyzing teaching performance of instructors using data mining techniques. *Informatics in Education*, 10(2), 245-257.

[2] **Ahmed, A. M., Rizaner, A., & Ulusoy, A. H.** (2016). Using data mining to predict instructor performance. *Procedia Computer Science*, 102, 137-142.

[3] **Ahuja, R., & Sharma, S. C.** (2021). Exploiting Machine Learning and Feature Selection Algorithms to Predict Instructor Performance in Higher Education. *Journal of Information Science & Engineering*, 37(5).

[4] **Asanbe, M. O., Osofisan, A. O., & William, W. F.** (2016). Teachers' performance evaluation in higher educational institution using data mining technique. *International Journals of Applied Information System (IJAIS)*, 10, 10-15.

[5] **Benosa, B., & Caguiat, M. R. R.** (2021). Prescriptive Faculty Performance Analysis: A Case at the Onset of Covid-19 Pandemic. In *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)* (pp. 839-844). IEEE.

[6] **Ahmed, A. M., Rizaner, A., & Ulusoy, A. H.** (2016). Using data mining to predict instructor performance. *Procedia Computer Science*, 102, 137-142.

[7] **Agaoglu, M.** (2016). Predicting instructor performance using data mining techniques in higher education. *IEEE Access*, 4, 2379-2387.

[8] **McConnell, J. R., Bruster, B. G., & Smith, V. B.** (2019). Predicting teacher effectiveness based on various preservice factors: Implications for higher education and the evaluation of teacher preparation. *International Journal of Educational Reform*, 28(1), 63-78.

[9] **Abunasser, B. S., AL-Hiealy, M. R. J., Barhoom, A. M., Almasri, A. R., & Abu-Naser, S. S.** (2022). Prediction of Instructor Performance using Machine and Deep Learning Techniques. *International Journal of Advanced Computer Science and Applications*, 13(7).

[10] **Shi, F., Wang, L., Liu, X., & Chiu, M. H.** (2021). Development and validation of an observation protocol for measuring science teachers' modeling-based teaching performance. *Journal of Research in Science Teaching*, 58(9), 1359-1388.

[11] **Hranchuk, K. S., & Williams, M. J.** (2021). Addressing the feasibility of the teacher performance rate and accuracy scale as a treatment integrity tool. *Behavioral Interventions*, 36(2), 355-368.

[12] **Ke, L., & Schwarz, C. V.** (2021). Supporting students' meaningful engagement in scientific modeling through epistemological messages: A case study of contrasting teaching approaches. *Journal of Research in Science Teaching*, 58(3), 335-365.

[13] **Almasri, A.** (2017). Experiences in Mining Educational Data to Analyze Teacher's Performance: A Case Study with High Educational Teachers.

[14] **Odegua, R.** (2019). An empirical study of ensemble techniques (bagging, boosting and stacking). *In Proc. Conf.: Deep Learn. IndabaXAt*.

[15] **Saeys, Y., Inza, I., & Larranaga, P.** (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19), 2507-2517.

[16] **Gunduz, G., & Fokoue, E.** (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.