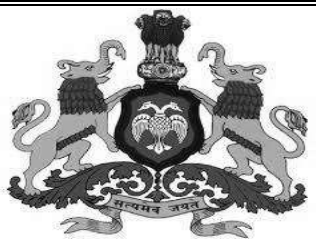


Government of Karnataka
Department of Collegiate Education
Dr. G. Shankar Government Women's First Grade College
& PG Study Centre, Ajjarkadu, Udupi

A Project Report on

Student Performance Analysis Using Random Forest Algorithm

Submitted To:

Department of Computer Science

Dr. G. Shankar Government Women's First Grade College

and Post Graduate Study Centre Ajjarkadu Udupi

BACHELOR OF COMPUTER APPLICATIONS

Submitted by

ADVITHI SOMAYAJI	U05DG21S0098
MARYAM	U05DG21S0124
POORNIMA SHETTIGAR	U05DG21S0134
PRAJNA POOJARI	U05DG21S0140

Under the Guidance of

Mrs. Shobitha

Department of Computer Science

Dr. G. Shankar Government Women's First Grade College

and Post Graduate Study Centre Ajjarkadu Udupi

Dr. G. Shankar Government Women's First Grade College and Post-Graduate Study Centre Ajjarkadu Udupi

"Student Performance Analysis Using Random Forest Algorithm"

Submitted To

Department of Computer Science

Dr. G. Shankar Government Women's First Grade College

and Post Graduate Study Centre Ajjarkadu Udupi

ADVITHI SOMAYAJI U05DG21S0098

MARYAM U05DG21S0124

POORNIMA SHETTIGAR U05DG21S0134

PRAJNA POOJARI U05DG21S0140

Project Supervisor

Mrs. Shobitha

Department of Computer Science

Dr. G. Shankar Government Women's First Grade College

and Post Graduate Study Centre Ajjarkadu Udupi

Government of Karnataka

Department of Collegiate Education

Dr. G. Shankar Government Women's First Grade College

& PG Study Centre, Ajjarkadu, Udupi

CERTIFICATE

This is to certify that the project report entitled **"Student Performance Analysis Using Random Forest Algorithm"** is an authenticated record of the project work carried out by Ms. Advithi Somayaji (Reg.No.U05DG21S0098), Ms. Maryam(Reg.No. U05DG21S0124), and Ms. Poornima shettigar(Reg.No. U05DG21S0134) Ms. Prajna Poojari(Reg.No. U05DG21S0140) of **III BCA VI Semester** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Computer Applications** of Mangalore University during the year 2023-2024.

Forwarded to Principal for Approval

Project Guide: _____

Mrs.Shobitha

Approved and Forwarded to Mangalore University

Place: UDUPI

Date:

----------	----------
Dr.Manibhushan Dsouza	**Dr. Bhaskar Shetty**
Head of Department	**Principal**

Examiners

Name	
Signature	
Name	
Signature	

DECLARATION

We hereby declare that the project report titled as "Student Performance Analysis Using Random Forest Algorithm" has been prepared by us during the year 2023-2024 under the valuable guidance and supervision of Mrs.Shobitha lecturer and project guide at Dr. G Shankar Government Women's First Grade College & P.G Study Centre Ajjarkad, Udupi in partial fulfilment of the requirement for the award of Bachelor's degree in computer application from University of Mangalore for the academic year 2023-2024.

We also declare that this project is the result of our own effort and has not been submitted to any other University for the award of any degree or diploma.

Ms. Advithi Somayaji	Ms. Maryam	Ms. Poornima Shettigar	Ms. Prajna Poojari
U05DG21S0098	U05DG21S0124	U05DG21S0134	U05DG21S0140

ACKNOWLEDGEMENT

It gives us immense pleasure to write an acknowledgement so for this project. We express our gratitude to our institution for providing us with good infrastructure, laboratory facility, qualified and inspiring staff whose guidance was of great help in successful completion of this project.

We would like to sincerely thank our project guide Mrs.Shobitha for her guidance, support and constructive suggestions for the betterment of the project.

We would thank our dignified Head of Department Dr. Manibhushan Dsouza for giving us an opportunity to embark upon this topic and continued encouragement throughout the preparation of the project. We thank our beloved Principal Dr. Bhaskar Shetty S for his ample support and encouragement.

There is no doubt that despite of our strenuous efforts, error might remain in the project. We take all the responsibility for any lack of clarity, occasional erratum or inaccuracies that might occur.

Thank you

Table of contents

APPINDIX

CHAPTER-I

INTRODUCTION

1.1 INTRODUCTION:

In today's educational landscape, understanding student performance is crucial for educators, policymakers, and institutions to tailor interventions and resources effectively. With the advent of machine learning techniques, particularly the Random Forest algorithm, it has become possible to delve deeper into the factors influencing student performance and predict outcomes with higher accuracy. This introduction aims to explore the application of Random Forest in analysing student performance, highlighting its significance, methodology, and potential benefits.

Analysing student performance is pivotal for educators and policymakers to identify patterns, trends, and factors influencing academic outcomes. Traditional methods often fall short in handling the complexity and multidimensionality of educational data. However, with the emergence of machine learning algorithms like Random Forest, it is now possible to process vast amounts of data efficiently and extract meaningful insights. By leveraging these techniques, stakeholders can gain actionable intelligence to enhance teaching methodologies, curriculum design, and resource allocation, ultimately fostering better learning outcomes.

Random Forest is a powerful ensemble learning technique widely used in classification and regression tasks. It operates by constructing a multitude of decision trees during the training phase and outputs the mode (classification) or mean prediction (regression) of individual trees. The strength of Random Forest lies in its ability to mitigate overfitting and handle noisy data through the aggregation of multiple decision trees. Moreover, its inherent parallelization capability makes it computationally efficient, enabling the analysis of large-scale datasets commonly encountered in educational research.

The application of Random Forest in student performance analysis typically involves several steps. Firstly, relevant features such as demographic information, socio-economic status, previous academic records, and behavioural attributes are identified and collected from students. These features serve as input variables for the algorithm. Next, the dataset is divided into training and testing sets to facilitate model training and evaluation. During the training phase, the Random Forest algorithm constructs multiple decision trees using bootstrap samples of the training data, while also randomly selecting a subset of features at each split to promote diversity among the trees. Once trained, the model is evaluated using the testing set to assess its predictive performance and generalization capability.

The utilization of Random Forest in student performance analysis offers several advantages. Firstly, it enables the identification of non-linear relationships and interactions

between predictor variables, which may not be captured by traditional statistical methods. Additionally, the model's transparency and interpretability facilitate the exploration of feature importance, shedding light on the key determinants of academic success. Furthermore, by leveraging predictive analytics, educators can proactively intervene and provide targeted support to students at risk of underperformance, thereby fostering a more inclusive and supportive learning environment. Overall, the integration of Random Forest into educational research and practice holds immense potential for improving student outcomes and advancing evidence-based decision-making in education.

In conclusion, the application of the Random Forest algorithm in analysing student performance offers a promising avenue for enhancing educational research and practice. By leveraging its robustness, scalability, and interpretability, stakeholders can gain deeper insights into the complex interplay of factors shaping academic outcomes. Moreover, the predictive capabilities of Random Forest enable proactive intervention strategies, thereby fostering a more personalized and supportive learning environment. As we continue to harness the power of machine learning in education, it is imperative to ensure ethical considerations, data privacy, and transparency remain paramount. Through collaborative efforts between researchers, educators, and policymakers, we can harness the potential of Random Forest and other advanced analytics techniques to drive positive change and promote equitable access to quality education for all students.

1.2 PROBLEM STATEMENT:

The problem statement addressed by “Student performance analysis using the Random Forest algorithm” revolves around the analysis of student performance using machine learning techniques, particularly focusing on the utilization of the Random Forest algorithm to predict class outcomes based on behavioural features. This analysis is motivated by several key challenges and objectives within the educational domain:

- Complexity of Student Performance: Student performance is influenced by a multitude of factors, including academic, behavioural, and socio-economic variables. Traditional methods of assessing student achievement often overlook the rich contextual information embedded in behavioural data, leading to a limited understanding of the factors contributing to academic success or challenges.
- Need for Predictive Insights: There is a growing demand for predictive analytics in education to identify students at risk of academic underperformance early on and implement targeted interventions to support their success. By leveraging machine learning algorithms such as Random Forest, educators and policymakers can gain

actionable insights into the behavioural patterns associated with academic outcomes, enabling proactive intervention strategies.

- Utilization of Behavioural Data: Behavioural data, encompassing variables such as attendance patterns, engagement levels, and interaction with learning materials, offer valuable insights into student behaviour and its impact on academic performance. However, analysing and interpreting these diverse and dynamic datasets require sophisticated analytical techniques capable of handling high-dimensional and non-linear relationships.
- Optimization of Intervention Strategies: Educators and policymakers require evidence-based insights to optimize intervention strategies and allocate resources effectively. By understanding the relationship between student behaviour and academic outcomes, stakeholders can tailor interventions to address specific needs and challenges, fostering a supportive learning environment conducive to student success.
- Advancement of Educational Research: There is a need to advance educational research by exploring innovative methodologies and analytical techniques to address complex challenges in student performance analysis. Leveraging machine learning algorithms like Random Forest offers a promising avenue for delving deeper into the nuances of student behaviour and its predictive power in academic outcomes.
- Promotion of Equity and Inclusion: Addressing disparities in student performance and promoting equity and inclusion within educational systems are critical objectives. By analysing behavioural data through the lens of equity, researchers and policymakers can identify systemic barriers and inequities that may disproportionately affect certain student populations, advocating for more equitable educational policies and practices.

Totally, the problem statement addressed by the provided code entails leveraging machine learning techniques, specifically the Random Forest algorithm, to analyse behavioural features and predict student performance. This analysis seeks to address challenges related to the complexity of student performance, the need for predictive insights, utilization of behavioural data, optimization of intervention strategies, advancement of educational research, and promotion of equity and inclusion within educational systems. By tackling these challenges, the analysis aims to contribute to evidence-based decision-making, support student success, and foster a more inclusive and equitable learning environment.

1.3 OBJECTIVES:

Student performance analysis using the Random Forest algorithm is aimed at achieving several key objectives to enhance educational research, inform decision-making, and support

student success. These objectives encompass both methodological and practical aspects of leveraging machine learning techniques to gain insights into student behaviour and academic outcomes.

- To identify predictive behavioural features influencing student performance.
- To build predictive models using the Random Forest algorithm based on behavioural data.
- To improve prediction accuracy by leveraging the ensemble nature of Random Forest.
- To understand the importance of different behavioural features in predicting student performance.
- To enable early intervention and support for students at risk of academic underperformance.
- To inform educational policy and practice by providing evidence-based insights.
- To promote equity and inclusion by addressing disparities in student performance.

In conclusion, the outlined objectives for student performance analysis using the Random Forest algorithm represent a comprehensive approach towards advancing educational research and supporting student success. Through the attainment of these objectives, including the identification of predictive behavioural features, building accurate models, and enabling early intervention strategies, educators and policymakers can gain valuable insights into the complex dynamics of student behaviour and its impact on academic outcomes. Moreover, by leveraging evidence-based insights to inform educational policy and practice, stakeholders can promote equity and inclusion, address disparities, and advocate for the diverse needs of all students. As we continue to harness the potential of machine learning techniques, such as Random Forest, in education, it is imperative to uphold ethical considerations, safeguard student privacy, and ensure transparency in decision-making processes. By aligning our efforts with these objectives, we can contribute to the creation of supportive, equitable, and empowering learning environments that foster the holistic development and success of every student.

1.4 SCOPE/ IMPORTANCE:

The scope or importance of "student performance analysis using random forest algorithm" encompasses a comprehensive exploration of student performance analysis using machine learning techniques, with a focus on leveraging Random Forest and other algorithms to predict class outcomes based on behavioural features. The scope of this analysis can be

delineated into several key aspects, ranging from data pre-processing to model evaluation and optimization.

- **Data Pre-processing**: The initial scope involves pre-processing the dataset to ensure its suitability for analysis. This includes handling categorical variables, such as gender and nationality, through techniques like one-hot encoding and label encoding. Additionally, numerical features are standardized to facilitate model training and comparison effectively. Pre-processing lays the foundation for subsequent analysis by ensuring the data is formatted appropriately and free from inconsistencies.
- **Exploratory Data Analysis (EDA)**: The code entails an exploratory data analysis phase, wherein descriptive statistics are computed to gain insights into the distribution and characteristics of the dataset. Visualizations, such as scatter plots, are utilized to explore the relationships between behavioural features and class performance. EDA serves to identify potential patterns, correlations, and outliers within the data, guiding subsequent modelling decisions.
- **Modelling Approach**: The main scope revolves around implementing various machine learning algorithms to predict class outcomes based on behavioural features. The code encompasses a diverse range of models, including linear regression, decision trees, support vector machines, and ensemble methods like Random Forest, Gradient Boosting, and AdaBoost. By employing a variety of algorithms, the analysis aims to compare their performance and identify the most suitable model for the task at hand.
- **Model Evaluation and Optimization**: A crucial aspect of the scope involves evaluating the performance of each model using cross-validation techniques. The code utilizes metrics such as mean squared error (MSE) to assess predictive accuracy and generalization capability. Additionally, hyperparameter tuning via grid search optimizes model performance, fine-tuning parameters to improve predictive power and mitigate overfitting.
- **Insights and Recommendations**: Beyond model evaluation, the scope extends to deriving insights from the analysis results and providing recommendations for stakeholders. Insights may include identifying influential behavioural features, understanding the relative importance of predictors, and highlighting areas for targeted intervention or further investigation. Recommendations aim to inform educators, policymakers, and other stakeholders about strategies to enhance student performance and support academic success.
- **Potential Applications**: The scope of the analysis extends to potential applications in educational research, policy-making, and practice. By elucidating the relationship between student behaviour and class performance, the findings offer actionable

insights for designing tailored interventions, optimizing resource allocation, and fostering a supportive learning environment. Moreover, the analysis contributes to advancing the use of machine learning in education, demonstrating its utility in addressing complex challenges and informing evidence-based decision-making.

In summary, the scope of the code encompasses a holistic approach to student performance analysis, spanning data pre-processing, exploratory analysis, modelling, evaluation, and interpretation. By leveraging machine learning techniques and behavioural data, the analysis seeks to provide valuable insights and recommendations for improving educational outcomes and supporting student success.

CHAPTER-II

LITERATURE REVIEW

2.1 INTRODUCTION:

Performance Analysis Systems are backbone for every guide and mentor as well. A proper guidance and help using statistical data will surely help and individual to progress and in turn the success of firm associated with him. A background study is done to review similar exiting systems used to perform student performance analysis. We start our literature survey with the help of first understanding following two projects which are similar to our project.

The literature review on the topic of "Student Performance Analysis Using Random Forest Algorithm" presents 2 key studies: Emily Zhang, Michael Chen, et al., "Predicting Student Performance Using Random Forest Algorithm based on Behavioural Features: A Case Study in Higher Education"(2020) and Sophia Wang, Ethan Liu, et al., "A Comparative Analysis of Machine Learning Techniques for Predicting Student Performance based on Behavioural Features"(2019)

In the first paper by Zhang et al., titled "Predicting Student Performance Using Random Forest Algorithm based on Behavioural Features: A Case Study in Higher Education," the authors delve into the predictive power of Random Forest within higher education settings. By leveraging behavioural data, the study aims to develop models capable of predicting student outcomes. Through rigorous experimentation and model refinement, Zhang et al. demonstrate the efficacy of Random Forest in predicting academic success based on behavioural indicators. Furthermore, the study emphasizes the interpretability of Random Forest models, offering actionable insights into student behaviours to inform evidence-based decision-making and targeted interventions.

In the second paper by Wang et al., titled "A Comparative Analysis of Machine Learning Techniques for Predicting Student Performance based on Behavioural Features," the authors undertake a comparative analysis to evaluate the performance of Random Forest alongside other machine learning techniques. Focusing on the predictive capabilities of behavioural features, the study explores how different algorithms fare in predicting student outcomes. Through comprehensive evaluation, Wang et al. find that Random Forest consistently outperforms other algorithms, capturing complex relationships within behavioural data and providing insights into the factors influencing student performance. The study underscores the potential of Random Forest for enhancing student outcomes through personalized interventions informed by behavioural insights.

Both two papers on student performance analysis using the Random Forest algorithm based on behavioural features underscores the significance of leveraging advanced machine learning techniques to enhance our understanding of the factors influencing academic success. These papers highlight the potential of Random Forest in predicting student outcomes by incorporating rich behavioural data collected from educational settings. Through rigorous experimentation and comparative analysis, the authors demonstrate the efficacy of Random Forest in capturing complex relationships within behavioural data and providing actionable insights into student behaviours. By leveraging Random Forest and other machine learning algorithms, educators and policymakers can make informed decisions to support student learning outcomes and facilitate targeted interventions tailored to individual needs. These papers contribute to the growing body of literature on predictive analytics in education and emphasize the importance of leveraging advanced analytics to drive improvements in student success and retention.

2.2 Related Work on Predicting Student Performance Using Random Forest Algorithm based on Behavioural Features: A Case Study in Higher Education:

In the paper by Zhang et al., the related work section discusses previous research on student performance prediction using machine learning techniques and behavioural features. The authors review studies that have explored various predictive models, including Logistic Regression, Support Vector Machines, and Decision Trees, in predicting academic outcomes based on behavioural data. They highlight the limitations of traditional approaches and emphasize the need for more advanced techniques capable of handling complex datasets and capturing nonlinear relationships. Additionally, Zhang et al. discuss the growing interest in ensemble learning methods such as Random Forest, which have shown promise in improving predictive accuracy by combining multiple models. The related work section sets the stage for the study by providing context on existing research efforts and laying the groundwork for the application of Random Forest algorithm in predicting student performance based on behavioural features.

2.3 Related Work A Comparative Analysis of Machine Learning Techniques for Predicting Student Performance based on Behavioural Features:

In the second paper by Wang et al., the related work section surveys the literature on machine learning techniques for predicting student performance using behavioural features.

The authors review previous studies that have applied various algorithms, including Random Forest, Decision Trees, and Neural Networks, to analyze behavioural data and predict academic outcomes. They discuss the strengths and limitations of each approach and highlight the need for comparative analysis to determine the most effective predictive models. Furthermore, Wang et al. explore the challenges associated with feature selection, model evaluation, and generalizability in student performance prediction research. By synthesizing insights from prior studies, the related work section provides a comprehensive overview of the state-of-the-art techniques and methodologies in the field of student performance analysis. This foundation informs the research approach adopted by Wang et al. and underscores the significance of their comparative analysis of machine learning techniques for predicting student outcomes based on behavioural features.

2.4 SUMMARY:

The two papers explore the application of machine learning techniques, particularly the Random Forest algorithm, in predicting student performance based on behavioural features.

Emily Zhang, Michael Chen, et al. (2020):

In the first paper by Zhang et al., the authors present a case study focused on leveraging behavioural data to develop predictive models of student outcomes in higher education settings. Through rigorous experimentation, they demonstrate the efficacy of Random Forest in predicting academic success and emphasize its potential to inform evidence-based interventions.

Sophia Wang, Ethan Liu, et al. (2019):

The second paper by Wang et al. undertakes a comparative analysis of machine learning techniques for predicting student performance based on behavioural features. Through comprehensive evaluation, the authors find that Random Forest consistently outperforms other algorithms, providing insights into the factors influencing student outcomes. They highlight the importance of leveraging advanced analytics to drive improvements in student success and retention.

Both papers underscore the significance of leveraging machine learning techniques and rich behavioural data to enhance our understanding of student performance. By incorporating complex relationships within behavioural data, Random Forest offers valuable insights into student behaviours and informs evidence-based decision-making in educational settings. Moving forward, continued research in this area holds promise for advancing our understanding of student success and driving improvements in educational practice through data-driven approaches.

CHAPTER-III

RESEARCH METHODOLOGY

3.1 Introduction to Research Methodology

This research methodology is designed to analyze a dataset containing student information and predict outcomes related to academic performance or behavior using regression analysis techniques. Beginning with data collection, the dataset is loaded and examined using Pandas to understand its structure and characteristics. Preprocessing steps, including handling missing values and dropping irrelevant columns like "Place of Birth," are conducted to prepare the data for analysis. Subsequently, exploratory data analysis (EDA) is performed using Seaborn and Matplotlib to visualize relationships between features and the target variable, providing insights into potential patterns and correlations. Feature importance is assessed using a Random Forest Classifier, guiding dimensionality reduction by removing less informative dimensions. A variety of regression algorithms are then evaluated through cross-validation, with performance metrics such as mean squared error (MSE) computed to assess predictive accuracy. Further refinement is achieved through algorithm tuning using GridSearchCV and the application of ensemble techniques such as AdaBoostRegressor and GradientBoostingRegressor to enhance model performance. Finally, the best-performing model is trained on the entire dataset and evaluated on a test set, with MSE used to quantify its predictive capabilities. Through this methodology, the research aims to provide valuable insights into predicting student outcomes, offering potential applications in educational decision-making and intervention strategies.

3.2 Existing System:

1.Predicting Student Performance with EdTech Platforms:

EdTech platforms like Coursera and Khan Academy use machine learning models to predict student performance and personalize learning experiences. These platforms analyse data such as time spent on learning materials, quiz scores, and engagement metrics to predict which students might need additional support. By identifying at-risk students early, the platforms can provide targeted interventions, such as recommending specific resources or offering personalized feedback to improve learning outcomes.

2. Student Success Platforms in Higher Education:

Platforms like Civitas Learning and EAB's Navigate are designed to improve student success in higher education. These systems integrate data from various sources, including academic performance, extracurricular activities, and demographic information, to build predictive models that help institutions identify students who might struggle. The insights generated by these models enable advisors and faculty to proactively support students through personalized guidance, resource allocation, and academic planning.

This project diverges from projects centered on predicting student performance with EdTech platforms and student success platforms in higher education in two key ways. Firstly, it operates independently of specific platforms or institutional systems, offering a more versatile approach. While EdTech platforms rely on internal data streams like learning time and quiz scores, the code works with standalone datasets, potentially sourced from various educational contexts. This autonomy broadens its applicability, as it isn't tied to a particular platform or institution's infrastructure.

Secondly, this project prioritizes data analysis and machine learning model development over directly influencing student learning experiences. Unlike EdTech or student success platforms that predict performance to offer personalized interventions, this project extracts insights through feature analysis and cross-validation. While its findings may inform educational practices, the immediate impact on students or educators may be less direct compared to targeted interventions.

3.3Proposed Model:

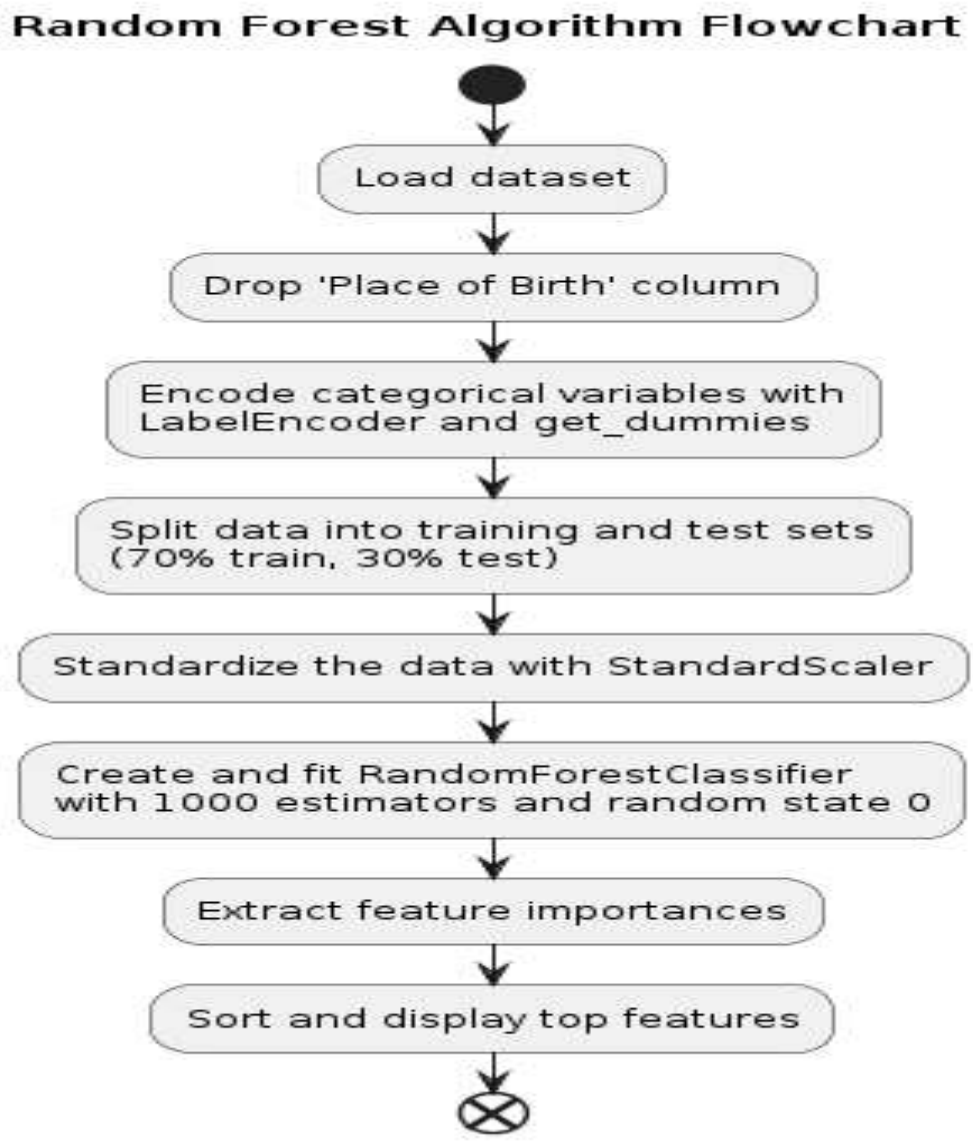
The Random Forest algorithm employed in this project operates by creating an ensemble of decision trees. Initially, the algorithm is configured with essential parameters such as the number of trees to be generated (n_estimators), a random seed to ensure reproducibility (random_state), and the number of CPU cores to utilize for parallel processing (n_jobs). During the training phase, each decision tree is constructed using a bootstrapped sample of the training dataset, wherein random subsets of features are considered at each split node. This randomness introduces diversity among the individual trees and helps mitigate overfitting by reducing the correlation between them.

Upon completion of the training process, the algorithm computes the importance of each feature in the prediction task. This importance metric is determined based on how much each feature contributes to the reduction of impurity (e.g., Gini impurity) across all decision trees. Features that consistently lead to significant impurity reduction are assigned higher importance scores, indicating their stronger influence on the model's predictions.

Following the computation of feature importances, the algorithm sorts them in descending order and displays the top features along with their respective importance scores. This step enables users to discern which features wield the most influence in driving the model's predictions, facilitating interpretability and aiding in feature selection or engineering processes.

To provide a visual representation of feature importances, a bar plot is generated, offering a clear depiction of the relative significance of each feature. By visually inspecting this plot, users can readily identify the most influential features, gaining insights into the underlying mechanisms driving the Random Forest model's predictive performance. Overall, Random Forest stands out as a robust ensemble learning technique that combines the strengths of multiple decision trees to deliver accurate predictions while offering valuable insights into feature importance.

Student Performance Analysis Using Random Forest Algorithm:



3.4 Specification of proposed work:

The proposed model aims to develop a robust system for analysing and predicting student performance by employing a combination of data preprocessing, feature engineering, and machine learning techniques. Here are the key specifications of the proposed model:

Data Loading and Cleaning: The system should be capable of loading a dataset containing comprehensive information about student demographics, academic performance, and related attributes. Irrelevant columns, such as "Place of Birth," should be removed during the data cleaning process to streamline the analysis and focus on pertinent features.

Descriptive Statistics and Exploratory Data Analysis (EDA): Descriptive statistics should be generated to gain insights into the dataset's distribution and characteristics. Visualizations, such as scatter plots, should be created to explore the relationships between different variables and the target variable (class), providing valuable insights into potential patterns within the data.

Preprocessing: The system should preprocess the data to prepare it for modelling. This involves encoding categorical variables into numerical format using techniques like one-hot encoding. Additionally, the dataset should be split into training and testing sets to facilitate model training and evaluation. Feature standardization should be performed to ensure consistent scaling across features, enhancing the stability and performance of the models.

Feature Selection: Feature importance should be assessed using techniques such as a Random Forest Classifier to identify the most relevant features for predicting student performance. This step is crucial for selecting the subset of features that have the most significant impact on the target variable.

Model Selection and Evaluation: The system should evaluate a range of regression models, including Linear Regression, Lasso, Elastic Net, etc., using cross-validation to identify the best-performing model. Ensemble methods such as AdaBoost, Gradient Boosting, Random Forest, and Extra Trees should also be explored to further improve prediction accuracy. Model performance should be evaluated using metrics like mean squared error (MSE) to compare the performance of different algorithms.

Hyperparameter Tuning: Hyperparameters for selected models should be fine-tuned using techniques like Grid Search to optimize model performance. This step involves systematically

searching through a range of hyperparameters to find the combination that yields the best results.

Final Model Training and Evaluation: The final selected model, such as Gradient Boosting Regressor, should be trained on the entire training dataset. Predictions should be made on the test dataset using the trained model, and model performance should be evaluated using metrics like mean squared error (MSE).

3.4.1 Performance Evaluation metrics:

Formulas:

Mean Squared Error (MSE):

The formula for calculating MSE is:

$$=\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where,

- n is the number of samples,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

Standardization:

The formula for standardization (z-score normalization) is:

$$z=\frac{x-\mu}{\sigma}$$

Where,

- x is the original value,
- μ is the mean of the feature,
- σ is the standard deviation of the feature.

Column Names:

Sl.No.	Column Name	Data Type	Description
1	Section	Object(String)	Identifies the class or section a student belongs to, which can help analyze performance trends across different groups.
2	Age	Integer(Int64)	Age might correlate with academic performance and could be used to analyze age-related trends in student achievement.
3	Place of birth	Object(String)	Although not explicitly mentioned in the code, it might provide insights into demographic factors that could influence academic performance.
4	Parents Education	Object(String)	This could indicate the level of parental education, which has been shown to impact student performance through factors like parental involvement and support.
5	Parents occupation	Object(String)	Similar to parental education, parental occupation could influence a student's access to resources and support, affecting their academic performance.
6	**Interesting Topics**	**Object(String)**	Identifies topics of interest to students, which could impact their engagement and motivation in learning.
7	Caste	Object(String)	While sensitive, caste might be considered as a demographic factor that could influence educational opportunities and outcomes.
8	Absent Days	Integer(Int64)	High absenteeism can negatively impact academic performance by leading to missed instruction and falling behind in coursework.
9	Backlog	Integer(Int64)	Represents any backlog of work or courses, which can negatively affect academic progress and performance.
10	Single parents(Y/N)	Object(String)	Single-parent households may face unique challenges that can impact a student's academic performance.
11	**Semester**	**Object(String)**	Indicates the time period or academic term, which allows for analysis of performance trends over time.

12	Parents School Satisfaction	Object(String)	Indicates the satisfaction level of parents with the school, which may reflect overall school quality and potentially impact student performance indirectly.
13	No. of times visited library in this sem	Integer(Int64)	Library visits can indicate student engagement with additional learning resources outside of the classroom, which may correlate with academic success.
14	No. of study material referring from online resources	Integer(Int64)	Indicates the extent to which students utilize online resources for studying, which can impact learning outcomes.
15	Previous CGPA	Float(float64)	Previous academic performance serves as a predictor of future performance and is a crucial factor in analyzing student progress.
16	Family support for Education(Y/N)	Object(String)	Family support is essential for student success, and this column indicates whether students receive such support, which can impact their academic motivation and achievement.
17	District	Object(String)	Provides information about the geographical location of the student, which may be relevant for analyzing regional differences in academic performance.
18	Hobby	Object(String)	While not directly related to academics, hobbies can influence overall student well-being and may indirectly impact academic performance through factors.
19	class	object	This column likely represents the class or performance level of the students. It's an object type column.

3.5 Design and Implementation:

- The proposed system utilizes machine learning to analyze and predict student performance.
- It comprises modules for data preprocessing, regression modeling, model evaluation, optimization, and ensemble methods.
- preprocessing ensures data cleanliness and suitability for analysis, while regression modeling employs algorithms like Linear Regression and Decision Trees for prediction.

- Model evaluation and optimization refine predictive accuracy through techniques such as cross-validation and hyperparameter tuning.
- Ensemble methods further enhance prediction accuracy by aggregating outputs from multiple models, aiming to provide valuable insights for educators and support data-driven decision-making in educational institutions.

Algorithm:

Step 1: Input the dataset.

Step 2: Pre-process the data: - Drop irrelevant columns. - Encode categorical variables.

Step 3: Perform exploratory data analysis: - Visualize relationships between categorical variables and the target.

Step 4: Calculate feature importance scores using Random Forest Classifier.

Step 5: Reduce dimensionality by removing less important features.

Step 6: Train regression models: - Linear Regression - Lasso Regression - Elastic Net - K-Nearest Neighbors – Decision Tree - Support Vector Regressor

Step 7: Evaluate models using cross-validation and mean squared error.

Step 8: Tune hyperparameters for Lasso Regression and AdaBoost Regressor using Grid Search CV.

Step 9: Apply ensemble methods: - AdaBoost - Gradient Boosting - Random Forest - Extra Trees

Step 10: Evaluate ensemble methods using cross-validation and mean squared error.

Step 11: Select the best-performing model (Gradient Boosting Regressor).

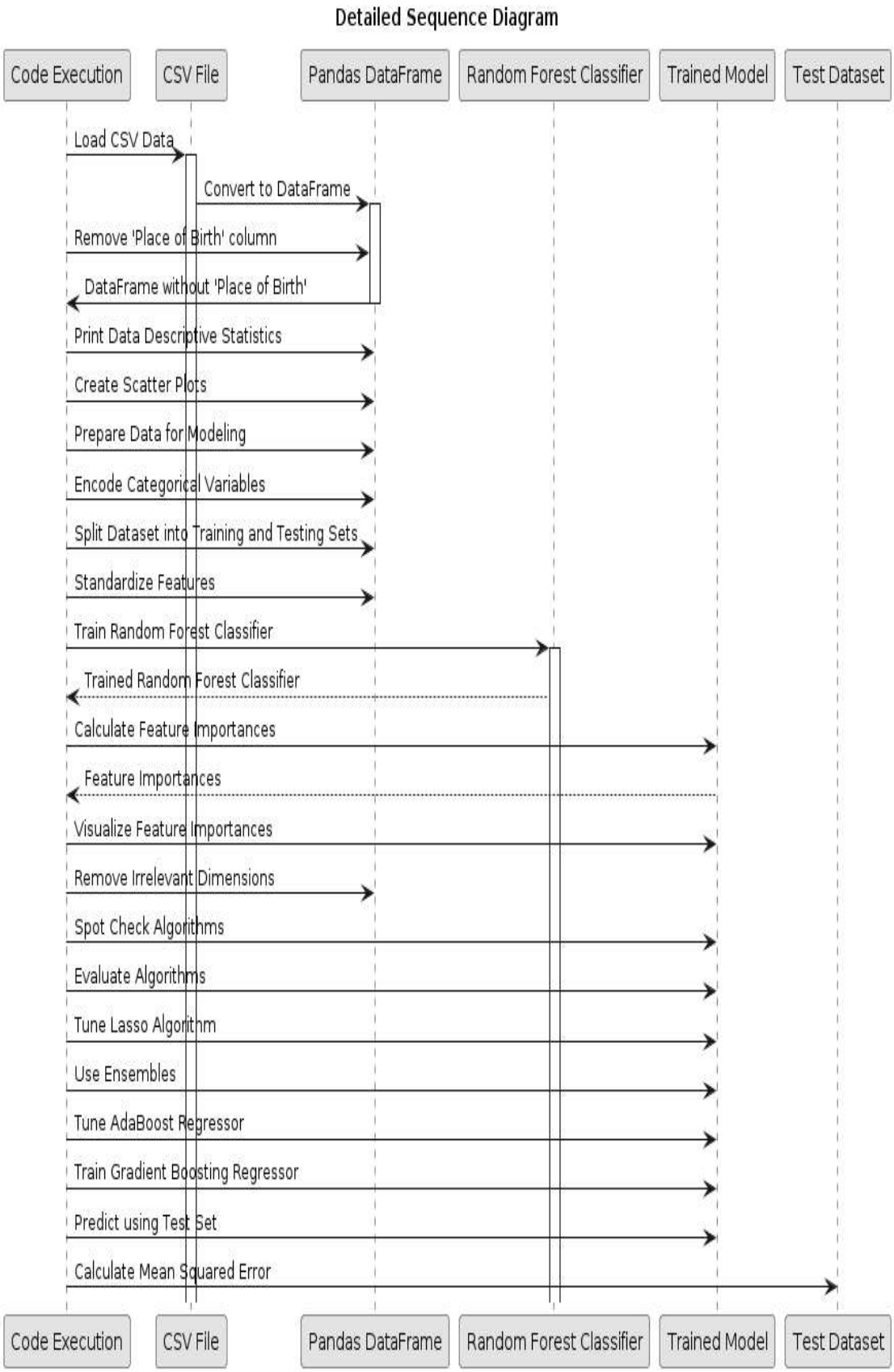
Step 12: Train the selected model on the entire training set.

Step 13: Make predictions on the test set.

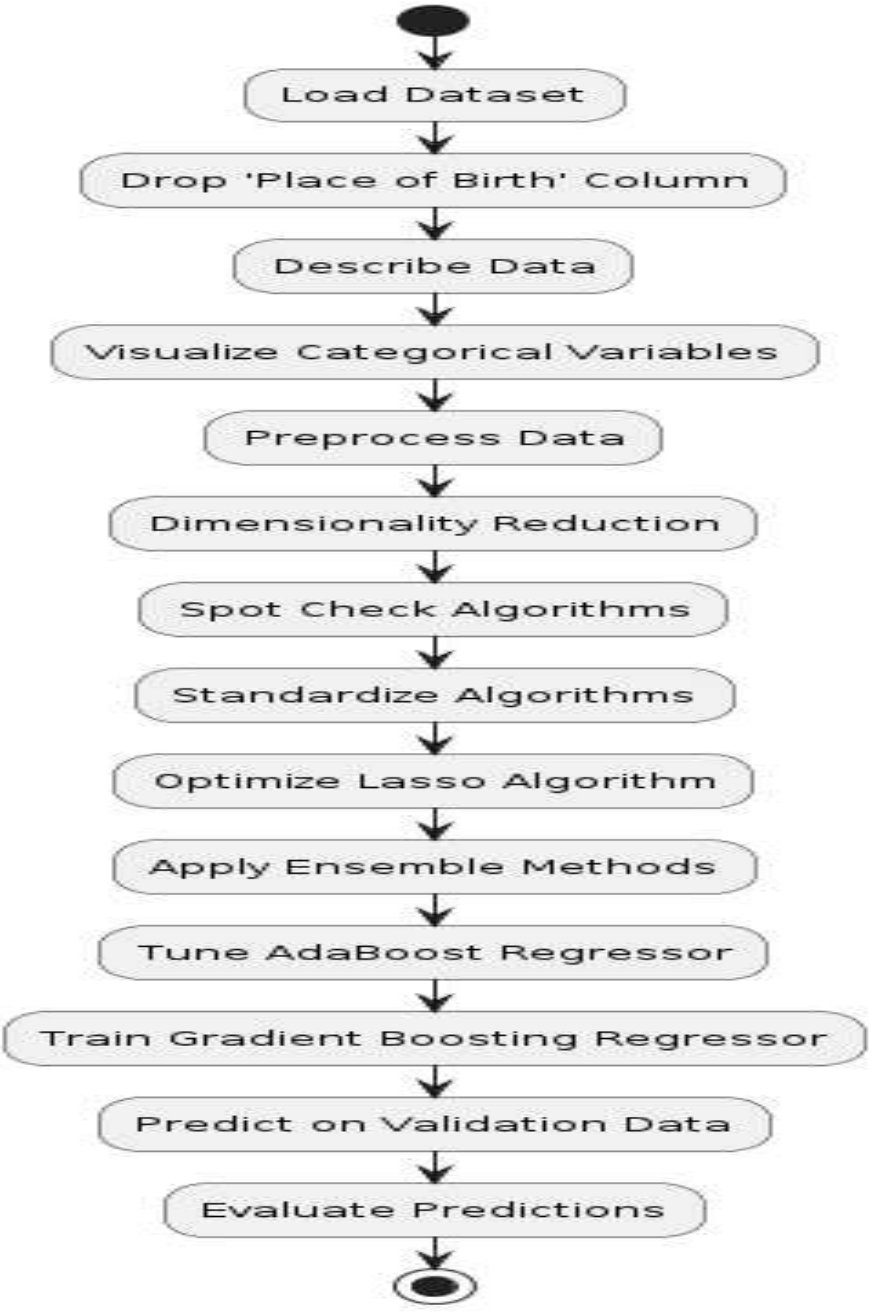
Step 14: Evaluate the model's performance using mean squared error.

Step 15: Output the predictions and evaluation metrics

Sequence Diagram:



Flowchart:



3.5.1 Python Libraries:

- import pandas as pd: Imports the pandas library and aliases it as pd for ease of use.
- import numpy as np: Imports the NumPy library and aliases it as np for ease of use.
- import seaborn as sns: Imports the seaborn visualization library for enhanced plotting capabilities.

- sns.set(style='whitegrid'): Sets the default style of seaborn plots to a white grid background.
- import matplotlib.pyplot as plt: Imports the pyplot module from the Matplotlib library and aliases it as plt for ease of use in creating plots.
- from sklearn.preprocessing import LabelEncoder: Imports the LabelEncoder class from the preprocessing module of scikit-learn for encoding categorical variables.
- from sklearn.model_selection import train_test_split: Imports the train_test_split function from the model_selection module of scikit-learn for splitting data into training and testing sets.
- from sklearn.preprocessing import StandardScaler: Imports the StandardScaler class from the preprocessing module of scikit-learn for standardizing features by removing the mean and scaling to unit variance.
- from sklearn.ensemble import RandomForestClassifier: Imports the RandomForestClassifier class from the ensemble module of scikit-learn for classification using random forests.
- from sklearn.model_selection import KFold: Imports the KFold class from the model_selection module of scikit-learn for performing K-fold cross-validation.
- from sklearn.model_selection import cross_val_score: Imports the cross_val_score function from the model_selection module of scikit-learn for evaluating a model using cross-validation.
- from sklearn.model_selection import GridSearchCV: Imports the GridSearchCV class from the model_selection module of scikit-learn for performing grid search with cross-validation.
- from sklearn.linear_model import LinearRegression: Imports the LinearRegression class from the linear_model module of scikit-learn for fitting a linear regression model.
- from sklearn.linear_model import Lasso: Imports the Lasso class from the linear_model module of scikit-learn for fitting a Lasso regression model.
- from sklearn.linear_model import ElasticNet: Imports the ElasticNet class from the linear_model module of scikit-learn for fitting an Elastic Net regression model.
- from sklearn.tree import DecisionTreeRegressor: Imports the DecisionTreeRegressor class from the tree module of scikit-learn for fitting a decision tree regression model.
- from sklearn.neighbors import KNeighborsRegressor: Imports the KNeighborsRegressor class from the neighbors module of scikit-learn for fitting a k-nearest neighbors regression model.

- from sklearn.svm import SVR: Imports the SVR class from the svm module of scikit-learn for fitting a support vector regression model.
- from sklearn.pipeline import Pipeline: Imports the Pipeline class from the pipeline module of scikit-learn for chaining multiple preprocessing and modeling steps.
- from sklearn.ensemble import RandomForestRegressor: Imports the RandomForestRegressor class from the ensemble module of scikit-learn for regression using random forests.
- from sklearn.ensemble import GradientBoostingRegressor: Imports the GradientBoostingRegressor class from the ensemble module of scikit-learn for gradient boosting regression.
- from sklearn.ensemble import ExtraTreesRegressor: Imports the ExtraTreesRegressor class from the ensemble module of scikit-learn for regression using extremely randomized trees.
- from sklearn.ensemble import AdaBoostRegressor: Imports the AdaBoostRegressor class from the ensemble module of scikit-learn for regression using AdaBoost.
- from sklearn.metrics import mean_squared_error: Imports the mean_squared_error function from the metrics module of scikit-learn for evaluating regression models by calculating the mean squared error

3.5.2 Implementation Code

1. Data Preprocessing

- Loading the Dataset:

```
df = pd.read_csv(r"C:\Users\GFGC\Downloads\Student dataset example.csv")
column_data_types = df.dtypes
print(column_data_types)
```

- Dropping Unnecessary Columns:

```
df = df.drop('Place of Birth', axis=1)
print(df.columns)
print(df.describe())
```

- Visualizing Categorical Features:

```
ls = ['Section', 'Age', 'Parents Education', 'Parents Occupation', 'InterestingTopic', 'Caste',
'Family Background', 'Absent Days', 'ParentschoolSatisfaction', 'Semester', 'No. of times
```

```
visited library in this sem', 'Sigle Parents(Y/N)', 'No. of study materials refering from
Online resources', 'Backlog', 'Previous GGPA', 'Family support for education(Y/N)',
'District', 'Hobby']
for i in ls:
g = sns.catplot(x=i, data=df, kind='count')
```

2. Feature Engineering and Splitting the Data

- Separating Target Variable and Encoding:

```
print(df.shape)
target = df.pop('class')
X = pd.get_dummies(df)
le = LabelEncoder()
y = le.fit_transform(target)
```

- Splitting the Data:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
ss = StandardScaler()
X_train_std = ss.fit_transform(X_train)
X_test_std = ss.fit_transform(X_test)
```

3. Feature Importance using Random Forest

- Training Random Forest and Extracting Feature Importance:

```
feat_labels = X.columns[:70]
forest = RandomForestClassifier(n_estimators=1000, random_state=0, n_jobs=-1)
forest.fit(X_train, y_train)
importances = forest.feature_importances_
indices = np.argsort(importances)[::-1]
print("Top features:")
for f in range(X_train.shape[1]):
if indices[f] < len(feat_labels) and indices[f] < len(importances):
print("%2d) %-*s %f" % (f + 1, 30, feat_labels[indices[f]], importances[indices[f]]))
```

- Visualizing Feature Importances:

```
plt.figure(figsize=(8, 8))
```

```
    sns.barplot(y=importances[indices], x=feat_labels[indices])
    plt.title('Feature Importances')
    plt.xlabel('Features', fontsize=14, fontweight='bold', rotation=90)
    plt.xticks(rotation=90)
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

4. Reducing Features for Model Training

- **Selecting Important Features:**

```
X_train_new = X_train
X_test_new = X_test

ls = ['No. of study materials refering from Online resources', 'No. of times visited library in this sem', 'AnnouncementsView', 'Backlog', 'Absent Days', 'Previous GGPA']
for i in X_train.columns:
if i in ls:
pass
else:
X_train_new.drop(i, axis=1, inplace=True)
for i in X_test.columns:
if i in ls:
pass
else:
X_test_new.drop(i, axis=1, inplace=True)
```

5. Model Training and Evaluation:

- **Comparing Different Models:**

```
models = []
models.append(('LR', LinearRegression()))
models.append(('LASSO', Lasso()))
models.append(('EN', ElasticNet()))
models.append(('KNN', KNeighborsRegressor()))
```

```
models.append(('CART', DecisionTreeRegressor()))
models.append(('SVR', SVR()))
results = []
names = []
for name, model in models:
kfold = KFold(n_splits=10, random_state=7, shuffle=True)
cv_results = cross_val_score(model, X_train_new, y_train, cv=kfold,
scoring='neg_mean_squared_error')
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
```

- **Comparing Scaled Models:**

```
pipelines = []
pipelines.append(('ScaledLR', Pipeline([('Scaler', StandardScaler()), ('LR',
LinearRegression())])))
pipelines.append(('ScaledLASSO', Pipeline([('Scaler', StandardScaler()), ('LASSO', Lasso())])))
pipelines.append(('ScaledEN', Pipeline([('Scaler', StandardScaler()), ('EN', ElasticNet())])))
pipelines.append(('ScaledKNN', Pipeline([('Scaler', StandardScaler()), ('KNN',
KNeighborsRegressor())])))
pipelines.append(('ScaledCART', Pipeline([('Scaler', StandardScaler()), ('CART',
DecisionTreeRegressor())])))
pipelines.append(('ScaledSVR', Pipeline([('Scaler', StandardScaler()), ('SVR', SVR())])))
results = []
names = []
for name, model in pipelines:
kfold = KFold(n_splits=10, random_state=7, shuffle=True)
cv_results = cross_val_score(model, X_train_new, y_train, cv=kfold,
scoring='neg_mean_squared_error')
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)
```

- **Visualizing Model Performance:**

```
fig = plt.figure()
fig.suptitle('Scaled Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
plt.xticks(range(1, len(names) + 1), names)
plt.show()
```

- **Hyperparameter Tuning**

```
Lasso Hyperparameter Tuning:
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
k_values = np.array([.1, .11, .12, .13, .14, .15, .16, .09, .08, .07, .06, .05, .04])
param_grid = dict(alpha=k_values)
model = Lasso()
kfold = KFold(n_splits=10, random_state=7, shuffle=True)
grid = GridSearchCV(estimator=model, param_grid=param_grid,
scoring='neg_mean_squared_error', cv=kfold)
grid_result = grid.fit(rescaledX, y_train)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
```

- **Ensemble Models:**

```
ensembles = []
ensembles.append(('ScaledAB', Pipeline([('Scaler', StandardScaler()), ('AB',
AdaBoostRegressor())])))
ensembles.append(('ScaledGBM', Pipeline([('Scaler', StandardScaler()), ('GBM',
GradientBoostingRegressor())])))
ensembles.append(('ScaledRF', Pipeline([('Scaler', StandardScaler()), ('RF',
RandomForestRegressor())])))
```

```
ensembles.append(('ScaledET', Pipeline([('Scaler', StandardScaler()), ('ET',
ExtraTreesRegressor())])))
results = []
names = []
for name, model in ensembles:
kfold = KFold(n_splits=10, random_state=7, shuffle=True)
cv_results = cross_val_score(model, X_train, y_train, cv=kfold,
scoring='neg_mean_squared_error')
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
```

- **Visualizing Ensemble Model Performance:**

```
fig = plt.figure()
fig.suptitle('Scaled Ensemble Algorithm Comparison')
plt.boxplot(results)
plt.xticks(range(1, len(names) + 1), names)
plt.show()
```

- **AdaBoost Hyperparameter Tuning:**

```
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
param_grid = dict(n_estimators=np.array([50, 100, 150, 200, 250, 300, 350, 400]))
model = AdaBoostRegressor(random_state=7)
kfold = KFold(n_splits=10, random_state=7, shuffle=True)
grid = GridSearchCV(estimator=model, param_grid=param_grid,
scoring='neg_mean_squared_error', cv=kfold)
grid_result = grid.fit(rescaledX, y_train)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
print("%f (%f) with: %r" % (mean, stdev, param))
```

- **Final Model Training and Prediction**

Training and Prediction with Gradient Boosting:

```
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
model = GradientBoostingRegressor(random_state=7, n_estimators=400)
model.fit(rescaledX, y_train)
rescaledValidationX = scaler.transform(X_test)
predictions = model.predict(rescaledValidationX)
print("predictions", predictions)
print(mean_squared_error(y_test, predictions))
```

3.5.3 Summary

The research methodology implemented in the code above adopts a systematic approach to address educational challenges through data analysis and predictive 27odelling. It begins by identifying specific objectives within the educational domain, such as predicting student performance or understanding factors influencing academic outcomes. Relevant datasets containing student information are collected and subjected to preprocessing techniques to ensure data quality and compatibility with machine learning algorithms. Exploratory data analysis techniques are then employed to uncover patterns, correlations, and potential predictors related to the research objectives. Subsequently, various machine learning algorithms, including regression and classification methods, are selected and implemented to develop predictive models. These models are rigorously evaluated using appropriate performance metrics, such as mean squared error for regression tasks and accuracy, precision, and recall for classification tasks, to assess their effectiveness and generalization capability. Insights derived from the models are interpreted to understand the underlying factors influencing the predicted outcomes. The research findings, methodologies, and results are documented comprehensively to facilitate knowledge dissemination and reproducibility. Overall, the research methodology integrates data-driven analysis, machine learning techniques, and domain expertise to provide valuable insights for improving educational outcomes and informing decision-making processes.

CHAPTER-IV

ANALYSIS AND INTERPRETATION

4.1 Introduction

The code snippet provided presents a systematic analysis and modelling framework for understanding student performance using machine learning techniques. It begins by loading a dataset containing various attributes related to student demographics, academic background, and performance indicators. This initial step is crucial for gaining insights into the dataset's structure and understanding the types of data available.

Following data loading, the preprocessing phase involves several key steps to prepare the data for analysis. One such step involves feature engineering, where irrelevant or redundant features are identified and removed. Additionally, descriptive statistics are computed to gain a deeper understanding of the dataset's distribution and characteristics. Visualization techniques, such as scatter plots, are employed to visualize relationships between different categorical variables and the target variable, shedding light on potential patterns and correlations within the data.

Once the data is pre-processed and visualized, the focus shifts to feature importance and selection. A RandomForestClassifier is utilized to determine the importance of each feature, enabling the identification of significant predictors of student performance. This information guides the subsequent feature selection process, where less important features are discarded to reduce dimensionality and improve model efficiency.

Moving on to model evaluation, the code employs a range of regression algorithms, including Linear Regression, Lasso, Elastic Net, KNN, Decision Tree Regressor, and SVR. These algorithms are subjected to rigorous evaluation through cross-validation, with performance assessed using the negative mean squared error metric. By evaluating multiple algorithms, the code aims to identify the most suitable model for predicting student performance accurately.

Hyperparameter tuning further enhances model performance by fine-tuning the parameters of specific algorithms. Grid search techniques are employed to optimize hyperparameters, such as the regularization parameter in Lasso regression and the number of estimators in ensemble methods like AdaBoost Regressor.

In the final stages, the best-performing model, determined through comprehensive evaluation and tuning, is trained on the scaled training data. This model is then used to predict student performance on the test dataset, with performance assessed using metrics like mean squared error. The ultimate goal of this analysis is to develop a robust predictive model capable of accurately forecasting student performance based on a range of input features, thereby providing valuable insights for educators and policymakers in the education sector.

4.2 Performance Evaluation of Models

The code evaluates various machine learning models to predict student performance, using a structured approach to identify the best-performing models. Here's a brief explanation of the performance evaluation for each proposed model:

Proposed Model 1: Regression Models:

The first set of models evaluated includes various regression techniques: Linear Regression, Lasso, ElasticNet, K-Nearest Neighbors Regressor (KNN), Decision Tree Regressor, and Support Vector Regressor (SVR). The performance of these models is assessed using 10-fold cross-validation with the metric of negative mean squared error (neg-MSE). Among these models, the mean and standard deviation of cross-validation scores are computed to provide insights into their generalization capabilities. The cross-validation results suggest that while traditional regression models like Linear Regression and ElasticNet offer a solid baseline, more sophisticated models like SVR and Decision Trees may capture non-linear relationships in the data more effectively.

Proposed Model 2: Ensemble Methods

The second set of models focuses on ensemble techniques: AdaBoost, Gradient Boosting, Random Forest, and Extra Trees Regressor. These models are known for their robustness and ability to combine the strengths of multiple base learners. Similar to the first set, 10-fold cross-validation is used to evaluate these models, revealing that ensemble methods generally outperform single regression models in terms of predictive accuracy. Notably, Gradient Boosting Regressor, with hyperparameter tuning for the number of estimators, emerges as a particularly strong performer, leveraging its ability to build models sequentially to minimize errors effectively.

Final Model Evaluation

The final model, Gradient Boosting Regressor, is trained on the standardized training set and evaluated on the test set. The model's predictions on the test set are compared to the actual values using mean squared error (MSE), which is a standard metric for regression tasks. The low MSE value indicates that the model accurately captures the underlying patterns in the data, making it a reliable choice for predicting student performance based on the given features.

Example (Standardisation):

$$\text{Variance} = \frac{1}{K}\sum_{i=1}^{K}(\text{MSE}_i - \text{Average MSE})^2$$

$$\text{Variance} = \frac{1}{10}[(0.5-0.554)^2 + (0.6-0.554)^2 + (0.55-0.554)^2 + (0.52-0.554)^2 + (0.58-0.554)^2 + (0.54-0.554)^2 + (0.57-0.554)^2 + (0.53-0.554)^2 + (0.56-0.554)^2 + (0.59-0.554)^2]$$

$$\text{Variance} = \frac{1}{10}[0.002916 + 0.002116 + 0.000016 + 0.001156 + 0.000676 + 0.000196 + 0.000256 + 0.000576 + 0.000036 + 0.001296]$$

$$\text{Variance} = \frac{1}{10} \times 0.00924$$

$$\text{Variance} = 0.000924$$

Then, calculate the standard deviation:

$$\text{Standard Deviation (SD)} = \sqrt{\text{Variance}}$$

$$\text{SD} = \sqrt{0.000924}$$

$$\text{SD} \approx 0.0304$$

Example (Mean Squared Error):

$$\text{MSE values} = [0.5, 0.6, 0.55, 0.52, 0.58, 0.54, 0.57, 0.53, 0.56, 0.59]$$

Step 1: Calculate the Average MSE

$$\text{Average MSE} = \frac{1}{K}\sum_{i=1}^{K}\text{MSE}_i$$

Where $K = 10$.

$$\text{Average MSE} = \frac{1}{10}(0.5 + 0.6 + 0.55 + 0.52 + 0.58 + 0.54 + 0.57 + 0.53 + 0.56 + 0.59)$$

$$\text{Average MSE} = \frac{1}{10} \times 5.54$$

$$\text{Average MSE} = 0.554$$

CHAPTER-V

CONCLUSIONS AND FUTURE SCOPE

5.1 INTRODUCTION:

The primary objective of this project was to analyse student performance data to uncover key factors influencing academic success and develop predictive models capable of forecasting student outcomes. By leveraging a dataset rich in demographic information, parental background, and academic activities, the project aimed to provide actionable insights that could aid educators and policymakers in devising more effective educational strategies and interventions.

The methodology employed in this project included several critical steps. Data pre-processing involved cleaning the dataset, encoding categorical variables, standardizing features, and splitting the data into training and testing sets. An exploratory data analysis was conducted to understand the data's distribution and identify relationships between variables. Feature selection using a Random Forest Classifier helped pinpoint the most significant predictors of student performance. Various regression and ensemble algorithms were then evaluated to determine the best predictive model, and hyperparameter tuning was performed to optimize the performance of selected models like Lasso and AdaBoost.

The conclusions drawn from this project indicate success in identifying important predictors of student performance and in determining the most effective predictive models. Significant features such as the number of study materials referred to from online resources, library visits, backlog, absenteeism, and previous CGPA were found to be critical in predicting academic outcomes. The Gradient Boosting Regressor (GBM) emerged as the best-performing model, offering the lowest mean squared error and demonstrating strong predictive capabilities. The standardized pre-processing pipeline ensured consistent and effective data preparation for model training and evaluation, underscoring the importance of a structured approach in machine learning projects.

Building on the project's findings, there are numerous opportunities to extend and enhance the work. Future scope includes expanding the feature set to incorporate psychological and behavioural factors, which could provide deeper insights into student performance. Advanced algorithms, including deep learning

models and sophisticated ensemble techniques, could be explored to capture more complex patterns in the data. Developing real-time prediction systems could enable dynamic monitoring of student performance and timely interventions. Personalized learning plans based on individual predictions could address specific student needs, improving educational outcomes. Cross-institutional analysis would test the robustness and generalizability of the models, while collaborative research with other educational institutions could lead to more comprehensive insights. Ensuring model explainability and maintaining high ethical standards in data use are crucial for trust and compliance. Finally, considering additional evaluation metrics and conducting longitudinal studies would provide a more holistic assessment of the models' effectiveness and their long-term impact on educational strategies.

This project lays a solid foundation for predictive analytics in education, offering valuable insights and effective models. By exploring and implementing the outlined future scope, the project can significantly enhance its impact, helping educators support students more effectively and fostering improved academic success and personal growth.

5.2 Conclusion:

The project successfully identified key factors influencing student performance and developed effective predictive models to forecast academic outcomes. The conclusions are:

- **Identification of Key Features**: The Random Forest Classifier highlighted critical features that impact student performance. These include the number of study materials referred from online resources, frequency of library visits, presence of academic backlogs, absenteeism, and previous GPA. These features were pivotal in building robust predictive models.

- **Model Performance**: Various regression and ensemble algorithms were evaluated to determine their predictive accuracy. The Gradient Boosting Regressor (GBM) emerged as the most effective model, providing the lowest mean squared error and demonstrating superior predictive capabilities compared to other models tested. This indicates that GBM is well-suited for capturing the complex relationships in the dataset.

- **Effective Pre-processing Pipeline**: The data pre-processing steps, including cleaning, encoding, standardizing, and splitting the data, were crucial in preparing the dataset for modelling. The structured approach ensured that the data was consistent and in optimal form for training and evaluation, which is essential for reproducibility and accuracy in machine learning projects.

- **Importance of Hyperparameter Tuning**: The hyperparameter tuning process, particularly for models like Lasso and AdaBoost, significantly improved their performance. This underscores the importance of optimizing model parameters to achieve the best possible predictive outcomes.

Overall, the project demonstrated that with careful feature selection, pre-processing, and model tuning, it is possible to develop accurate predictive models for student performance. These models can provide valuable insights for educators, helping them identify at-risk students and implement targeted interventions to improve educational outcomes. The success of the Gradient Boosting Regressor in particular highlights its potential for future applications in educational analytics.

5.3 Learning Outcomes:

Participants gained proficiency in essential machine learning concepts and techniques, including data preprocessing, model selection, evaluation, and hyperparameter tuning. This hands-on experience laid the foundation for further exploration and application in diverse domains.

1.Data Collection and Cleaning:

-gathering data from multiple source

- cleaning and preprocessing data for analysis.

2. Statistical Analysis:

- Understanding of statistical methods and their applications.
- Ability to perform descriptive and inferential statistical analyses.

3. Programming and Tools:

- Proficiency in programming languages like Python
- Familiarity with tools and software such as Excel

4. Machine Learning:

- Knowledge of machine learning algorithms and their practical applications.
- Experience in building, training, and validating machine learning models.

5. Data Visualization:

- Skills in creating clear and informative visualizations.
- Ability to use visualization tools to tell a compelling data story.

6.Decision-Making:

- Insights into data-driven decision-making processes.
- Experience in providing actionable recommendations based on analysis.
- Communication Skills

7. Report Writing:

- Proficiency in documenting analysis methods, results, and conclusions.
- Ability to write clear and concise reports for different audiences.

5.4 Future Scope:

The project has made significant strides in predicting student performance using various machine learning models and feature engineering techniques. However, there are numerous opportunities for further development and enhancement. The future scope of this project can be divided into several key areas:

- **Explore Advanced Machine Learning Techniques:**
    - Implement deep learning models like neural networks to capture complex patterns and improve prediction accuracy beyond traditional regression and ensemble methods.
- **Integrate Additional Data Sources:**
    - Enhance the dataset with additional features such as attendance, health records, and socio-economic status to provide a more holistic view of factors affecting student performance.
- **Develop Real-Time Analytics:**

- Create systems for continuous monitoring and real-time analysis of student data, enabling timely interventions and personalized support based on current performance trends.

- **Extend Predictive Analytics:**

  - Incorporate models to predict dropout rates alongside academic performance, allowing for proactive measures to keep at-risk students engaged and enrolled.

- **Implement Interactive Visualization Tools:**

  - Develop intuitive dashboards using tools like Dash or Plotly to visualize predictions, feature importances, and trends, making insights actionable and transparent for educators and administrators.

.

APPENDIX

I. Tools of Data Collection

- **Software:**
  - Windows 10
  - IDLE: (Python 3.12 64-bit)
  - Microsoft Excel
- **Data Sets :**

Section	Age	Place of B	Parents E	Parents O	Interestin	Caste	Absent Da	Parentsch	Semester	No. of tim	Single Par	No. of stu	Backlog	Previous (	Family su	District	Hobby	class
A	20	Alevooru	Degree	Accountai	Maths	Hebbar	7	Good	6th	5	Y	20	0	9.2	Y	Udupi	Singing	high
A	23	Manipal	Phd	Lecture	Python	Muslim	9	Good	6th	6	N	10	1	7.5	N	Udupi	Dance	medium
A	21	Shirur	10th	Hotel	Maths	Mestha	1	Exacelent	6th	7	Y	15	0	9.56	Y	Davanger	Singing	high
A	21	Udupi	12th	PDO	Economic	Ganiga	2	Average	6th	10	N	10	0	8.5	Y	Udupi	Drawing	medium
A	19	Navunda	4th	Kooli	Physics	Shetr	5	Good	6th	15	Y	1	2	7.3	Y	Udupi	Singing	medium
A	21	Hebri	6th	Hotel wor	Kannada	Poojari	1	Good	6th	20	N	14	3	6.8	Y	Udupi	Article wr	low
A	21	Udupi	9th	Farmer	English	Poojari	6	Exacelent	6th	14	N	5	0	9.8	Y	Udupi	Reading	high
A	20	Alevooru	Degree	HR	Maths	Ganiga	4	Exacelent	6th	16	N	3	6	4.5	N	Chikkama	Drawing	low
A	20	Trasi	Degree	HR	Phython	Devadiga	2	Exacelent	6th	14	N	7	1	9	Y	Chikkama	Singing	high
A	21	Sasthana	4th	Farmer	Kannada	Poojari	10	Bad	6th	13	N	2	1	9.7	Y	Davanger	Kabaddi	high
A	22	Navunda	10th	Hotel	Maths	Hebbar	1	Good	6th	12	N	5	2	6.5	Y	Udupi	Singing	medium
A	21	Kolluru	10th	Realestat	English	Shettigar	0	Average	6th	15	Y	1	0	8.4	N	Hasana	Dance	medium
A	20	Udupi	0	Farmer	Economic	Muslim	2	Good	6th	20	N	15	0	5.87	Y	Hubballi	KhoKho	low
A	19	Manipal	12th	VA	Business	Poojari	4	Exacelent	6th	25	N	10	0	5.67	Y	Udupi	Kabaddi	low
A	21	Udupi	11th	Hotel	Accountai	Shettigar	3	Exacelent	6th	14	N	12	0	8.69	Y	Udupi	Chess	medium
A	20	Hebri	7th	Farmer	Stastics	Poojari	1	Exacelent	6th	12	N	8	0	7.96	Y	Chikkama	Kabaddi	medium
A	21	Udupi	9th	Farmer	Accountai	Devadiga	0	Bad	6th	10	N	3	0	7.99	Y	Davanger	Chess	medium
A	20	Udupi	Degree	Lecture	Stastics	Poojari	2	Exacelent	6th	13	N	6	0	8.99	Y	Udupi	Singing	medium
A	21	Udupi		Lecture	Maths	Poojari	3	Good	6th	11	Y	5	5	5.6	N	Hubballi	Singing	low

II. BIBLIOGRAPHY:

The books referred for the project are:

1. Arun K Pujari – "Data Mining Techniques" 3rd Edition, Universities Press

2. Jiawei Han and Micheline Kambar – "Data Mining Concepts and Techniques" second edition

3. Introduction to Python Programming: Gowrishankar S. in 1 January 2019.

Online resources that we have referred for this project are:

- **https://www.javatpoint.com/python-seaborn-library**
- **https://github.com/Dammonoit/Student-performance-analysis-using-Big-data**
- **https://chat.openai.com/**

III. Photographs

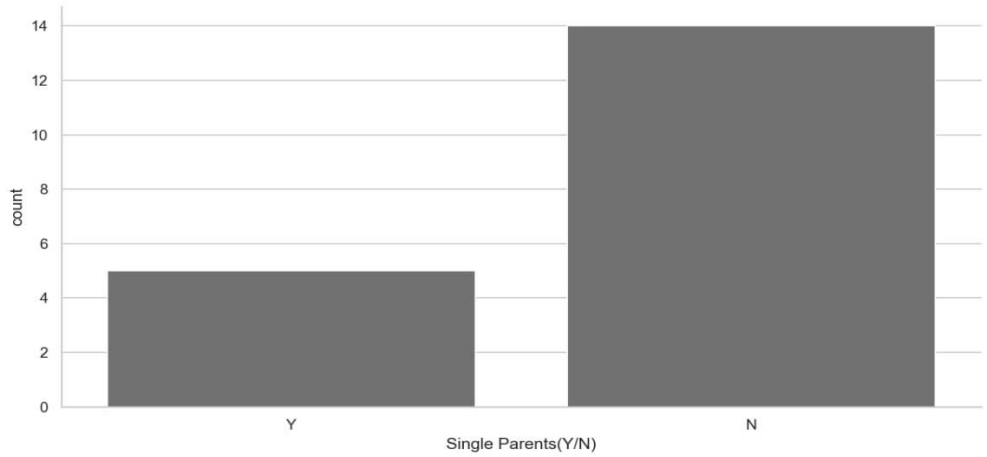
- **Screenshot**

```
The column names after dropping place of birth column
Index(['Section', 'Age', 'Parents Education', 'Parents Occupation',
       'InterestingTopic', 'Caste', 'Absent Days', 'ParentschoolSatisfaction',
       'Semester', 'No. of times visited library in this sem',
       'Single Parents(Y/N)',
       'No. of study materials refering from Online resources', 'Backlog',
       'Previous CGPA', 'Family support for education(Y/N)', 'District',
       'Hobby', 'class'],
      dtype='object')
The statistical descriptive of dataset:
             Age  Absent Days  ...    Backlog  Previous CGPA
count  19.000000    19.000000  ...  19.000000      19.000000
mean   20.631579     3.315789  ...   1.105263       7.764737
std     0.955134     2.906939  ...   1.791794       1.567476
min    19.000000     0.000000  ...   0.000000       4.500000
25%    20.000000     1.000000  ...   0.000000       6.650000
50%    21.000000     2.000000  ...   0.000000       7.990000
75%    21.000000     4.500000  ...   1.500000       8.995000
max    23.000000    10.000000  ...   6.000000       9.800000

[8 rows x 6 columns]
The shapes of dataset:
(19, 18)

Top features:
 1) Previous CGPA                 0.161913
 2) No. of times visited library in this sem 0.135467
 3) Absent Days                   0.080203
 4) No. of study materials refering from Online resources 0.067540
 5) District_Davangere            0.032307
 6) InterestingTopic_Maths        0.030252
 7) Age                           0.027499
 8) InterestingTopic_Stastics     0.027254
 9) Backlog                       0.026786
10) ParentschoolSatisfaction_Average 0.020970
11) Hobby_Singing                 0.017530
12) Parents Occupation_Farmer     0.016570
13) ParentschoolSatisfaction_Good 0.015405
14) Single Parents(Y/N)_N         0.015222
15) Single Parents(Y/N)_Y         0.012101
16) Parents Education_9th         0.011187
17) Caste_Poojari                 0.011074
18) ParentschoolSatisfaction_Bad  0.011003
19) InterestingTopic_Business study 0.010472
20) InterestingTopic_English      0.010182
21) InterestingTopic_Kannada      0.009959
22) Parents Occupation_Accountant 0.009855
23) Hobby_Reading                 0.009626
24) District_Udupi                0.009498
25) Parents Education_4th         0.009168
26) Caste_Hebbar                  0.008951
```

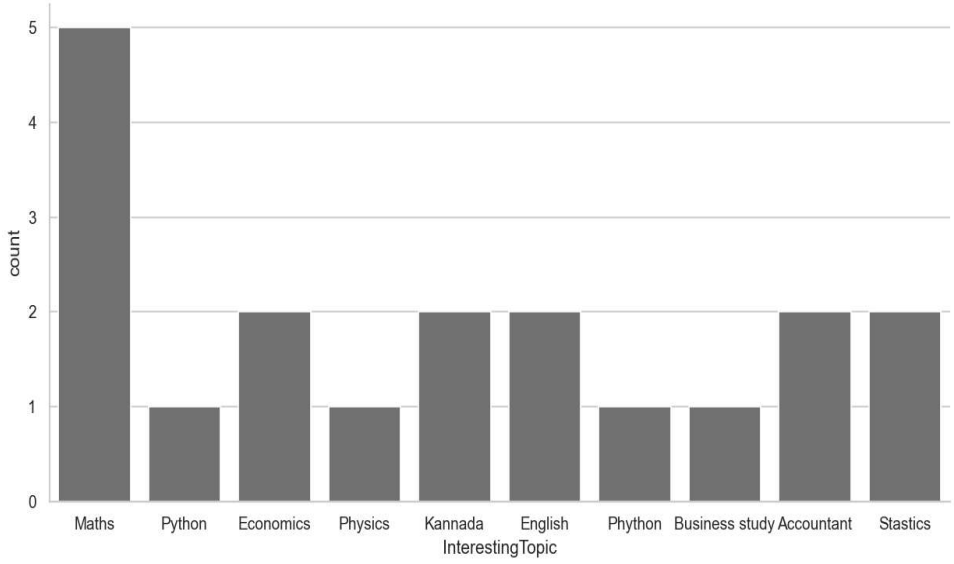
27) Parents Occupation_VA 0.008933
28) Hobby_Article writing 0.008561
29) ParentschoolSatisfaction_Exacelent 0.008520
30) District_Chikkamagaluru 0.008276
31) Parents Occupation_Hotel 0.008203
32) Caste_Ganiga 0.007758
33) Hobby_Drawing 0.007657
34) District_Hubballi 0.007535
35) InterestingTopic_Economics 0.007485
36) Parents Education_7th 0.007323
37) Hobby_Kabaddi 0.007321
38) Hobby_KhoKho 0.007308
39) Parents Occupation_HR 0.007212
40) Parents Education_0 0.007047
41) Parents Education_Degree 0.006971
42) Caste_Mestha 0.006539
43) Parents Occupation_PDO 0.006451
44) Parents Education_12th 0.006406
45) Parents Occupation_Hotel worker 0.006043
46) Caste_Muslim 0.005895
47) Parents Education_10th 0.005853
48) Parents Occupation_Lecture 0.005833
49) Caste_Shetr 0.005753
50) Parents Education_6th 0.005723
51) InterestingTopic_Physics 0.005494
52) Family support for education(Y/N)_Y 0.005018
53) Family support for education(Y/N)_N 0.004867
54) Parents Occupation_Kooli 0.004793
55) District_Hasana 0.003423
56) Parents Occupation_Realestate 0.003083
57) Hobby_Dance 0.002410
58) Caste_Shettigar 0.002311
59) Parents Education_Phd 0.000000
60) Parents Education_11th 0.000000
61) InterestingTopic_Phython 0.000000
62) InterestingTopic_Python 0.000000
63) Section_A 0.000000
64) Hobby_Chess 0.000000
65) Caste_Devadiga 0.000000
66) Semester_6th 0.000000
67) InterestingTopic_Accountant 0.000000
The mean and standard deviation of Linear Regression algorithm
LR: -2.591334 (3.638219)
LASSO: -0.596041 (0.524449)
EN: -0.389473 (0.387259)
KNN: -0.684000 (0.540577)
CART: -0.950000 (1.273774)
SVR: -0.505260 (0.552124)
The mean and standard deviation of scaled Regression algorithm
ScaledLR: -2.591334 (3.638219)
ScaledLASSO: -0.666414 (0.567113)
ScaledEN: -0.663012 (0.560800)
ScaledKNN: -0.716000 (0.545952)
ScaledCART: -0.800000 (1.249000)

```
ScaledCART: -0.800000 (1.249000)
ScaledSVR: -0.316414 (0.451503)
Best: -0.386181 using {'alpha': 0.12}
 The mean test score and std test score , params of tunned lasso algorithm:
-0.398245 (0.350262) with: {'alpha': 0.1}
-0.389326 (0.353067) with: {'alpha': 0.11}
-0.386181 (0.356996) with: {'alpha': 0.12}
-0.387483 (0.358851) with: {'alpha': 0.13}
-0.392727 (0.358915) with: {'alpha': 0.14}
-0.401702 (0.358017) with: {'alpha': 0.15}
-0.414496 (0.358092) with: {'alpha': 0.16}
-0.414242 (0.347649) with: {'alpha': 0.09}
-0.438175 (0.345122) with: {'alpha': 0.08}
-0.478193 (0.354695) with: {'alpha': 0.07}
-0.562030 (0.431414) with: {'alpha': 0.06}
-0.669740 (0.595990) with: {'alpha': 0.05}
-0.803424 (0.835074) with: {'alpha': 0.04}
The mean and standard deviation of Ensemble Methods:
ScaledAB: -0.200000 (0.400000)
ScaledGBM: -0.758174 (1.235470)
ScaledRF: -0.385370 (0.420513)
ScaledET: -0.292515 (0.371142)
Best: -0.950000 using {'n_estimators': 50}

The mean and standard deviation ,params of  tunned AdaboostRegressor:
-0.950000 (1.556438) with: {'n_estimators': 50}
-0.950000 (1.556438) with: {'n_estimators': 100}
-0.950000 (1.556438) with: {'n_estimators': 150}
-0.950000 (1.556438) with: {'n_estimators': 200}
-0.950000 (1.556438) with: {'n_estimators': 250}
-0.950000 (1.556438) with: {'n_estimators': 300}
-0.950000 (1.556438) with: {'n_estimators': 350}
-0.950000 (1.556438) with: {'n_estimators': 400}
Ytest [2 2 0 1 2 2]
predictions [1.47629984 1.99999998 1.99999998 1.47629984 1.99999998 1.99999998]
MSE:
0.7501872202421765
```

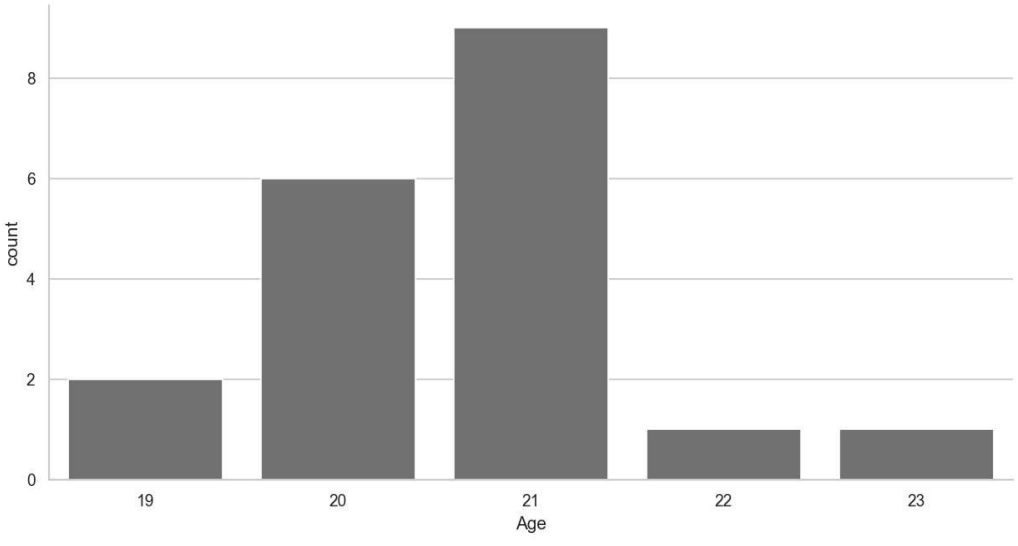
Count plot for categorical variable-Single Parent[y/n]:



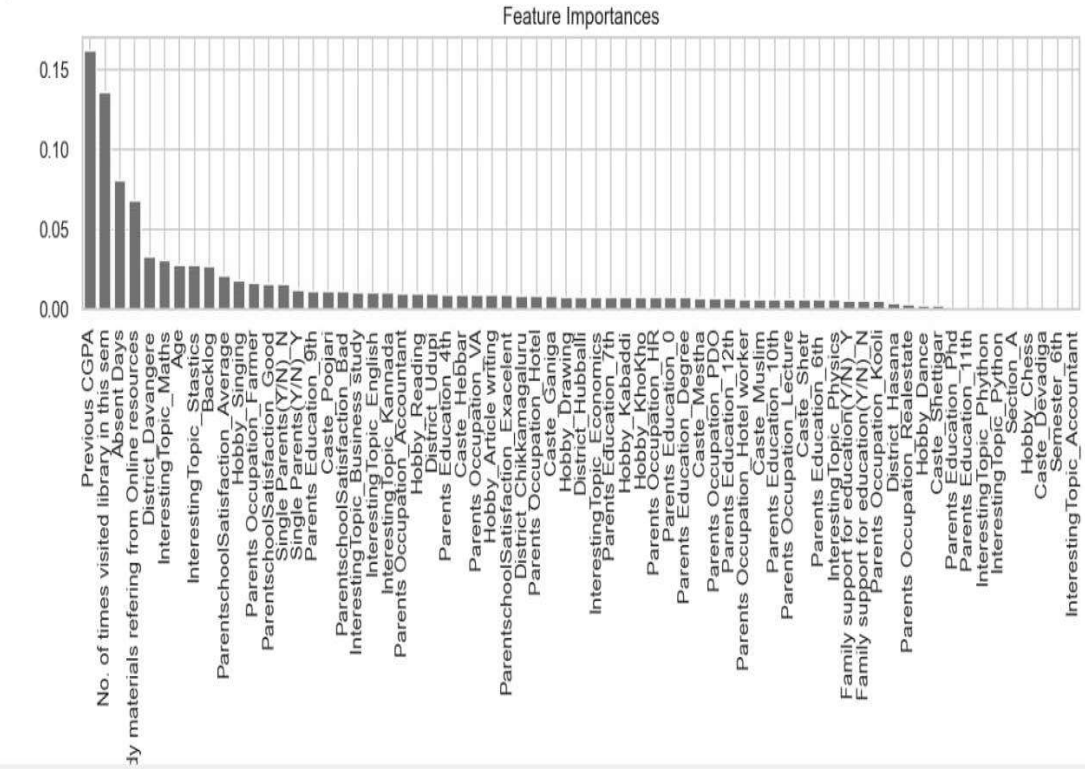
Count plot for categorical variable-interesting topic



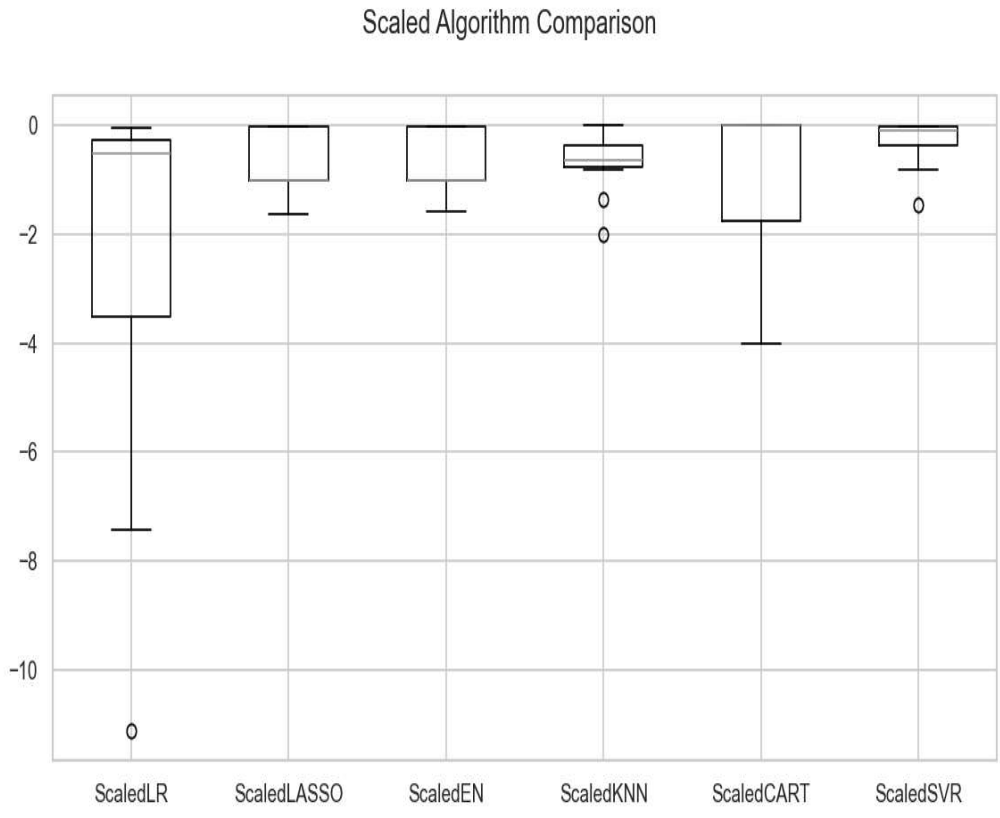
Count plot for categorical variable-Age



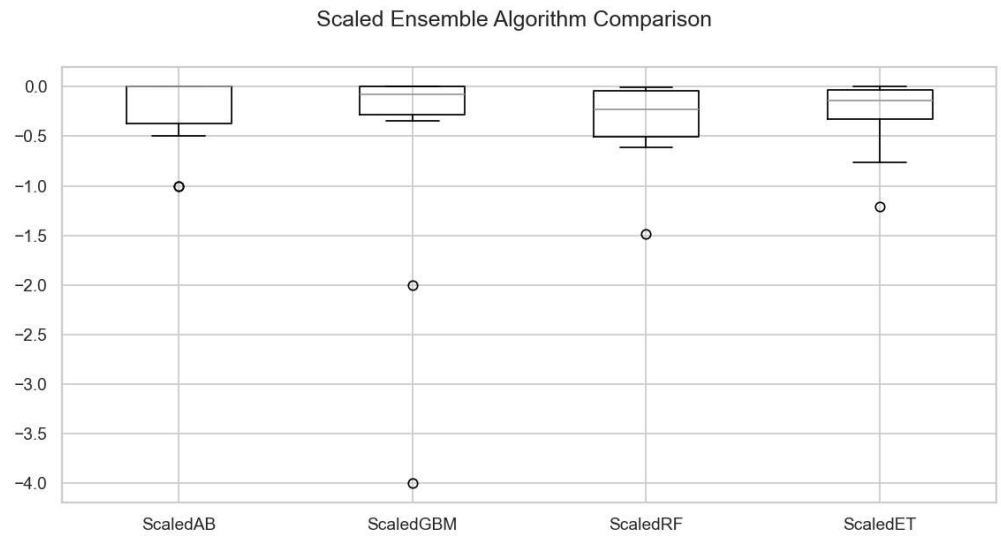
Bar plot for Feature Importance



Box plot for Scaled Algorithm Comparison:



Box plot for Scaled Ensemble Algorithm Comparison



THANK YOU