IJEMD-CSAI, 2 (1) (2023), 1 –71 https://doi.org/10.54938/ijemdcsai.2023.02.1.254

International Journal of Emerging Multidisciplinaries:
Computer Science and Artificial Intelligence

Research Paper
Journal Homepage: www.ojs.ijemd.com
ISSN (print): 2791-0164 ISSN (online): 2957-5036

Analysing Influential Factors in Student Academic Achievement: Prediction Modelling and Insight

Fahmida Faiza Ananna[1*], Ruchira Nowreen[1], Sakhar Saad Rashid Al Jahwari[1], Enzo Anindya Costa[1], Lorita Angeline[1] and Siva Raja Sindiramutty[1]

[1] Schools of Computer Science Taylor's University Subang Jaya, Selangor, Malaysia
**Corresponding author*

Abstract

The fascination with understanding student academic performance has drawn widespread attention from various stakeholders, including parents, policymakers, and businesses. The 'Students Performance in Exams' dataset, available on platforms like Kaggle, stands as a treasure trove. It extends beyond test scores, encompassing diverse student attributes like ethnicity, gender, parental education, test preparation, and even lunch type. In our tech-driven age, predicting academic success has become a compelling pursuit. This study aims to delve deep into this dataset, utilizing data mining methods and robust classification algorithms like Logistic Regression and Random Forest in a Jupyter Notebook environment. Rigorous model training, testing, and fine-tuning strive for the utmost predictive accuracy. Data cleaning and preprocessing play a crucial role in establishing a reliable dataset for accurate predictions. Beyond numbers, the project emphasizes data visualization's impact, transforming raw data into comprehensible insights for effective communication. The Logistic Regression Model exhibits an impressive 87.6% accuracy, highlighting its potential in predicting academic performance. Moreover, the Random Forest Model excels with a remarkable 100% accuracy in forecasting student grades, showcasing its effectiveness in this domain.

Keywords: Classification Algorithms; Data Mining Methods; Data Visualization; Jupyter Notebook.

Email addresses: **anannafaiza102@gmail.com (F.F.Ananna), ruchira.nowreen@sd.taylors.edu.my (R.Nowreen), sakharsaadrashidaljahwari@sd.taylors.edu.my (S.S.R.AlJahwari), anindyacosta@gmail.com (E.A.Costa), angeline.lorita@gmail.com (L.Angeline), siva.sindiramutty@taylors.edu.my (S.R.Sindiramutty)**

1. Introduction

A. Definition of Machine learning and Data Mining

Machine learning is a subset of artificial intelligence that focuses on developing algorithms and models that allow computers to learn and make predictions or decisions without being explicitly programmed (Sharifani, 2023; Zahra, Jhanjhi, Brohi, et al., 2022). It is all about creating systems that can learn from data and improve their performance over time. Machine learning algorithms use statistical techniques to identify patterns and relationships in data, which they then use to make predictions, classify objects, or automate decision-making processes (Kufel et al., 2023, Kar et al., 2015;). One of the fundamental aspects of machine learning is its ability to generalize from the data it is trained on to make predictions on new, unseen data (Omar et al., 2023; Dou et al., 2023). Common types of machine learning include supervised learning, unsupervised learning, and reinforcement learning (Pandey et al., 2023; Menon et al., 2023). Supervised learning involves training a model on labeled data to make predictions, while unsupervised learning deals with finding patterns in unlabeled data (Rani et al., 2023). Reinforcement learning is used for making sequential decisions, often in a dynamic environment (Moerland et al., 2023).

Data mining, on the other hand, is the process of discovering hidden patterns and valuable insights within large datasets (Saha & Rathore, 2022; Saeed et al., 2020). It involves the extraction of knowledge from data, including identifying trends, associations, clusters, and anomalies. Data mining techniques are often used to explore historical data and uncover meaningful information that can guide decision-making or improve business processes (S. Khan & Shaheen, 2021). Data mining operates at the intersection of various disciplines, including statistics, machine learning, and database management (Hamadani et al., 2023). It employs techniques like clustering, association rule mining, and outlier detection to uncover valuable information within data. Businesses and organizations frequently use data mining to enhance customer relationships, optimize operations, and make data-driven decisions (Jain et al., 2023; Zahra et al., 2022).

While machine learning and data mining share some similarities, their primary differences lie in their objectives and methodologies. Machine learning is more concerned with building predictive models and decision-making systems (Ma et al., 2023; Razaque et al., 2023), whereas data mining is focused on knowledge discovery and pattern identification within existing data (Tsui et al., 2023; Zaheer et al., 2022). In many cases, data mining can be considered a precursor to machine learning, as the insights gained from data mining can inform the feature selection, preprocessing, and model building stages of a machine learning project. In summary, machine learning and data mining are complementary fields that play essential roles in extracting knowledge and making informed decisions from data. Machine learning is more about building predictive models, while data mining is about exploring and uncovering insights from large datasets. Both are invaluable tools in today's data-driven world, with applications spanning industries such as healthcare, finance, e-commerce, and more.

B. Definition of Predictive Modeling, and Classification

Predictive modelling is a statistical and computational technique used to make predictions or forecasts about future events or outcomes based on historical data (Bharadiya, 2023; Chaudhary et al., 2022). It involves building a mathematical model that relates a set of input variables (features) to an output variable (target) (Priya et al., 2022). The goal is to learn and understand the underlying patterns and relationships in the data, which can then be used to predict future values or outcomes (Dhamala et al., 2023). In predictive modelling, the model is trained on a labelled dataset, which means the historical data includes both input features and the corresponding known outcomes (Alalawi et al., 2023; Simon et al., 2022., Adeyemo et al., 2019;). This dataset is divided into a training set for model development and a test set for model evaluation. Popular predictive modelling techniques include linear regression, decision trees, random forests, and neural networks (Wang et al., 2023). These models can be used for various tasks such as predicting sales, stock prices, disease diagnosis, customer churn, and much more.

Classification is a specific type of predictive modelling that focuses on assigning data points to predefined categories or classes (Zeineddine et al., 2021; Saleh et al., 2022 Sennan et al., 2021). It is used when the target variable is categorical, meaning it represents distinct groups. Classification models aim to learn decision boundaries in the feature space that separate these classes (Zhang et al., 2023; Alex et al., 2022). For example, it can be used to classify emails as spam or not spam, detect fraud in financial transactions, or identify the species of a plant based on its characteristics. Common classification algorithms include logistic regression, support vector machines, k-nearest neighbours, and deep learning models like convolutional neural networks (CNNs) for image classification (Yang et al., 2023; Shafiq et al., 2021, Shafiq at el., 2021, S. Verma at el., 2021). Evaluation metrics such as accuracy, precision, recall, and F1 score are used to assess the performance of classification models.

In summary, predictive modelling is a broader concept that encompasses various techniques for making predictions based on data, while classification is a specific type of predictive modelling focused on assigning data points to distinct categories. Both are essential tools in data science, providing valuable insights and enabling data-driven decision-making across a wide range of applications. Whether it's optimizing business operations, improving healthcare, or enhancing customer experiences, these techniques empower organizations to extract knowledge from data and drive better outcomes.

C. Importance of Data Visualization

Data visualization refers to the techniques used to convey data or information by representing it as visual elements within graphics (Alieva, 2021; Sujatha et al., 2021). The objective is to provide a deeper understanding of a dataset by presenting its key aspects in a more intuitive and meaningful manner than raw numbers alone. Data visualization can reveal patterns, trends, and correlations that might remain hidden when working with text-based data (George et al., 2023; Gaur et al., 2021., Gaur Afaq, et al, 2021; Gouda et al., 2022;). To facilitate a clear and efficient comprehension of information, data visualization employs statistical graphics, plots, information graphics, and various other tools (Hofer-Pottala, 2023, Hussain et al., 2019;). Analyzing data through tables and datasets can often be a time-consuming and labor-intensive process. In contrast, visualizations offer a more efficient way to grasp the intricate relationships

within the data, enabling users to quickly discern the nuances and patterns present in the information (U. Ali, 2023; Wassan et al., 2021). Numeric data can be represented using various visual elements, such as dots, lines, or bars, which enhance the visual interpretation of quantitative information (Singh et al., 2023). Well-crafted visualizations function as valuable tools that enable users to delve into and analyze data effectively, resulting in a deeper understanding of the numerical insights within the dataset (Alsamman et al., 2023; S. Lee et al., 2021; Lim et al., 2019, Lim et al., 2021). They make complex data more accessible, comprehensible, and usable. However, it's essential to note that the efficiency of data visualization relies significantly on the quality and accuracy of the underlying data. Ingesting incorrect data into the visualization engine can lead to erroneous, incomplete, or outdated data representations (Few & Edge, 2007).

The application of Python for data visualization plays a crucial role in the domain of data exploration and presentation (Mundargi et al., 2023; Fatima-Tuz-Zahra et al., 2019; Nanglia et al., 2022). This approach encompasses a comprehensive exploration of key libraries, such as matplotlib and seaborn, to create visually appealing and insightful data representations. For example, let's consider a research project focused on analyzing weather data. Python, combined with the matplotlib library, allows for the creation of visually striking temperature trend graphs that effectively convey seasonal variations over the years (Shouman, 2023). Additionally, Python's capabilities extend to generating sophisticated statistical graphics (Melitoshevich, 2023; Kok et al., 2020). For instance, a study centered on economic data can leverage Python to create regression plots that emphasize the correlations between variables, offering valuable insights for researchers and decision-makers. Furthermore, Python empowers users to manipulate time series data, enabling the development of interactive financial market visualizations (Cruz et al., 2023). In summary, this comprehensive approach equips data practitioners with a versatile toolkit for uncovering patterns, conveying insights, and making data-driven decisions across various domains, ranging from environmental sciences to economics."

D. Scope of the paper

The scope of this paper is to delve into the realm of academic achievement, a critical aspect of a student's educational journey that extends from primary school through college and university. This study, therefore, takes on the formidable task of meticulously analyzing and identifying the multifaceted factors that exert a substantial impact on a student's academic journey. By doing so, it aspires to not only comprehend these factors but also to predict students' performance in examinations, offering a roadmap towards enhancing educational outcomes. To accomplish this, the study harnesses the power of the "Students Performance in Exams" dataset from Kaggle, comprising of the academic marks secured by 1000 students across various subjects. This dataset comprises 1000 rows and 8 columns, with a keen focus on five independent variables examined in relation to the dependent variables represented by the scores in mathematics, reading, and writing. Through meticulous analysis, modeling, and evaluation, this study aims to distill meaningful insights that can inform educational practices, empower educators, and contribute to a deeper understanding of the intricate factors at play in academic achievement.

RELATED WORK

A. Taxonomy Mapping:

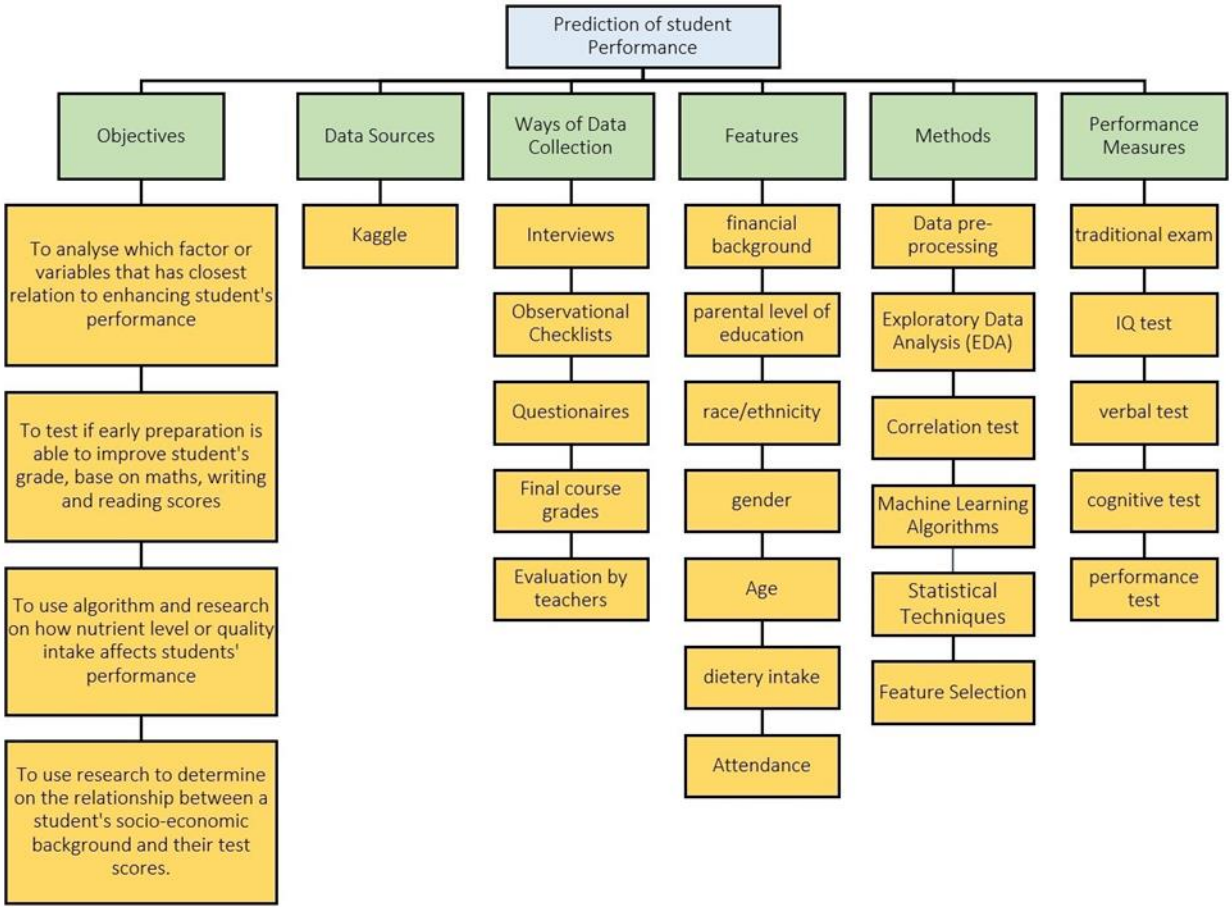

Figure 1: Taxonomy Mapping

In the taxonomy framework described above, our research journey commences with the initial branch, where we obtained our articles and datasets, primarily sourcing our data from the reputable platform Kaggle. Moving along the second branch, we explore the diverse methods used for data collection, which encompass interviews, observational checklists, questionnaires, final course grades, and teacher evaluations. This reflects the multifaceted nature of our research. The third branch centers on the crucial feature selection process, aiming to understand the impact of various variables on student performance. These variables include elements such as financial background, parental education levels, race/ethnicity, gender, age, dietary habits, and attendance records. In the subsequent branch, we delve into the various metrics and techniques employed to measure student performance. These encompass a range of assessments, including IQ tests, verbal tests, cognitive evaluations, performance assessments, and traditional exams. Transitioning to the fifth branch, our focus shifts to data preprocessing, a vital step to ensure the integrity and impartiality of the collected data. Here, we meticulously carry out tasks like data cleansing, handling duplicate and missing values, managing outliers, data transformation, label encoding, and targeted data extractions.

Continuing down the research journey, the next branch reveals the array of statistical methods used for in-depth data analysis. We employ Exploratory Data Analysis (EDA) techniques to uncover underlying patterns, and utilize correlation tests, including tools such as correlation coefficients, chi-square tests, and p-values, to gain valuable insights. Within the branch dedicated to machine learning algorithms, we showcase the various models constructed, which include multiple linear regression, polynomial regression, logistic regression, and the robust random forest algorithm. In the final stage of our research expedition, the last branch sheds light on the assessment of model accuracy. We employ the confusion matrix to rigorously evaluate the performance and precision of our models. This comprehensive framework guides our research, from data collection through the intricate stages of analysis and modeling, fostering a holistic understanding of the factors that influence student performance.

B. Critical Review

The journal article authored by Masini et al. (2022), focuses on the 'Relationship between Nutritional Status and Cognitive Performance among Primary School Students.' The aim of the study is to assess the influence of nutritional status on the cognitive performance of school-age children. Data collection involved personal data gathering and various tests, including anthropometric tests, dietary intake assessments, and cognitive tests, such as the Wechsler IQ test, verbal tests, and performance assessments.

The study's conclusion highlights that students' body fat levels have a long-term impact on their cognitive performance, potentially reducing their ability to focus. Moreover, the development of students' brains and bodies is strongly correlated with the intake of essential micronutrients like iodine, zinc, and vitamins. The study also revealed that a healthy diet led to improvements in children's verbal IQ, with a mean score of 98.43±0.99, performance IQ, with a mean score of 95.09±1.14, and total IQ, with a mean score of 96.69±0.42. Conversely, high sugar and fat intake were associated with decreased performance. In conclusion, a satisfactory diet and healthy nutrition play a crucial role in supporting normal brain development, especially during critical growth stages.

Another study conducted by Gemechu Abera Gobena investigated the effect of a family's socio-economic status on students' academic achievement. The research revealed a significant, strong, positive relationship between parents' level of education ($r(170) = 0.73$, $p < 0.05$, two-tailed) and students' academic achievement. This suggests that parental education level can have a positive impact on students' performance. Additionally, the study found that family income could affect academic achievements for both genders. Another study focusing on the effects of family background on children's educational achievement showed that 34.4% of the variation in children's test scores could be attributed to family background (Li and Qiu, 2018). These findings are relevant to our dataset as we also examine how parental education levels influence student grades at school. Based on the literature, it is evident that a family's financial and educational background can significantly influence students' performance in school.

In the report authored by H. Li and Xiong (2018), the focus is on the relationship between test preparation and test performance. Given the high stakes associated with state-mandated accountability assessments, more teachers are using test-preparation strategies to ensure students perform well on state exams. However, it remains uncertain how test preparation impacts students' success in state exams. This study investigates the link between test preparation and students' performance on state tests using the Measure of Effective Teaching (MET) longitudinal dataset.

The study found that students who performed poorly in the first year of tests received additional test preparation in the second year. However, the impact of test preparation on individual students' performance on state exams was minimal and varied. Racial disparities were observed, with Black and Hispanic students receiving more test preparation than White students. For Black students, the effect of test preparation, as assessed by the item 'practicing for the state test,' on state test performance was notably stronger compared to White students. In conclusion, test preparation can influence students' performance, with some cases showing minor impacts, while in others, the impact can be more substantial.

C. Conclusion

From the literature review, it is evident that all the factors included in our dataset have the potential to influence the outcome, which is students' grades. The research papers have provided valuable insights into various methods for data preprocessing and collection. Additionally, they have given us a preliminary understanding of the statistical methods and machine learning algorithm modeling that can be employed to extract meaningful insights from our data. This knowledge serves as a strong foundation for our research and data analysis, enabling us to make informed decisions and draw meaningful conclusions from our dataset.

PROBLEM STATEMENT

A. Why does this problem exist?

In today's modern world, academic success holds significant importance as it paves the way for favorable outcomes in the future. Individuals who excel in their studies often have access to better job opportunities and higher earning potential. Their productivity in the workforce contributes positively to the overall economy, and they are less likely to engage in illegal activities. Therefore, the primary objective of our study is to gain a deeper understanding of the determinants of academic achievement in students. Our research focuses on analyzing data from students in the United States to investigate the interconnections between their grades and various contributing factors. The ultimate aim is to leverage this knowledge to develop predictive models that can forecast students' academic performance. By doing so, we intend to provide valuable insights and support to help students succeed academically, thereby contributing to their future prospects and societal well-being.

B. What is the problem?

A school's overall success is intricately tied to the academic performance of its students. Schools that consistently produce high-achieving students tend to enjoy greater prestige and have the ability to attract top-tier students. This is exemplified by Ivy League institutions, known for their academic excellence. On the flip side, some schools face challenges because a significant portion of their students struggle academically, which can adversely affect the school's reputation. The core problem we are addressing revolves around understanding the factors that influence student performance, ultimately shaping the perception of the school. The challenge here is the sheer multitude of variables that can impact how students perform, making it impossible to consider them all comprehensively. To address this challenge, we have

focused on a select set of key factors that research has identified as strongly correlated with student success, as revealed by our dataset. By homing in on these critical insights, we aim to devise strategies and interventions that can enhance student outcomes and contribute to the overall success of the school.

C. When did this problem happen?

For a significant duration, educational institutions have wrestled with the task of predicting and improving student performance. The primary objective of every school is to provide effective education that empowers students to excel academically. Students are the cornerstone of a school's functioning, and their achievements have a profound impact not only on the school's standing but also on its long-term viability.

D. Who are the stakeholders involved in this problem?

The stakeholders involved in addressing this issue encompass individuals and entities deeply invested in the success and well-being of both the school and its students. These stakeholders include teachers, staff members, principals, parents, community members, school board members, local business leaders, and government entities. Each of these groups plays a vital role in the collective effort to improve and support student performance.

E. Where can you see these problems?

This problem is ubiquitous and transcends school types, affecting both private and public institutions, international schools, and even home-schooling scenarios. In nearly every educational setting, there exists a scenario where some schools outperform others, with variations in student performance having significant repercussions on a school's reputation and overall success. It serves as a critical determinant that distinguishes one school from another, regardless of its type.

Data Acquisition

A. Source of the data

The dataset we used here is Student Performance which we obtained from Kaggle. The URL to the dataset is stated below:

StudentsPerformance.csv (https://www.kaggle.com/spscientist/students-performance-in-exams?select=StudentsPerformance.csv)

Description of data and it's context

We use df.info function in panda to obtain concise information about a Data Frame.

```
<bound method DataFrame.info of      gender race/ethnicity parental level of education         lunch  \
0    female        group B           bachelor's degree      standard
1    female        group C                some college      standard
2    female        group B            master's degree      standard
3      male        group A          associate's degree  free/reduced
4      male        group C                some college      standard
..      ...            ...                         ...           ...
995  female        group E            master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    test preparation course  math score  reading score  writing score
0                      none          72             72             74
1                 completed          69             90             88
2                      none          90             95             93
3                      none          47             57             44
4                      none          76             78             75
..                      ...         ...            ...            ...
995               completed          88             99             95
996                    none          62             55             55
997               completed          59             71             65
998               completed          68             78             77
999                    none          77             86             86

[1000 rows x 8 columns]>
```

Table 1: Overview of dataset

From our original data set it is shown we have 1000 rows and 8 columns.

Our independent variables/ categorical are : gender = male / female, lunch = standard / free, parental level of education = bachelor degree/ some college/ master degree/ associate degree/ high school/ master's degree, test preparation course = complete / not completed, race/ethnicity = Group A/B/C/D/E. Dependent variables / numerical are : math score, reading score writing score

Now lets look into the basic Information of each column using info() function.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

Table 2: Basic information of each column

As described above, our dataset consists of a combination of 3 numerical variables and 5 categorical variables. This dataset is derived from high school student data in the United States. It includes 5 independent variables, namely students' gender, race/ethnicity, parental educational background, test preparation, and lunch (nutrient intake). These independent variables are hypothesized to impact the dependent variables, which are the math, reading, and writing scores used to assess student performance. The aim of our study is to ascertain the degree of correlation between the independent variables and the dependent variables, specifically the student scores. A higher correlation would indicate stronger relationships between these attributes, shedding light on how these factors influence student performance. This analysis will help us better understand the factors contributing to academic success in high school students.

The basic statistic description for numerical/quantitative data

	count	mean	std	min	25%	50%	75%	max
math score	1000.0	66.089	15.163080	0.0	57.00	66.0	77.0	100.0
reading score	1000.0	69.169	14.600192	17.0	59.00	70.0	79.0	100.0
writing score	1000.0	68.054	15.195657	10.0	57.75	69.0	79.0	100.0

Table 3: Basic statistic description for numerical/quantitative data

From the above we can observe students' overall score in each subject.

Now let's look at the first 5 row using the df.head() function

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Table 4: First five row of dataset

Research Methodology

A. Research Questions

1. How can the application of advanced data mining and machine learning techniques in research methodology enhance the accuracy of predicting students' examination performance?
2. What are the underlying relationships and correlations between various factors, such as parental education level, gender, test preparation, and student performance, within the dataset?
3. How does students' race/ethnicity relate to their test scores, and are there any disparities among different ethnic groups in terms of academic performance?
4. To what extent do students' test scores correlate with their overall grades, and can test performance reliably predict final course grades?
5. How do the size and composition of the training and testing sets impact the accuracy and generalization capabilities of machine learning models in predicting student performance?
6. Among all student attributes, which one(s) exhibit the strongest correlation with academic performance across all three subjects?

B. Search Documentation and Selection Sources:

i. **Keywords**: Machine Learning, Data Mining, Correlation, Testing set, Training set, Academic Performance, Predictive Models

ii. **Search Documentation and Selection Sources**

A reliable and well-planned information source and search strategy are essential for achieving the study objectives and obtaining meaningful results. To address our research questions comprehensively and systematically, we conducted a thorough search across three online databases: ScienceDirect, IEEE Xplore, and Google Scholar. We employed a combination of carefully selected keywords based on the work of B. Kitchenham, organized as follows: "estimation OR prediction OR forecasting" AND "student academic performance OR student

performance" AND "machine learning OR data mining OR educational data mining" AND "approaches OR methods OR applications OR 2015 technique."

In the initial phase, our team screened the titles and abstracts of potential studies to ensure their alignment with our established inclusion and exclusion criteria. This preliminary screening helped us identify papers with potential relevance. Subsequently, we conducted a more detailed assessment by thoroughly reviewing the full text of these shortlisted studies. In cases where a study's relevance appeared unclear, we conducted an exhaustive full-text review to determine its suitability.

In situations where there was a difference of opinion among team members regarding the selection of a particular study, we reached a consensus through collaborative discussions, ensuring that every choice was well-founded and agreed upon collectively. Throughout this process, we relied on EndNote X9 software as our primary tool for gathering and organizing references. This tool also proved invaluable in eliminating duplicate entries, ensuring the uniqueness and comprehensiveness of our final selection.

A. Inclusion and Exclusion Criteria

Inclusion

- Studies related to Student’s Performance Prediction.
- Research Papers that were accepted and published in a blind peer-reviewed Journals or conferences.
- Papers that were from 2015 to 2023 era.
- Papers that were in the English language.

Exclusion

- Studies other than Student’s Performance Prediction using ML.
- Papers which had not conducted experiments or had validation of proposed methods.
- Short papers, Editorials, Business Posters, Patents, already conducted Reviews, Technical Reports, Wikipedia Articles, and extended papers of already reviewed papers.

Presentation, Visualization and Quantification of the Data and Image

We have created a data frame with column whether NULL values is there or not with their respective data types to find how many missing values exist in each column.

	isNullExist	NullSum	type
gender	False	0	object
race/ethnicity	False	0	object
parental level of education	False	0	object
lunch	False	0	object
test preparation course	False	0	object
math score	False	0	int64
reading score	False	0	int64
writing score	False	0	int64

Table 5: Null Value Check and Data Type Overview

It is shown that the data has no null value.

A. Frequency distribution of categorical data for parental level of education column

We use **matplotlib** library to create a bar graph illustrating the frequency distribution of parental education levels. We begin by importing the necessary libraries, including **matplotlib** and **NumPy**. Subsequently, the **plot.bar()** method is employed on the Data Frame to generate the bar chart, with the 'parental level of education' column serving as the source of data.

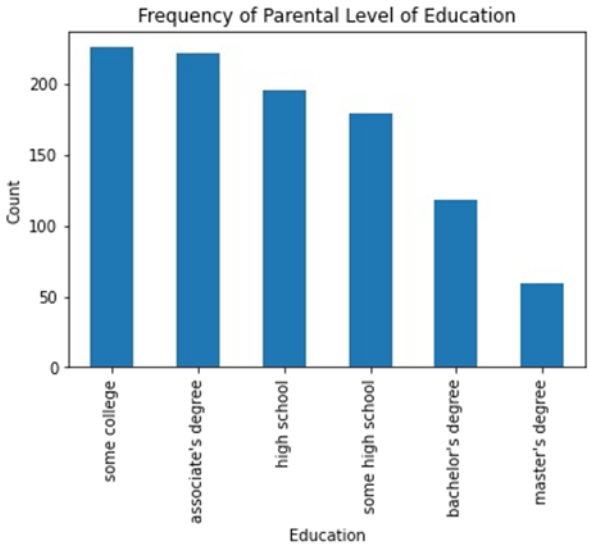

Figure 2: Frequency distribution of parental level of education

From the categorical data statistical analysis, it's shown that most parents have educational background up until college and associate degree.

Let's visualize and understand the distribution of lunch types in our dataset.

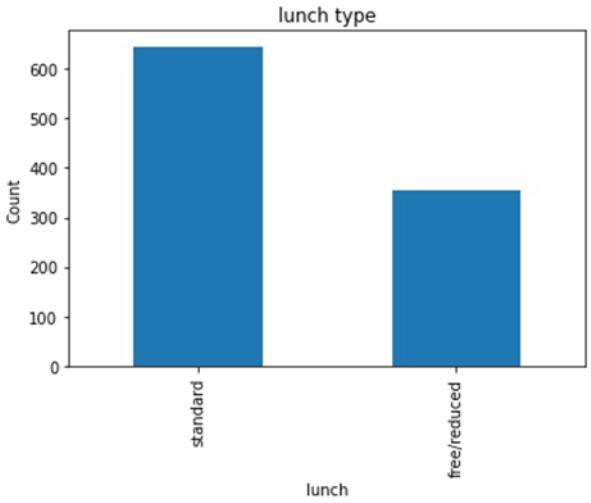

Figure 3: Frequency distribution of lunch type

The "Standard" lunch category has approximately 64.5% of students, while the "Free/Reduced" lunch category has about 35.5%. It suggests that a larger percentage of students fall into the "Standard" lunch category, which could be considered as the majority group, while a smaller percentage falls into the "Free/Reduced" lunch category.

Let's visualize and understand the distribution of gender in our dataset.

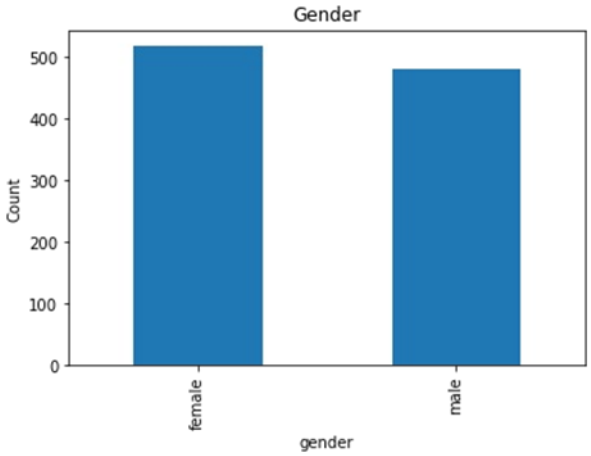

Figure 4: Frequency distribution of gender

The graph provided indicates the gender distribution in the dataset, with 518 female students and 482 male students. Female students are approximately 51.8% while male students are approximately 48.2%.

Let's visualize and understand the distribution distribution of race/ethnicity

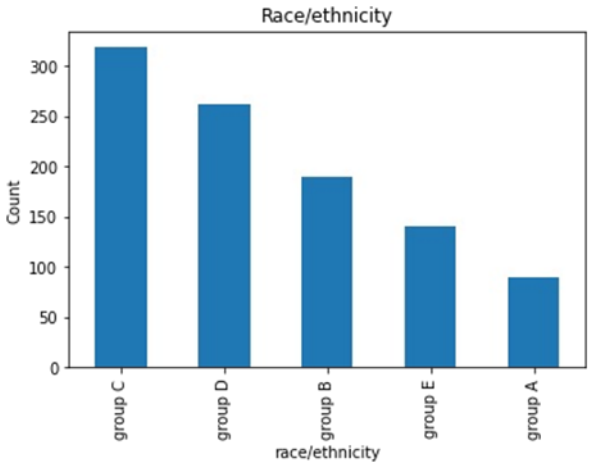

Figure 5: Frequency distribution of race/ethnicity

From the above graph, among the race/ethnicity groups, "Group C" is the most prominent, with 319 students, representing the highest count. On the other end, "Group A" is the least represented, with 89 students, showing the lowest count. To put these numbers in perspective, "Group C" accounts for approximately 31.9% of the total students, followed by "Group D" at around 26.2%, "Group B" with about 19%, "Group E" with approximately 14%, and "Group A" with about 8.9%.

Let's visualize and understand the distribution of Test Preparation Course

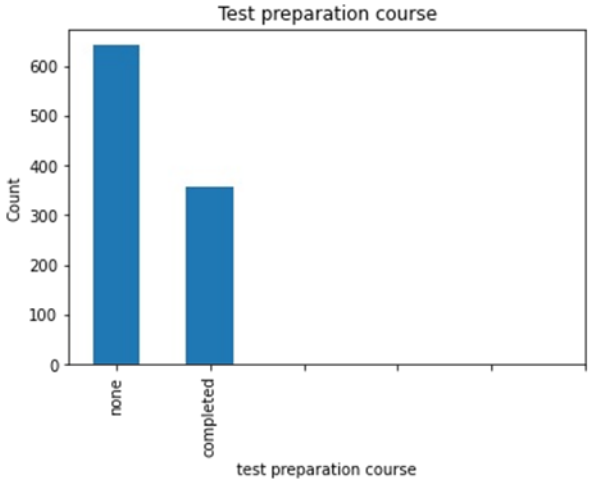

Figure 6: Frequency distribution of Test preparation course

From the chart, we see that only 35.8% of the student attended/completed the test preparation course.

Let's visualize and understand the distribution for quantitative data

Comparison of all other attributes with respect to Math Marks

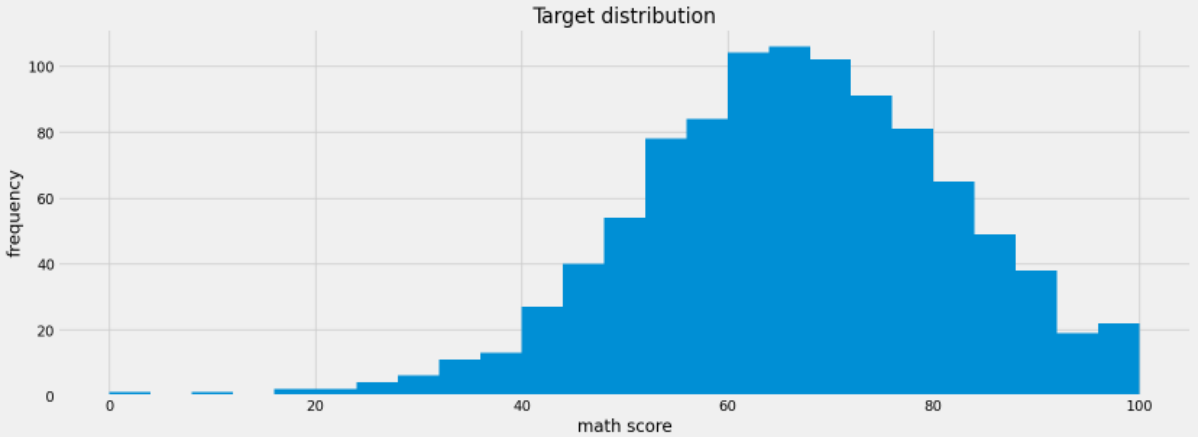

Figure 7: Frequency distribution of math scores

This plot provides a histogram representation of the distribution of math scores. The distribution is approximately bell-shaped (normal distribution), but there seems to be a slight skewness to the left. Most students have scores concentrated around the 60-80 range. Fewer students have scores at the extreme low and high ends.

Correlation of Math Scores with Reading and Writing Scores by Gender

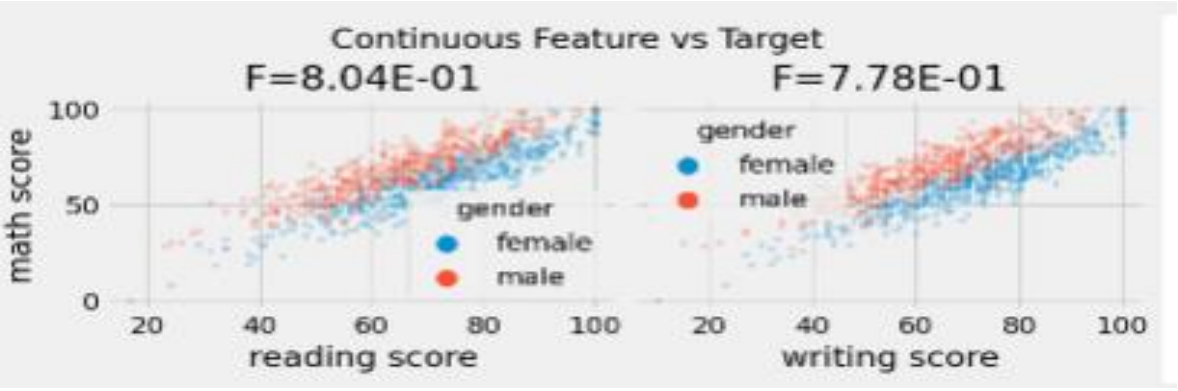

Figure 8: Correlation of Math Scores with Reading and Writing Scores by Gender

These scatter plots show the relationship between continuous features (reading and writing scores) and the target (math score).

F-Statistics represents the overall significance of the linear relationship between the continuous feature and the target. A larger F-value suggests a stronger relationship. There's a strong linear relationship (as indicated by the F-statistic value) between reading scores and math scores. As reading scores increase, math scores generally increase as well. Similarly, there's a strong linear relationship between writing scores and math scores. Higher writing scores are associated with higher math scores.

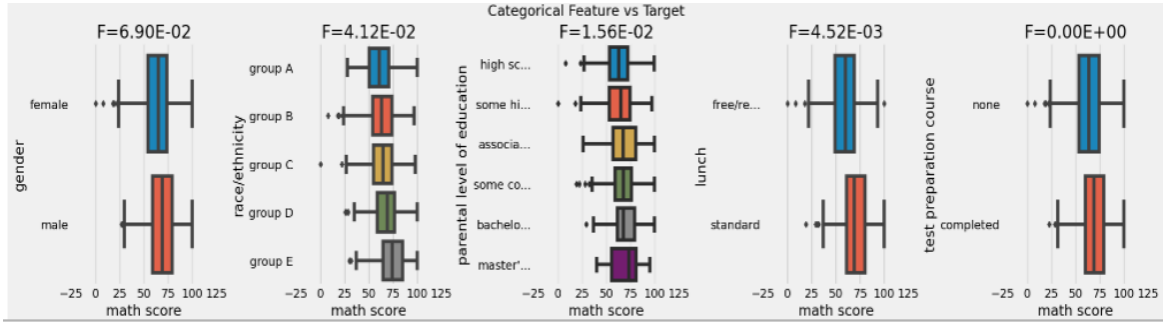

Figure 9: Impact of Categorical Features on Math Scores Distribution

These are box plots showing the distribution of math scores across different levels of categorical features. F-Statistics represents the significance of the difference in math scores across the categories of the feature. A larger F-value suggests that the feature has a more significant impact on math scores.

We can say from the figure, male students tend to have a slightly higher median math score compared to female students. Different racial/ethnic groups have varying distributions of math scores. Some groups have a higher median math score than others. There's variation in math scores based on the parental level of education. Students whose parents have higher educational qualifications might tend to have different math score distributions. The type of lunch (free/reduced or standard) also seems to influence math scores to some extent. The box plot shows the difference in math scores between students who completed a test preparation course and those who didn't. The F-statistic value is zero, indicating that there's no significant difference in math scores based on this feature.

Comparison of all other attributes with respect to Reading Marks

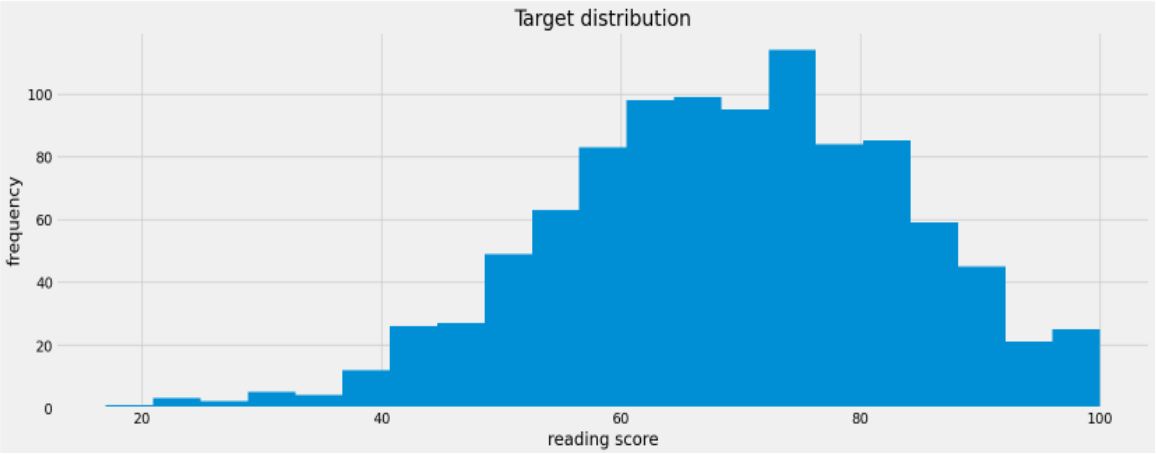

Figure 10: Frequency distribution of reading scores

The distribution appears approximately bell-shaped (similar to a normal distribution), though there's a slight right-skew. The majority of students scored between 60 and 80. Few students achieved very high or very low scores.

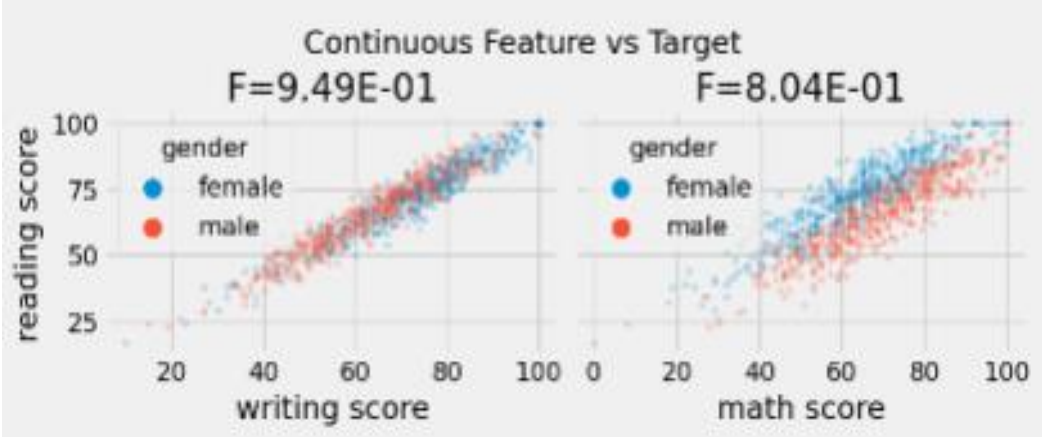

Figure 11: Correlation of Reading Scores with Math and Writing Scores by Gender

There's a strong linear association between writing scores and reading scores. Students with higher writing scores generally have higher reading scores. A significant linear relationship exists between math scores and reading scores. As math scores increase, reading scores tend to increase as well.

Figure 12: Impact of Categorical Features on Reading Scores Distribution

From the figure we can see females tend to have a slightly higher median reading score than males. The difference, as indicated by the F-statistic, is significant but not overwhelmingly so. Different racial/ethnic groups showcase varying distributions of reading scores. For instance, group C and group D have higher median scores compared to groups A and B. The distribution of reading scores seems to differ based on the parental education level. For instance, students with parents holding master's degrees exhibit higher median reading scores than those with parents who've only completed high school. There's a discernible difference in reading scores based on lunch type. Students with standard lunch generally have higher median scores than those with free/reduced lunch. The box plot contrasts reading scores between students who've completed a test prep course and those who haven't. Interestingly, the F-statistic value is zero, implying that there's no statistically significant difference in reading scores based on this factor.

B. Feature Extraction

We add attributes to analyse 'pass_maths' 'pass_reading' 'pass writing' 'total score' 'percentage' 'grade'. Our target variable will be 'grade' to predict student performance.

'pass math' = students who passed the maths test

'pass reading' = student who passed the reading test

'pass writing' = student who passed the writing test

'total score' = the total score student obtain in maths, reading, and writing test

'percentage' = the average score of students in the 3 subjects

'status' = to classify students who pass the 3 exams overall

'grade' = to classify the grades student obtained

Analysis on students who 'pass_math'

We set the pass mark at 40 for the three subjects. Creating a new column pass_math, to identify whether the students pass or fail. The "pass_math" column distinguishes students who have passed (labelled as "Pass") from those who have failed (labelled as "Fail") in the mathematics section of the exam. This classification is based on whether a student's math score is above or below the specified pass marks threshold.

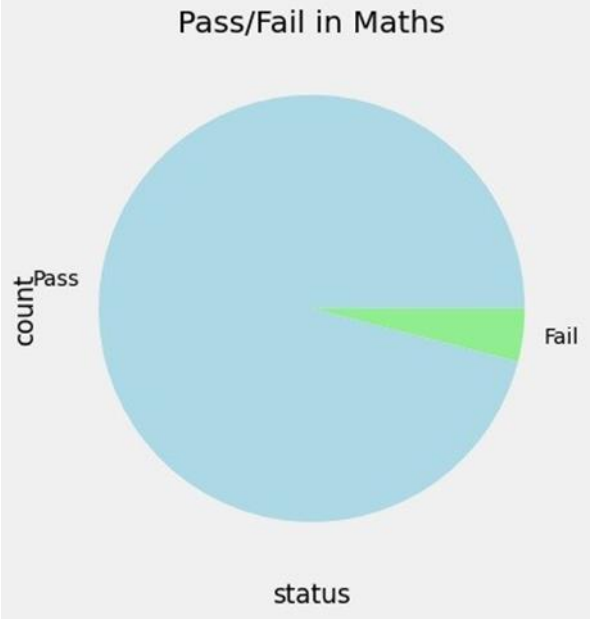

Figure 13: Pass/Fail in Maths

The blue segment represents the number of students who passed, while the green segment shows those who didn't meet the pass marks criteria. We can see from the graph only a very small portion of students have failed in Math

Analysis on students who 'pass_reading'

Creating a new column pass_reading, to identify whether the students pass or fail. The "pass_reading" column distinguishes students who have passed (labelled as "Pass") from those who have failed (labelled as "Fail") in the reading section of the exam. This classification is based on whether a student's reading score is above or below the specified pass marks threshold.

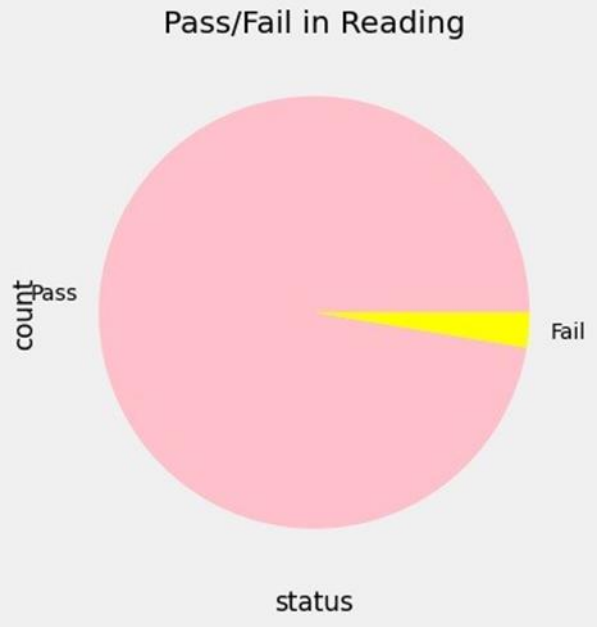

Figure 14: Pass/Fail in Reading

The pink segment represents the number of students who passed, while the yellow segment shows those who didn't meet the pass marks criteria. We can see from the graph only a very small portion of students have failed in Reading

Analysis on students who 'pass_writing'

Creating a new column pass_writing, to identify whether the students pass or fail. The "pass_writing" column distinguishes students who have passed (labelled as "Pass") from those who have failed (labelled as "Fail") in the writing section of the exam. This classification is based on whether a student's writing score is above or below the specified pass marks threshold.

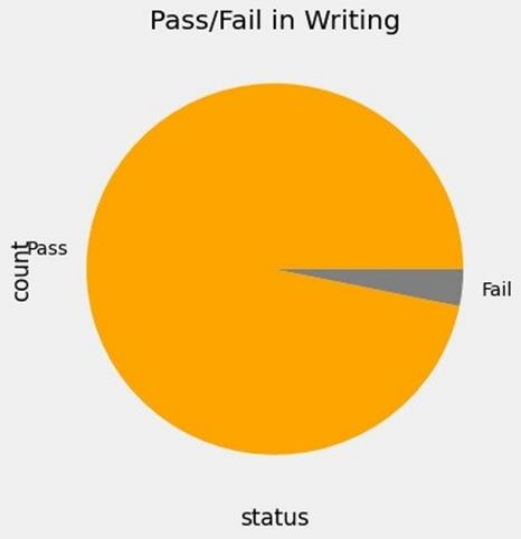

Figure 15: Pass/Fail in writing

The orange segment represents the number of students who passed, while the Gray segment shows those who didn't meet the pass marks criteria. We can see from the graph only a very small portion of students have failed in Writing.

Analysis on students' total score 'total_score'

First, we compute the 'total_score' by adding the scores from the three subjects: math, reading, and writing. We use a distribution plot, represented in magenta color, to visualize the distribution of total scores across all students in the dataset. The distribution plot provides insights into the central tendency and spread of the data. It enables us to identify trends and patterns in student achievement across the dataset.

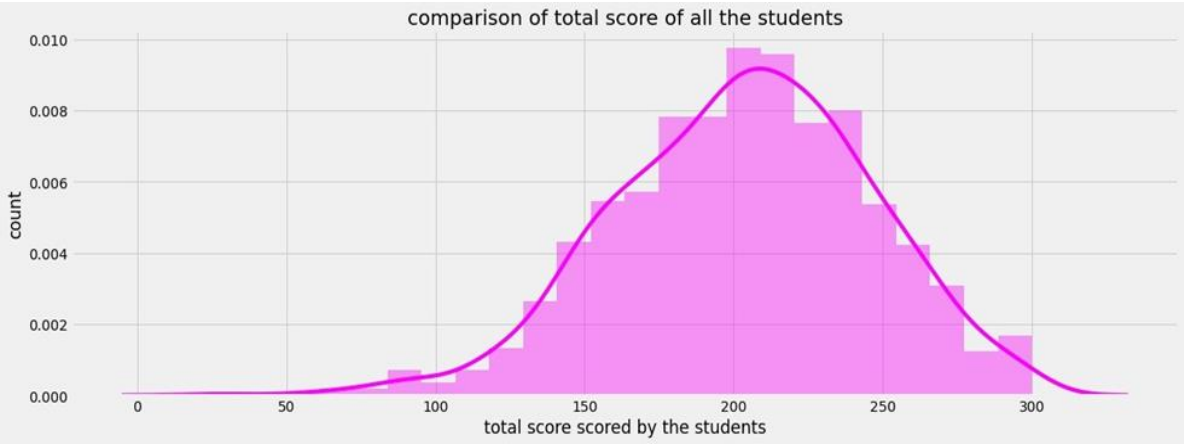

Figure 16: Comparison of total score of all the students

From the plot, it's apparent that most students achieve a total score around the 150 to 250 range, with the peak or the mode of the distribution lying slightly above 200. This peak indicates that the majority of the students have their cumulative scores around this value. The shape of the distribution is somewhat bell-

shaped or normal, suggesting a typical distribution pattern where most students score around the average, with fewer students achieving either very high or very low scores.

Analysis on student average score as 'percentage'

In this analysis, we are calculating and visualizing the percentage scores achieved by each student in the dataset. We start by computing the percentage score for each student by dividing their total score (the sum of math, reading, and writing scores) by 3. The **math.ceil()** function is then applied to round the percentages up to the nearest whole number. To visualize the distribution of the calculated percentage scores, we create a distribution plot. The plot is in the colour orange, which makes it visually distinct.

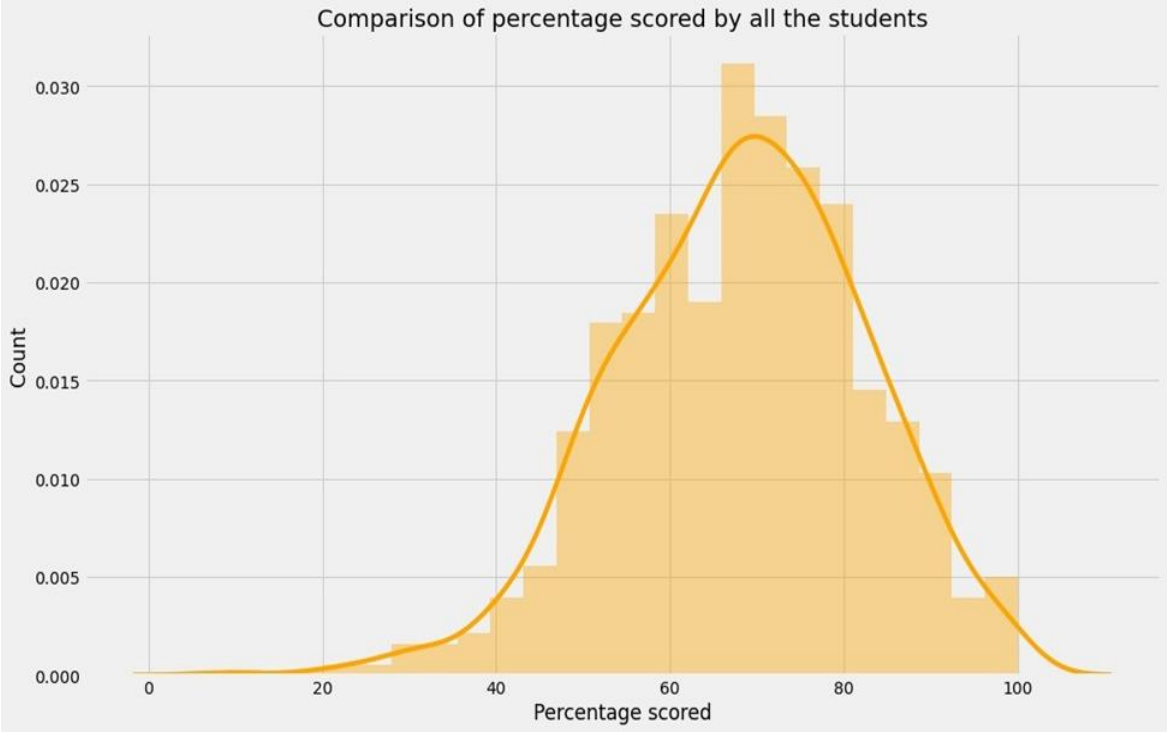

Figure 17: Comparison of percentage scored by all the students

Plot provides insights into the spread and central tendencies of the data. The majority of students achieved respectable scores, with the densest concentration around the 80% range, and fewer students scoring below 40%.

To analyze which student failed overall the 3 subject we created a status column 'status'

In this analysis, we are determining the overall results of students, specifically whether they have passed or failed based on their performance in different subjects. We begin by assessing each student's status, classifying them as either "Fail" or "Pass." This classification is determined based on the students' individual

performance in math, reading, and writing. If a student has failed in any of these subjects, they are marked as an overall "Fail"; otherwise, they are marked as "Pass." This is done using a lambda function that checks the pass/fail status in each subject and aggregates it to determine the overall status.

To visually represent the distribution of overall results, we create a pie chart. The chart uses grey for "Fail" and crimson for "Pass." Next, to classify the grades we decided to create a column 'grade' to see which category did the student obtain by using their percentage.

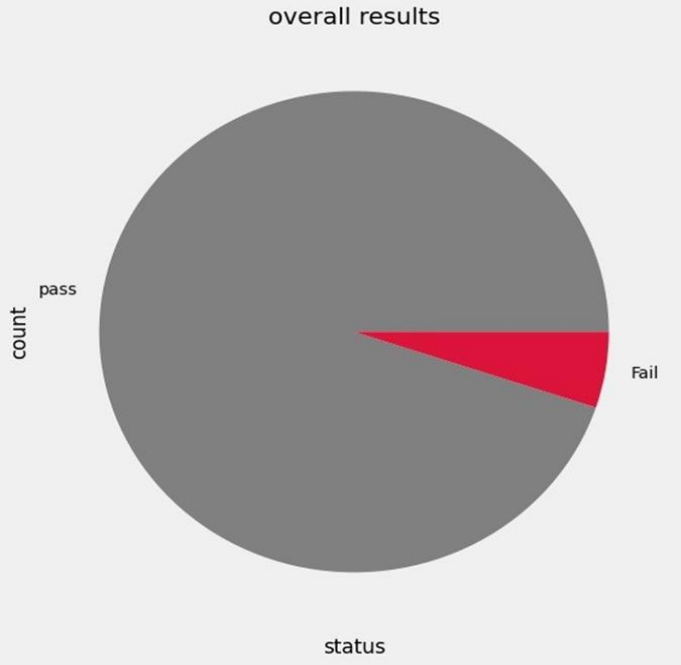

Figure 18: Overall Student Performance Distribution

A significant majority of the students have passed, as indicated by the large grey portion of the pie chart. This suggests that most students in the dataset met the minimum criteria or threshold set for passing in their respective subjects. The smaller crimson section represents the students who did not meet the criteria and thus failed.

0 - 40 marks : grade E

41 - 60 marks : grade D

60 - 70 marks : grade C

70 - 80 marks : grade B

80 - 90 marks : grade A

90 - 100 marks : grade A*

A function named **getgrade** is defined to assign grades to students based on their percentage and status (Pass or Fail). This function takes two parameters, **percentage** and **status**. It uses these values to determine the appropriate grade for a student. The grading scale includes 'A*', 'A', 'B', 'C', 'D', and 'E'. Students who

have failed are assigned an 'E' grade. The Data Frame is updated by applying the **getgrade** function to each row. The result is a new column, 'grades,' which contains the assigned grades for each student. Next we use a pie to visualize the distribution of grades among the students.

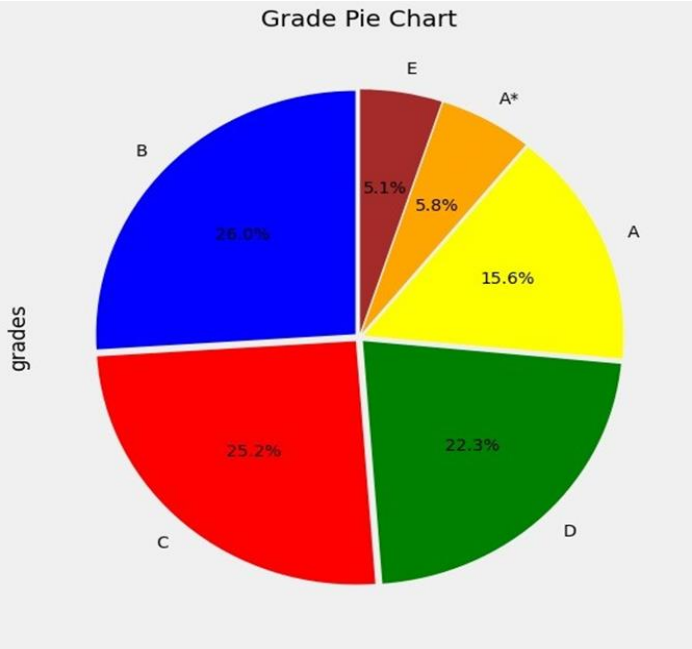

Figure 19: Distribution of Student Grades in percentage

The resulting pie chart, titled "Grade Pie Chart," provides a visual representation of the distribution of grades among the students. The majority of students obtained a grade of B or C (51.2%), whereas A* and E grades are the lowest, which are 5.8% and 5.1% respectively.

Data Processing

A. Data Cleaning: Upon examining the dataset, we observed that it initially comprised 1000 rows and 15 columns, as indicated by the shape of the data. Subsequently, we conducted a duplicate value check using the "drop_duplicates" function, and the dataset's dimensions remained consistent at 1000 rows and 15 columns. This outcome signifies that there were no duplicate records present within the dataset.

B. Check Null Values: In our data analysis, we conducted an examination of the dataset to identify the data types of each column and evaluate the presence of null values using the **isna()** function. The results from this assessment indicate that there are no null values within the dataset.

	isNullExist	NullSum	type
gender	False	0	object
race/ethnicity	False	0	object
parental level of education	False	0	object
lunch	False	0	object
test preparation course	False	0	object
math score	False	0	int64
reading score	False	0	int64
writing score	False	0	int64
pass_math	False	0	object
pass_reading	False	0	object
pass_writing	False	0	object
total_score	False	0	int64
percentage	False	0	float64
status	False	0	object
grades	False	0	object

Table 6: Null Value Check and Data Type Overview

It is shown that the data has no null value.

C. Removing the outliers: We will only check the outlier for 'math score' 'reading score' 'writing score' as these 3 are the main attributes used to determine other attributes. We used various functions to create and customize a grid of box plots for different subjects' scores such as plt.subplots() to set up the subplot layout, sns.boxplot() to generate the box plots, and remove() to eliminate an unused subplot. Additionally, we use set_xlabel() to label the x-axes appropriately and ensures proper subplot arrangement with plt.tight_layout(). Finally, we display the grid of box plots with plt.show(), providing a visual representation of the score distributions in the dataset.

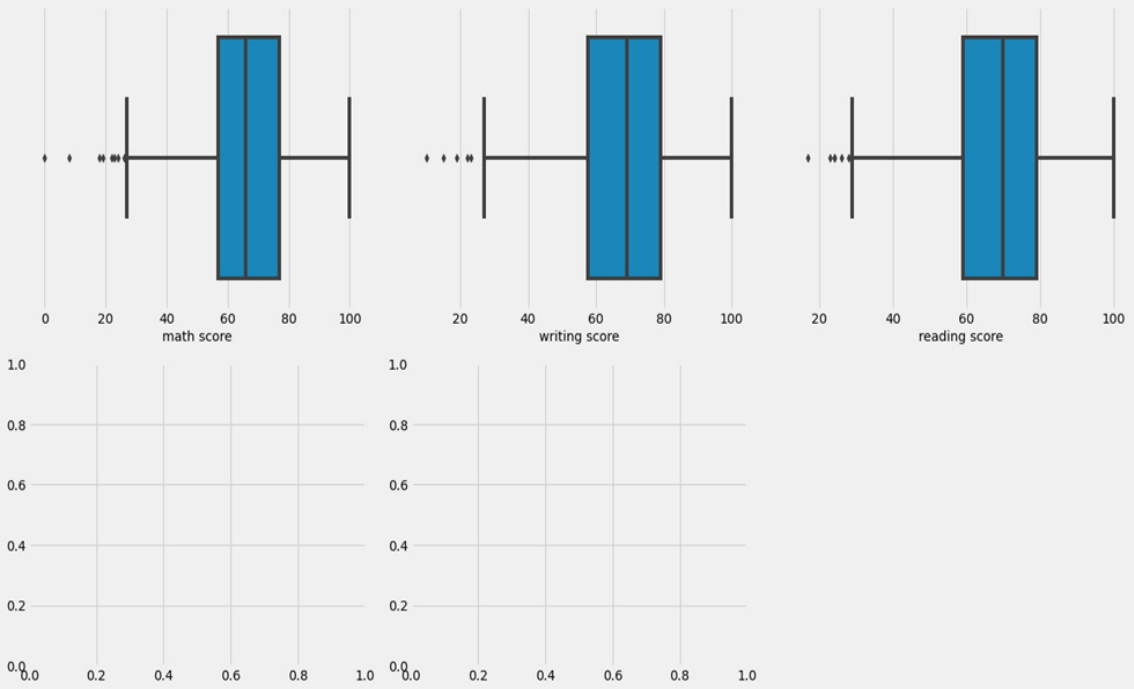

Figure 20: Score Distribution: Box Plots for Math, Writing, and Reading

The above acceptable range will be at (Q1 - 1.5 x IQR to Q3 + 1.5 x IQR), Q1 represents the lower quartile and Q3 is the higher quartile. Whereas IQR is the interquartile range of the attribute. The outliers are the ones outside the range and will then be smoothed out by minimum and maximum acceptable values.

However, in our case, we chose not to remove or smoothen them out as each score is distinct and important in our data analysis. It represents some students' ability regardless of higher or lower scores and it is very informative on the data collection process and subject area. Therefore, no further smoothing is done.

D. Data transformation using label encoder: Data transformation is then performed using label encoder to encode the independent variables into Machine Learning (ML) readable format. We employ the LabelEncoder from the scikit-learn library. It encodes categorical variables like "test preparation course," "lunch," "parental level of education," "gender," "pass_math," "pass_reading," "pass_writing," "status," and "grades" into numerical values. For the "race/ethnicity" column, it manually maps specific categories like "group A," "group B," etc., to numerical values, transforming them into discrete integers. Similarly, for the "grades" column, it maps letter grades ("A*", "A," "B," etc.) to corresponding numeric values to represent the order or ranking of grades.

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	pass_math	pass_reading	pass_writing	total_score
0	0	2	1	1	1	72	72	74	1	1	1	2
1	0	3	4	1	0	69	90	88	1	1	1	2
2	0	2	3	1	1	90	95	93	1	1	1	2
3	1	1	0	0	1	47	57	44	1	1	1	1
4	1	3	4	1	1	76	78	75	1	1	1	2

Table 7: Encoded Data Overview: Student Performance Metrics

By applying label encoding, the code prepares the dataset for machine learning by ensuring that the algorithms can work with the data in a numerical format, ultimately facilitating the analysis and modeling of the dataset.

8. Exploratory Data Analysis

A. Basic Statistic of our data

1. Checking the skewness of the data

The scores for each subject range from 0 to 100, and they're displayed on the x-axis. The y-axis represents the density of scores. This is different from a histogram in that it gives a continuous and smoothed representation of the distribution of scores.

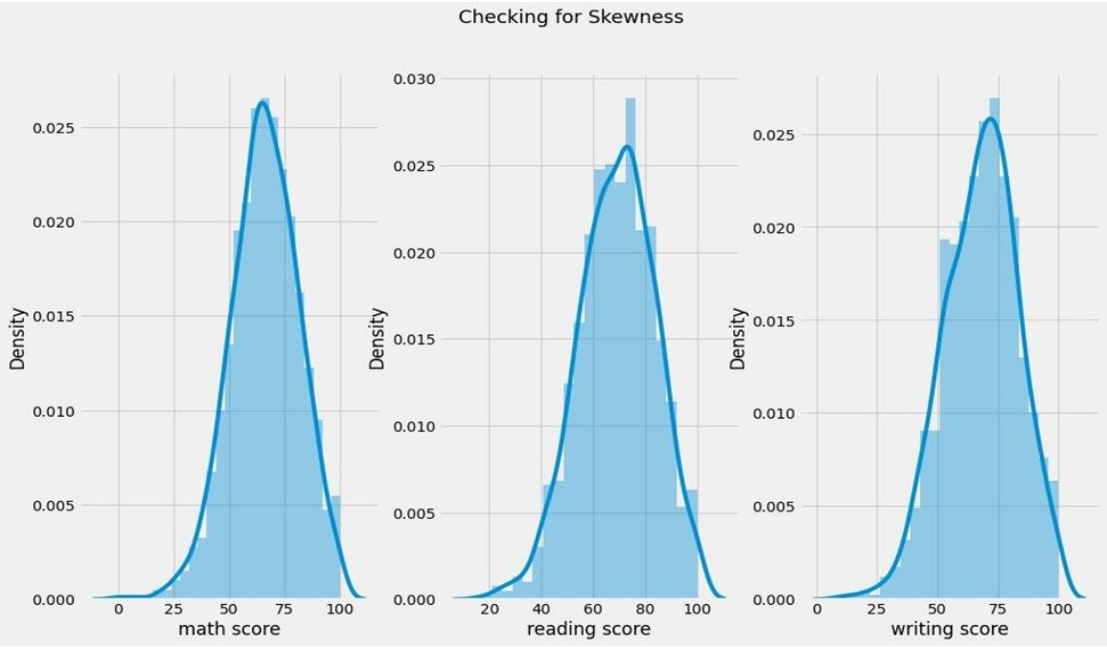

Figure 21: Distribution of Student Scores: Assessing Skewness in Math, Reading, and Writing

The presented density plots provide a visual representation of the distribution of student scores in three subjects: Math, Reading, and Writing. A crucial aspect assessed in these plots is the skewness of the data.

Skewness refers to the degree of asymmetry observed in the distribution around its mean. In an ideally symmetrical distribution, the left and right sides of the density curve would be mirror images of each other, resembling the shape of a bell.

Upon examination of the plots, the distributions for all three subjects appear to be relatively symmetrical, indicating that scores are roughly normally distributed around their respective means. However, subtle deviations can be observed. For instance, the Math and Writing scores display slight inclinations towards the higher end, suggesting a marginal right-skewness. This means that there might be a slightly higher concentration of students achieving scores above the mean in these subjects. Conversely, the Reading scores manifest a near-perfect symmetrical distribution, suggesting minimal skewness in this dataset.

2. Now let's explore the mean score of math, reading and writing against race

We use the groupby function to group data by 'race/ethnicity' and the agg function to compute statistics: mean math and reading scores for each group, along with the mean and count of writing scores. The output provides insights into the academic performance of different ethnic groups within the dataset.

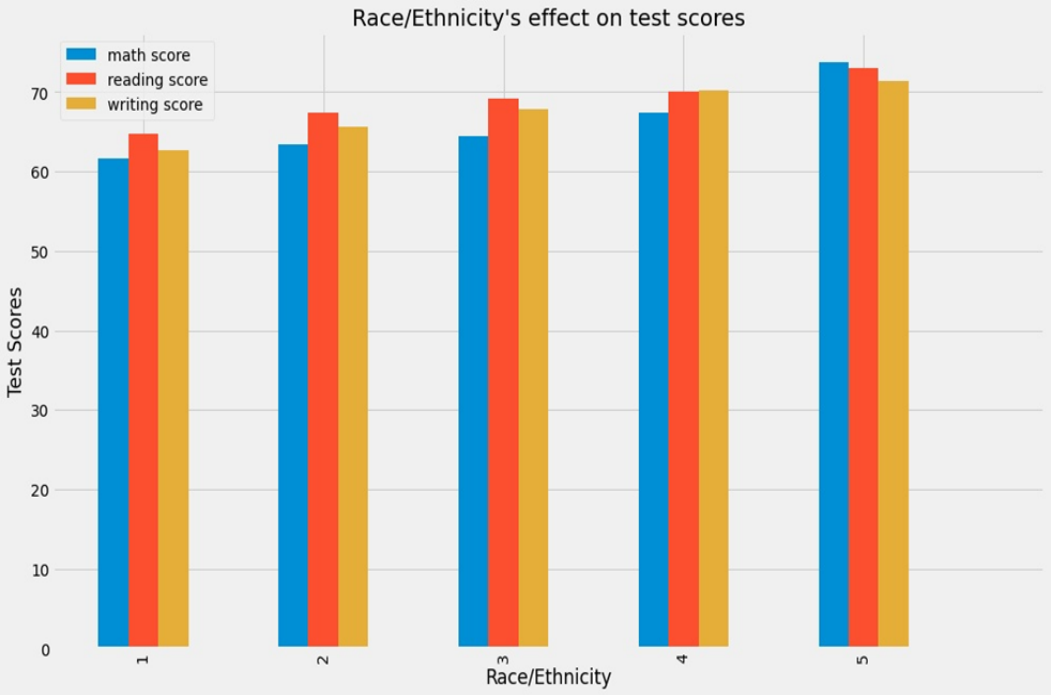

Figure 22: Race/Ethnicity's effect on test scores

From the above bar chart, we are able to observe that Group E/ category 5, scores higher on average for all three tests. On the other hand, Group A/ category 1 scores the lowest on average for all tests.

3. Observe the gender, test preparation course with math score

We group the dataset by gender and test preparation course, calculating mean math scores and counts of students for each combination. Then create a bar plot showing the mean math scores, allowing for a visual comparison of how gender and test preparation courses impact students' math performance, with titles and labels for clarity.

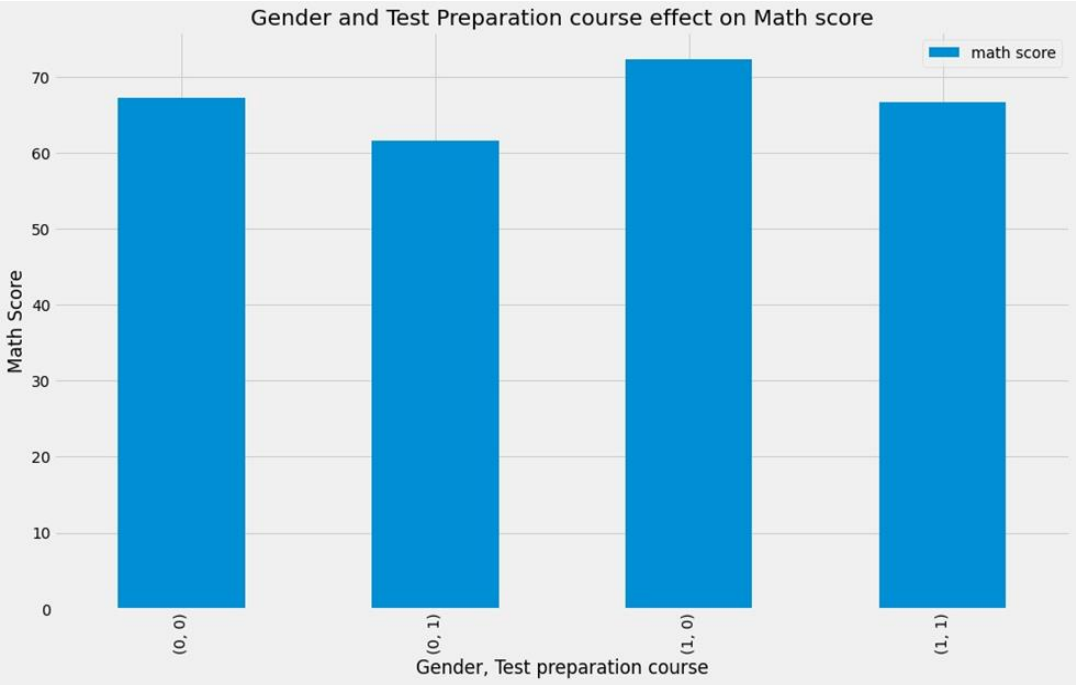

Figure 23: Gender and Test preparation course effect on math score

From the graph, we can see that both female and male who completed the test preparation course scored higher in their math exam, which are bars (0,0) and (1,0) respectively, as compared to those who did not take the test preparation course. It can also be seen that male [Bars (1,0), (1,1)] score higher than females [Bars (0,0), (0,1)] in the math test, regardless of whether they took the test preparation course.

4. Let's observe how lunch quality intake affects student performance by gender. As we know, standard lunch has more nutrient compared to free lunch

We group the dataset by lunch type and gender, calculating the median math, writing, and reading scores for each combination. It then creates a bar plot visualizing the median test scores, allowing for a comparison of how lunch type and gender influence students' performance on these tests. The plot is titled "Lunch and Gender effect on test scores" and includes appropriate labels for clarity.

Figure 24: Lunch and gender effect on test scores

Students who have standard lunch can score higher on their tests, as depicted in the graph by bars (1,0) and (1,1). Female students scored better than male students in both the standard lunch and free/reduced lunch categories for reading and writing tests. Male students scored better than the female students for the math test whether they had standard or free/reduced lunch.

This suggests that standard lunch which has higher nutrition value as compared to the free/reduced lunch, is able to aid students in getting higher test scores. It can also be inferred that female students do better on reading and writing tests, whereas male students do better on the math test.

5. Let's see how parental education background can affect student performance statistics.

We group the dataset by parental level of education, calculating both the mean and count of the 'percentage' variable for each education level. Then we create a bar plot to visualize the mean test score percentages for each parental education level, aiming to understand how parental education impacts students' performance. The plot is titled "Parental level of education's effect on test score percentage" and includes appropriate labels for clarity.

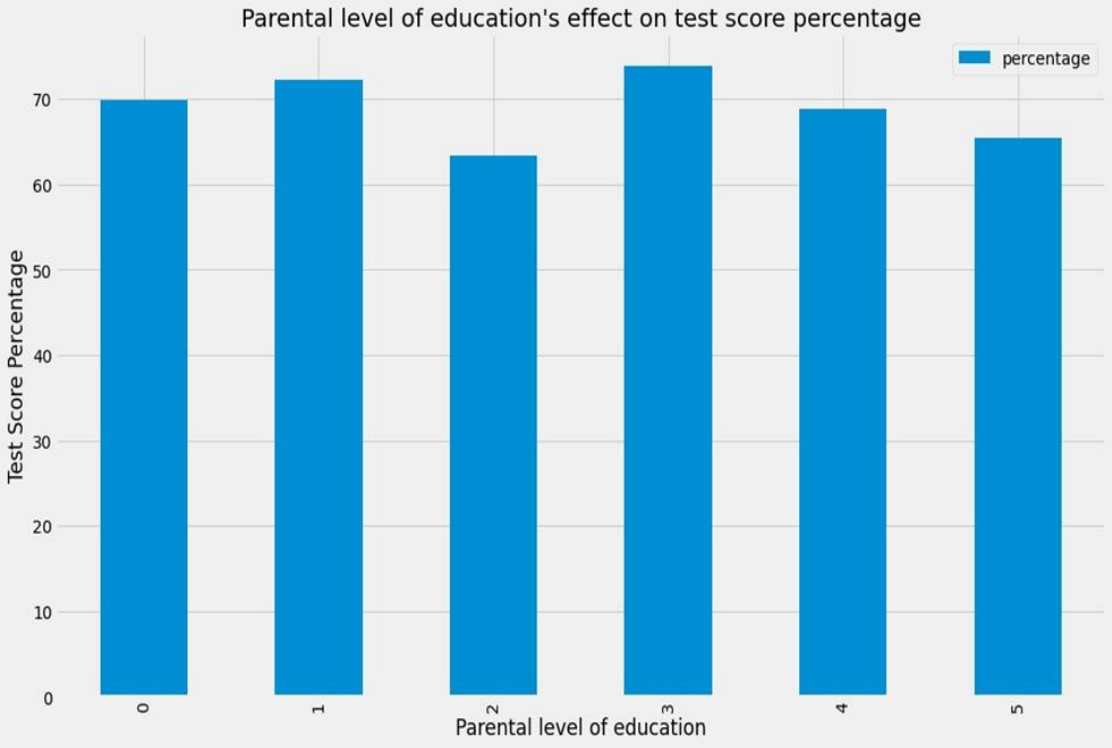

Figure 25: Parental level of education's effect on test score percentage

From the above statistical analysis, we can see that students' parent level of education who is in category 3 which is master's degree score the highest in exams based on the average percentage which is 73.9. It is followed by category 2 (bachelor's degree) at 72.2, category 1 (associate degree) at 69.9, and the rest of the categories in descending order are category 4 (some college), category 5 (some high school), and category 2 (high school).

This indicates that there is a relationship between parental education background and student performance on tests. The higher the level of parental education, the better the students' performance on tests.

Next, we make a score heatmap to determine the relationship between the tests results

We create a heatmap using Seaborn to visualize the correlations (relationships) between different variables. The heatmap is annotated with correlation values and displays how strongly these test scores are related to each other. The title of the heatmap is "Test Scores Heatmap," and it includes annotations and grid lines for clarity.

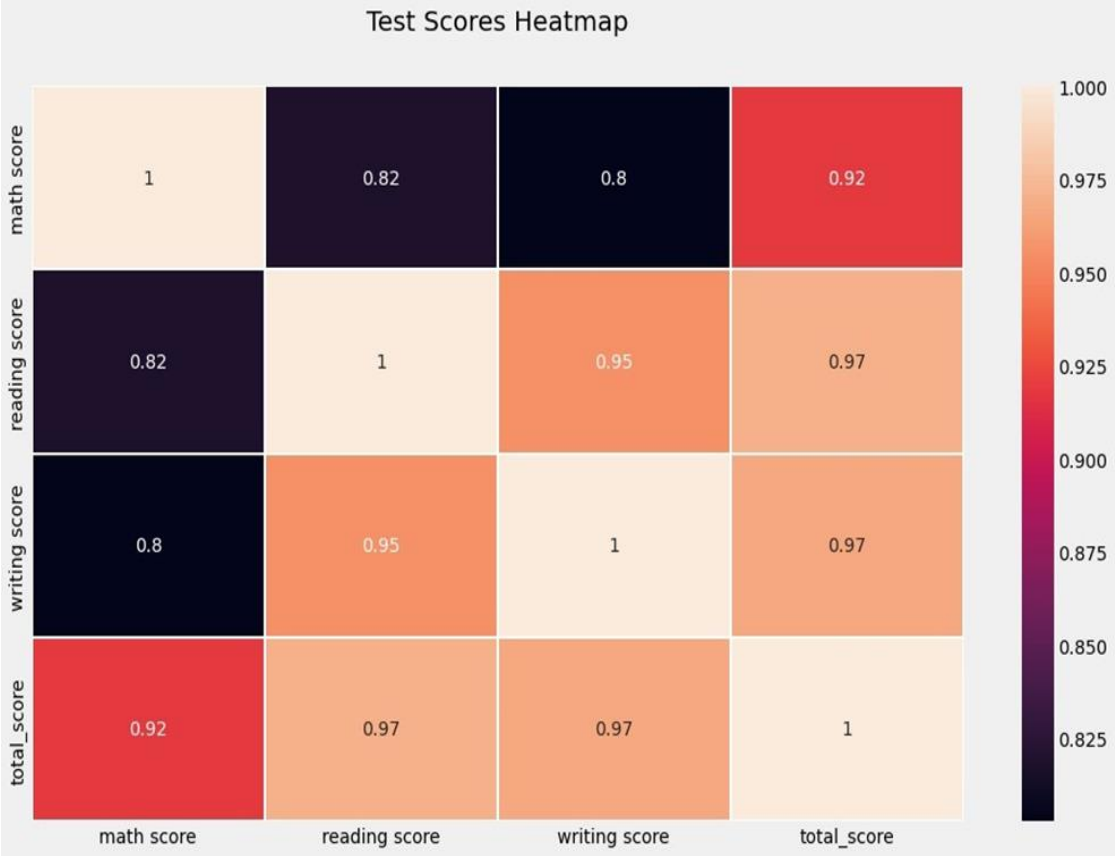

Figure 26: Test scores heatmap

The correlation between all the test scores is high and close to each other. This could imply that generally, students who do well in one subject will also do well in the other subjects, leading to a higher total score. For example the correlation between Math and Reading scores is 0.82, which is relatively high. This indicates that students who perform well in Math tend to also perform well in Reading and vice versa. With a correlation of 0.97, the Writing score also has a profound impact on the Total Score. Students who excel in Writing will, in all likelihood, have a higher Total Score.

B. Feature Extraction

We came up with new features in our data set which are 'pass_math' 'pass_reading' 'pass_writing' 'total_score' 'percentage' 'status' 'grade' previously. These features are used to aid us in predicting the student performance.

#	Column	Non-Null Count	Dtype
0	gender	1000 non-null	int32
1	race/ethnicity	1000 non-null	int64
2	parental level of education	1000 non-null	int32
3	lunch	1000 non-null	int32
4	test preparation course	1000 non-null	int32
5	math score	1000 non-null	int64
6	reading score	1000 non-null	int64
7	writing score	1000 non-null	int64
8	pass_math	1000 non-null	int32
9	pass_reading	1000 non-null	int32
10	pass_writing	1000 non-null	int32
11	total_score	1000 non-null	int64
12	percentage	1000 non-null	float64
13	status	1000 non-null	int32
14	grades	1000 non-null	int64

Table 8: New features summary table

C. Feature Selection

By inspecting the properties of each attribute. As observed, the attributes 'math score', 'reading score', 'writing score', 'total_score' and 'grades' are quantitative attributes made up of discrete integer values, the attribute 'percentage' is a quantitative attribute made up of continuous float values, the attributes 'gender', 'test preparation course', 'pass_math', 'pass_reading', 'pass_writing' and 'lunch' are categorical attributes made up of binary values, and the attributes 'parental level of education', 'race/ethnicity' are categorical attributes made up of nominal values.

Next, we will be using the univariate selection to select top 10 features to be used in the modelling process using their significance of each feature. After selecting the non-significant features will be dropped. Univariate feature selection is a technique for selecting the most important features (independent variables) from a dataset based on their individual relationships with a target variable (dependent variable) without considering interactions between features. We use the chi-squared statistical test as a scoring function to evaluate the relationship between each feature and the target variable independently. Then, you select the top features based on their scores. This method is a type of univariate feature selection because it assesses each feature's relevance to the target variable separately, without considering the combined effects of multiple features.

	Feature	Score
11	total score	8373.930025
7	writing score	2925.245597
5	math score	2781.712049
12	percentage	2779.708154
6	reading score	2683.478388
13	status	51.000000
3	lunch	32.207757
8	pass math	31.013072
4	test preparation course	23.583950
10	pass writing	19.684330

Table 9: Top 10 features in the dataset

Above are the top 10 features selected. Next, we create a new data frame to put the 10 features together with its target 'grade'

	total_score	writing score	math score	percentage	reading score	status	lunch	pass_math	test preparation course	pass_writing	grades
0	218	74	72	73.0	72	1	1	1	1	1	3
1	247	88	89	83.0	90	1	1	1	0	1	2
2	278	93	90	93.0	95	1	1	1	1	1	1
3	148	44	47	50.0	57	1	0	1	1	1	5
4	229	75	76	77.0	78	1	1	1	1	1	3

Table 10: Top Features and Grade Summary Table"

We plot a heatmap to show the correlation between each attribute

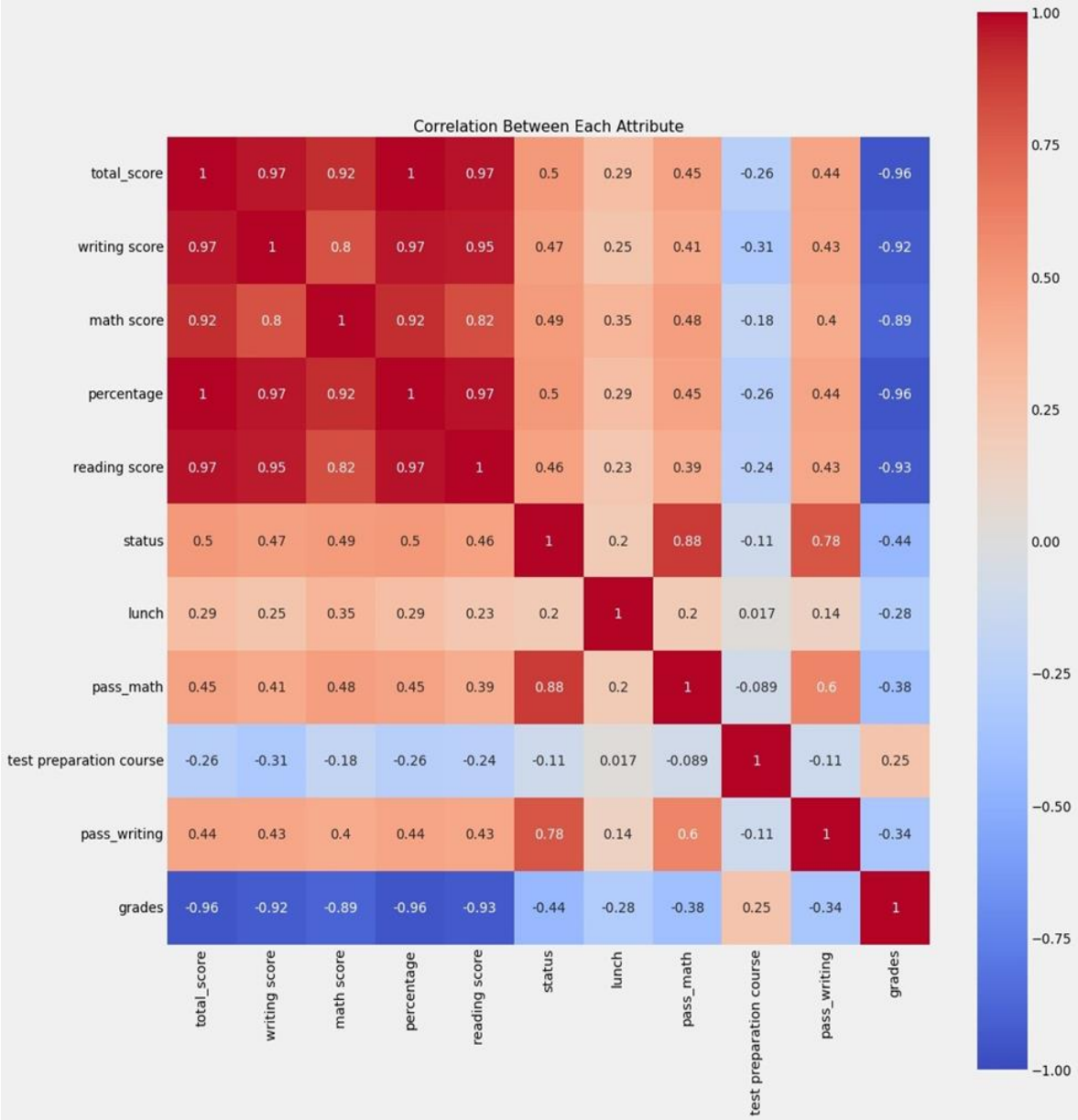

Figure 27: Feature Correlation Heatmap for Multicollinearity Analysis

Heatmap is used to see whether there are high correlating variables between the 10 selected variables, then we remove the highly correlated variables to avoid multicollinearity issue. So, the final features selected is 'percentage' 'status' 'lunch' 'test preparation course.

Figure 28: Final Features Correlation Heatmap for Model Selection

It is shown that now the selected variables are not highly correlated with each other.

D. Model Development

1. Let's split the dataset into two different sets, for training and test.

We split our dataset into training and testing to ensure that after we took the action of training our classification algorithm, the new data can be able to predict the performance. The dataset is split into 75% train and 25% test. The purpose of splitting our dataset into training and testing sets is to assess how well our classification algorithm can predict student performance on new, unseen data.

The training set is where our model learns patterns and relationships between the features (such as test scores, lunch type, test preparation courses, etc.) and the target variable (student grades). This step involves training the model to make predictions based on the training data.

Once our model has been trained, we use the testing set, which the model has never seen during training, to evaluate its performance. This testing set acts as a simulation of real-world scenarios where the model encounters new data. We can assess how well our model generalizes by making predictions on the testing set and comparing those predictions to the actual student grades.

2. Developing model via the usage of machine learning and algorithm

Logistic Regression

Logistic regression is used to predict the probability of a categorical dependent variable based on one or more predictor variables. The output is a probability that the given input point belongs to a certain class, which is transformed into a binary outcome via a threshold (e.g., if the output probability is greater than 0.5, classify as class 1, otherwise class 0) (Menard, S. 2018). The reason why we use logistic regression is because our dependent variable is in category. Logistic regression is suitable for classifying binary data. So, when we use logistic regression, we can describe our data and explain the relationship between our categorical dependent variable with our independent variables.

First, an instance of the logistic regression model is created using scikit-learn's LogisticRegression() function. The model is then fitted with the training data (x_train and y_train) to learn the relationships between the features and target variable. After training, the model is used to make predictions on the testing data (x_test), and the predictions are stored in the y_pred variable. Additionally, predictions are made for the first 10 observations in the testing data to provide a glimpse of the model's output. Finally, the training and testing accuracies are calculated and displayed. The training accuracy, approximately 94.53%, indicates how well the model fits the training data, while the testing accuracy, approximately 87.6%, provides an estimate of the model's ability to generalize to new, unseen data. These accuracy scores assess the model's predictive performance.

Result of prediction

Index	y_test	y_pred
726	3	3
243	5	5
342	3	3
976	4	4
919	1	1
658	5	5
257	3	3
634	2	2
33	6	6
922	4	4
331	6	6
485	3	3
210	2	3
536	5	4
739	5	5
52	5	5
472	2	3
396	4	5
337	5	5
809	5	5

Table 11: Result of prediction of Logistic Regression

This table shows the results for the logistic regression model where the result of y prediction is shown and the result of y test is shown. The y-test shows the first 20 outputs for grades where the data type is int64. It is shown that there are some inaccurate figures.

Random Forest

Random Forest is an ensemble of decision trees, usually trained with the bagging method. Typically, it aggregates the outputs by averaging (for regression problems) or by taking a majority vote (for classification problems). It has the ability to handle a large dataset with higher dimensionality and can handle missing values (Probst, P., & Boulesteix, A. L. 2020). First, an instance of the Random Forest Classifier model is created using scikit-learn's RandomForestClassifier() function. The model is configured to have 100 decision trees (n_estimators=100) and a specified random seed (random_state=111) for reproducibility. Next, the model is trained using the training data (x_train and y_train) by fitting it with the data. Random Forests build multiple decision trees and aggregate their predictions to make more robust and accurate predictions. After training, the model is used to predict the target variable for the testing data (x_test), and the predicted values are stored in the y_predict variable. Finally, the code calculates and displays the training and testing accuracies. The training accuracy, which is 100%, indicates that the model has perfectly

learned the training data. The testing accuracy, also 100%, suggests that the model is making perfect predictions on the testing data.

Result of prediction

Index	y_test	y_pred
726	3	3
243	5	5
342	3	3
976	4	4
919	1	1
658	5	5
257	3	3
634	2	2
33	6	6
922	4	4

Table 12: Result of prediction of Random Forest

We can see that using random forest the result is more accurate.

DEPLOY, MAINTAIN AND VISUALIZE RESULT

After testing is done above by predicting the result, we now see its validation using the confusion matrix. A confusion matrix is to validate the prediction result with the original result with the original result y_test to see the performance of our model.

A. Logistic Regression Confusion Matrix

A confusion matrix is created and displayed for the logistic regression model's predictions. It provides insights into how well the model is classifying data points.

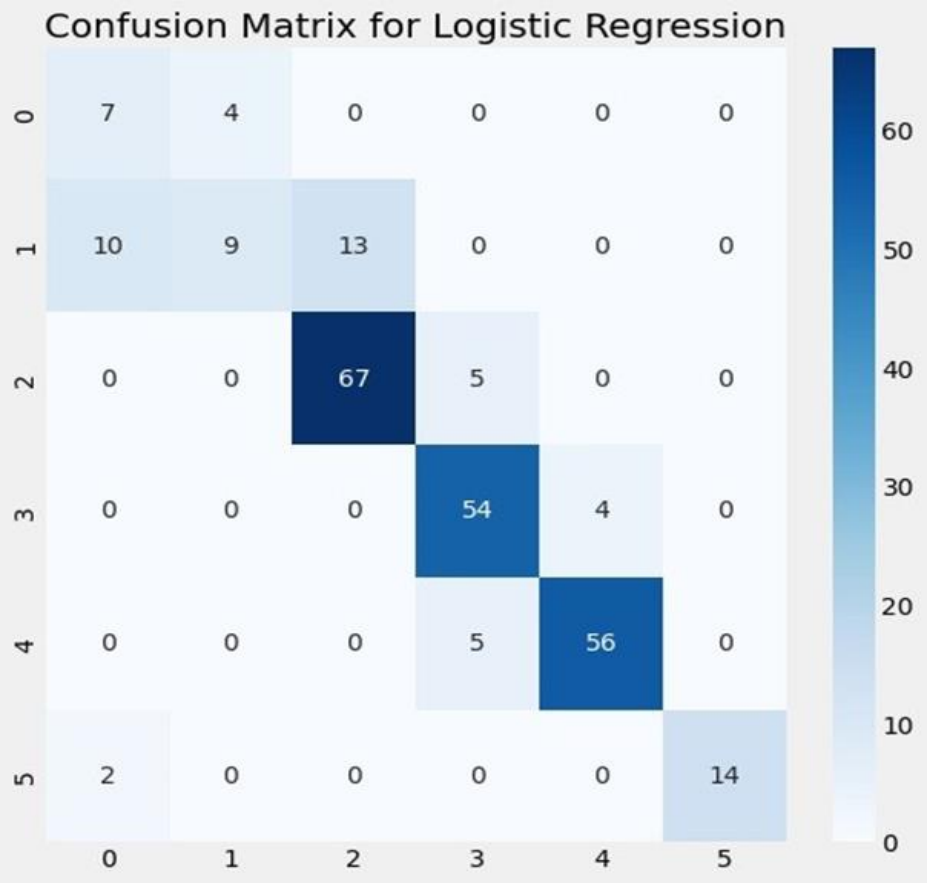

Figure 29: Confusion Matrix for Logistic Regression

From the above we can see that there are a lot of errors therefore improvement is needed. Class 2 and Class 4 have relatively high true positive values, meaning the model is predicting them well. Class 1 has been misclassified several times, particularly into Classes 0 and 2. This suggests that the model is having difficulty distinguishing Class 1 from those classes. The other classes also have some misclassifications, but they are fewer in number.

B. Random Forest Confusion Matrix

A confusion matrix is created and displayed for the Random Forest model's predictions. It provides insights into how well the model is classifying data points.

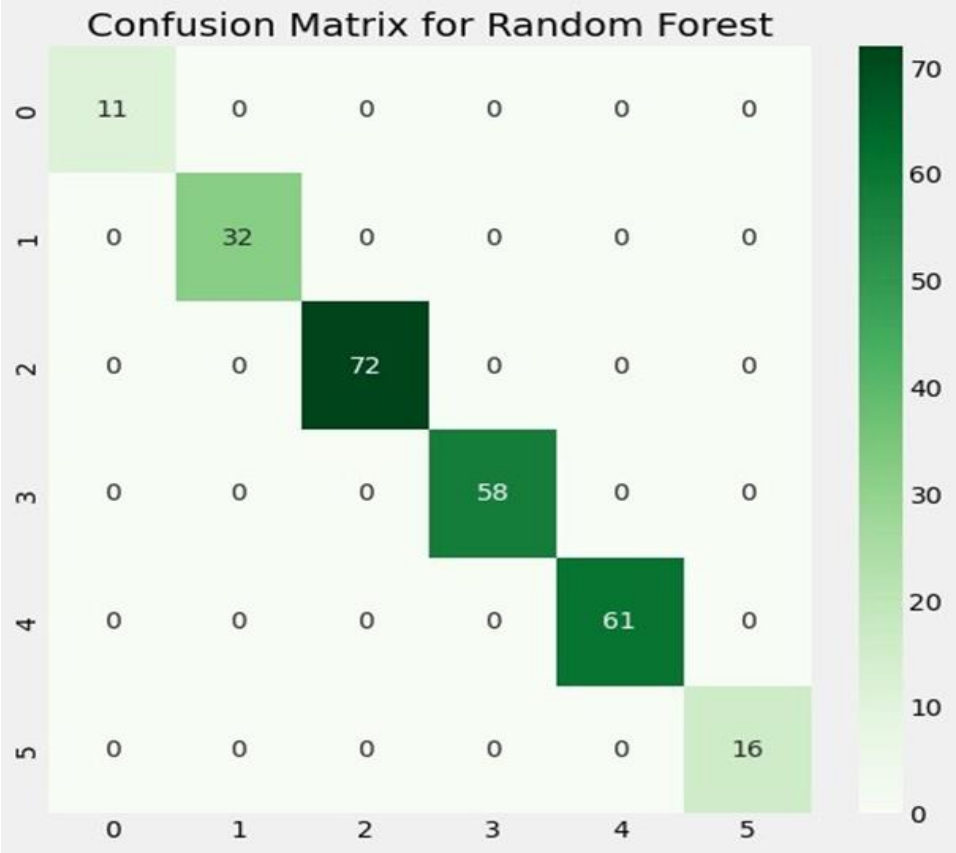

Figure 30: Confusion Matrix for Random Forest

From above we can see the Random Forest model has predicted every instance correctly for this set of data, as indicated by the fact that the off-diagonal values are all zero.

Next, We used the score method to test its validation.

C. Score for Logistic Regression

In scikit-learn, the score method is a convenient way to evaluate the performance of a machine learning model, particularly for supervised learning tasks like classification and regression. The score method calculates a performance metric based on the model's predictions and the true target values. The resulting output, "0.828," represents the accuracy of the model's predictions on the testing data. Specifically, it indicates that the logistic regression model correctly predicted the target variable for approximately 82.8% of the observations in the testing dataset. This accuracy score is a measure of the model's performance, with higher values indicating better prediction accuracy. Improvement is needed for logistic regression model.

D. Score for Random Forest

The score method calculates a performance metric based on the model's predictions and the true target values. The resulting output, "1," represents that the model has made perfect predictions on the testing data. In other words, the model correctly predicted the target variable for every observation in the testing dataset. Therefore, no improvement needs to be made.

E. Improvement for Logistic Regression

Normalization of data, especially for algorithms that rely on distance or gradient calculations like logistic regression, can often lead to improvements in model performance. The Standard Scaler is a popular technique to normalize data. What it essentially does is to transform the data such that its distribution has a mean of 0 and a standard deviation of 1. It calculates the mean and standard deviation from the training data and then uses these values to scale the training data as well as any future data. This is done for each feature (or column) independently.

The Standard Scaler normalizes data using the following formulas:

For each data point x in a feature:

1. Calculate the mean μ of the feature.
2. Calculate the standard deviation σ of the feature.
3. Normalize the data point using:

$$Z = \frac{X - \mu}{\sigma}$$

Here:

x is the original data point.

μ is the mean of the feature.

σ is the standard deviation of the feature.

z is the normalized value (often referred to as the z-score).

The result of this transformation is that the feature will have a mean of 0 and a standard deviation of 1. This ensures all features have the same scale, making models like logistic regression more stable and faster to train.

After performing Normalization:

From above logistic regression the result we got is accuracy of 0.828. Now we try to normalise our data using standard scaler for better accuracy on logistic regression model. We apply the fit_transform method of the scaler to two subsets of your dataset: x_train and x_test. This process standardizes the features in both the training and testing data, transforming them to have a mean of 0 and a standard deviation of 1.

After normalizing the data we train again on logistic regression to test its accuracy. Now we get training accuracy of 0.945 and testing accuracy of 0.876. Thus, the accuracy of logistic model has improved from 0.828 to 0.876.

10. Interpreting model and visualizing model data

We use plt.bar to create a bar chart. The names list contains the model names, and the results list contains their corresponding accuracy scores. The bars are colored differently, with "Logistic Regression" in pale violet red and "Random Forests" in gainsboro (light gray).

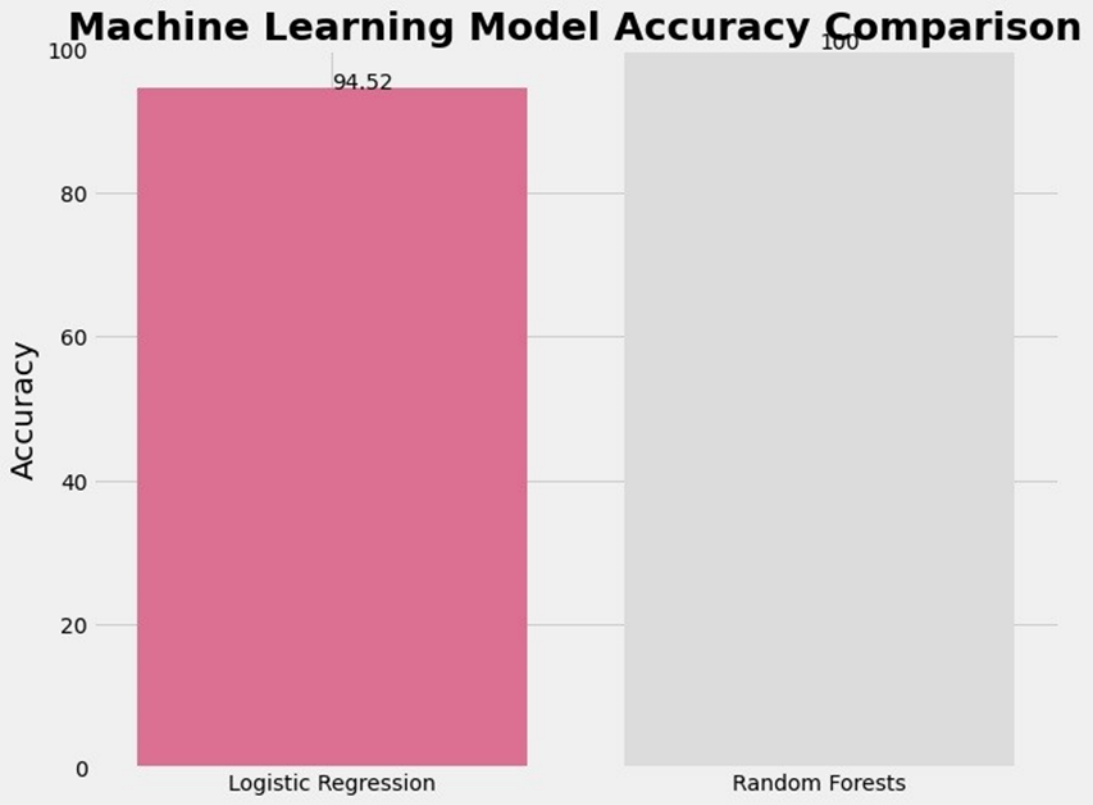

Figure 31: Machine Learning Model Accuracy Comparison

The accuracy for Logistic Regression model and Random Forest are depicted as approximately 94.52% and 100% respectively.

A. Below is how the random forest performing classification.

We plot tree number 0

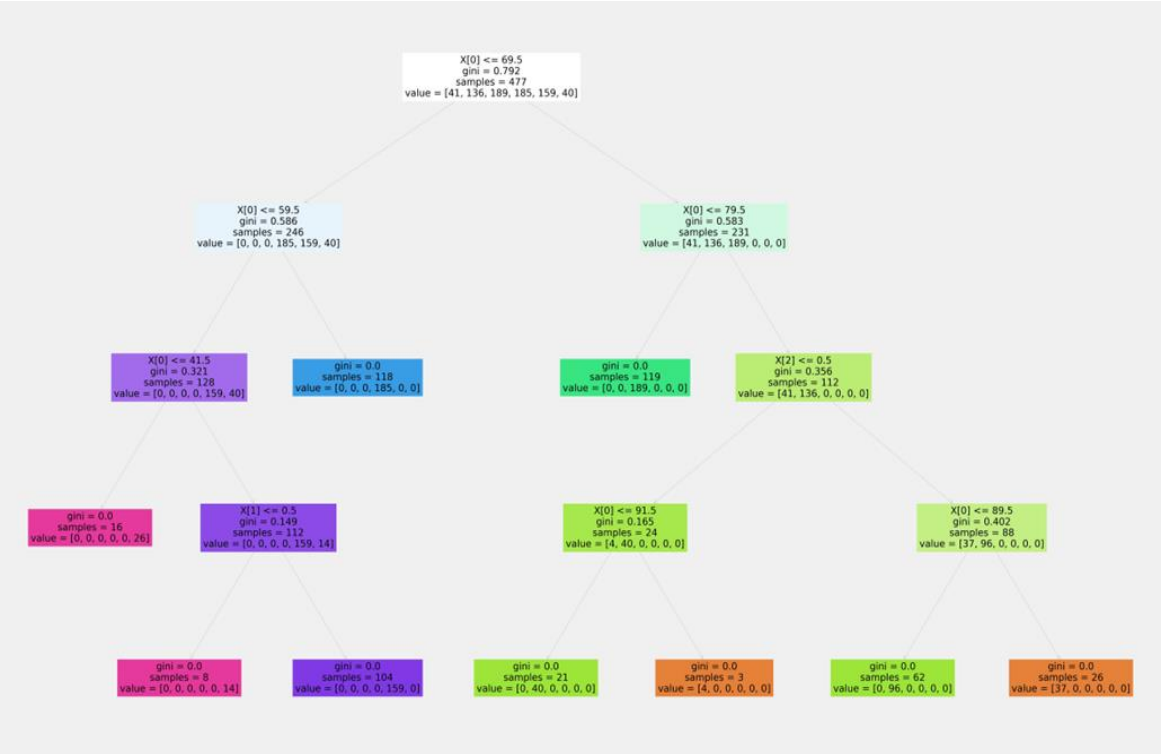

Figure 32: Decision Tree Visualization for Tree #0

We plot tree number 27

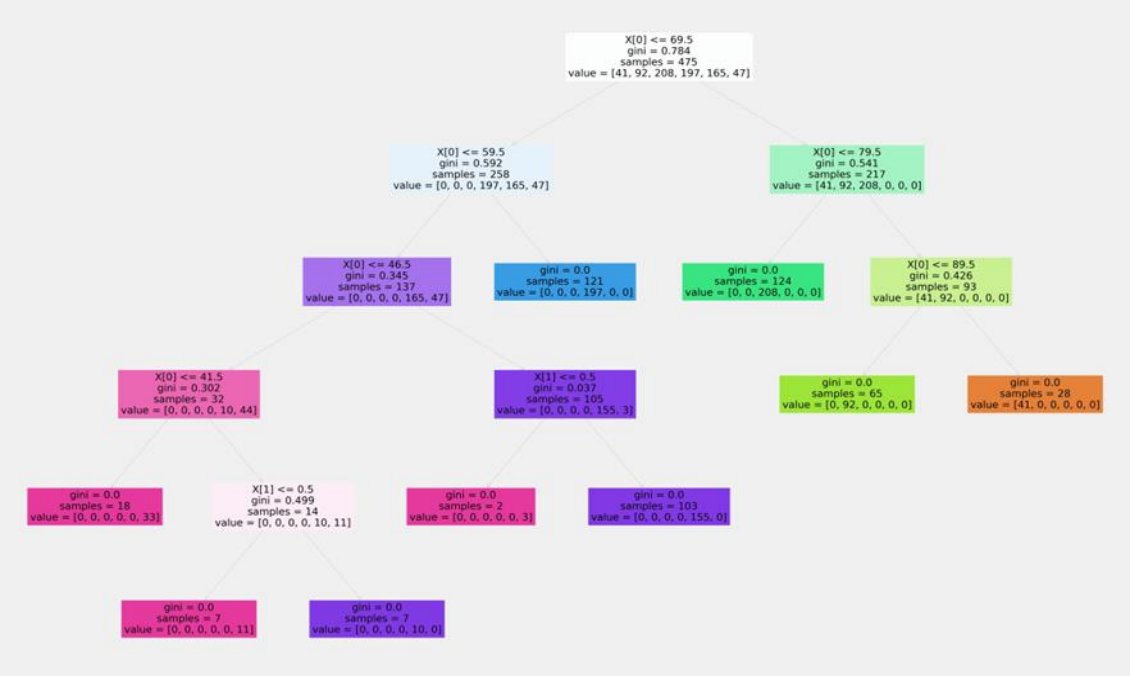

Figure 33: Decision Tree Visualization for Tree #27

We plot tree number 70

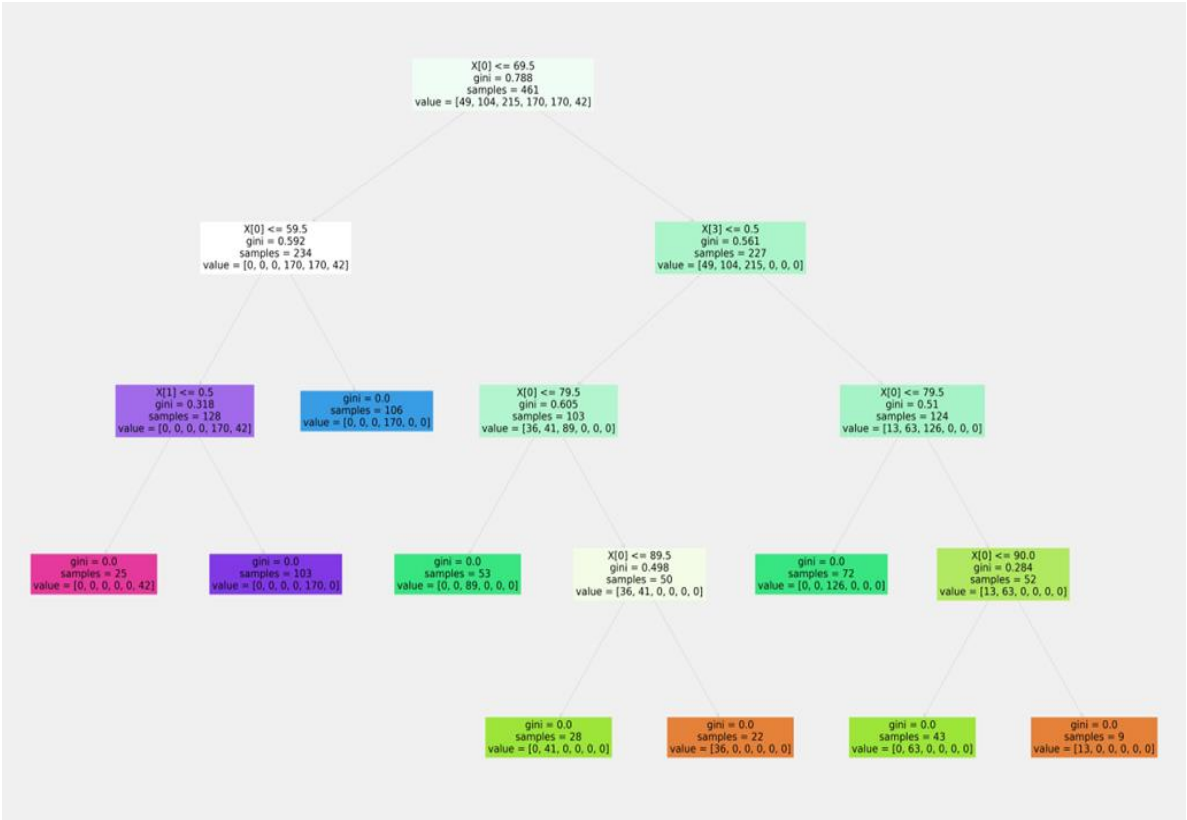

Figure 34: Decision Tree Visualization for Tree #70

Previously, we set that our random forest is made up of 100 decision trees. We only shown 3 of them to visualize how they classify at tree 0, 27 and 70. The concept of decision tree is that they perform data classification by choosing the most votes of the classification from the decision tree as final result.

Key Observation from the diagram-

- Many leaves have a Gini impurity of 0.0, indicating clear classifications contributing to the high accuracy.
- Trees use different features and thresholds at diverse levels, showing the model's flexibility.
- Nodes have mixed class distributions, illustrating the model's multi-class classification ability.
- Each tree has distinct branches, highlighting the ensemble nature of Random Forest, aiding in accuracy.

This model has led to an accuracy of 100%

B. Feature Importance Computed with SHAP Values

The SHAP interpretation is used to compute the importance of various features from a Random Forest model. This is done by using the Shapley values from game theory to estimate how each feature contribute to the prediction. Given a prediction function f , we can represent its output as:

$$f(x) = \phi_0 + \sum_{i-1}^{N} \phi_i x_i$$

Where:

- $f(x)$ is the prediction for input x.
- ϕ_0 is the base (or expected) value.
- N is the total number of features.
- x_i is the value of the ith feature.
- ϕ_i is the SHAP value of the ith feature which represents its contribution to the prediction.

From the given SHAP plot, the SHAP value (ϕ) for each feature can be interpreted as:

$$\phi_{\text{Feature 1}} > \phi_{\text{Feature 0}} > \phi_{\text{Feature 2}} > \phi_{\text{Feature 3}}$$

This indicates that the contribution of "Feature 1" to the prediction is highest, followed by "Feature 0", "Feature 2", and lastly "Feature 3".

The following figure shows the SHAP feature importance for the random forest model for predicting grades.

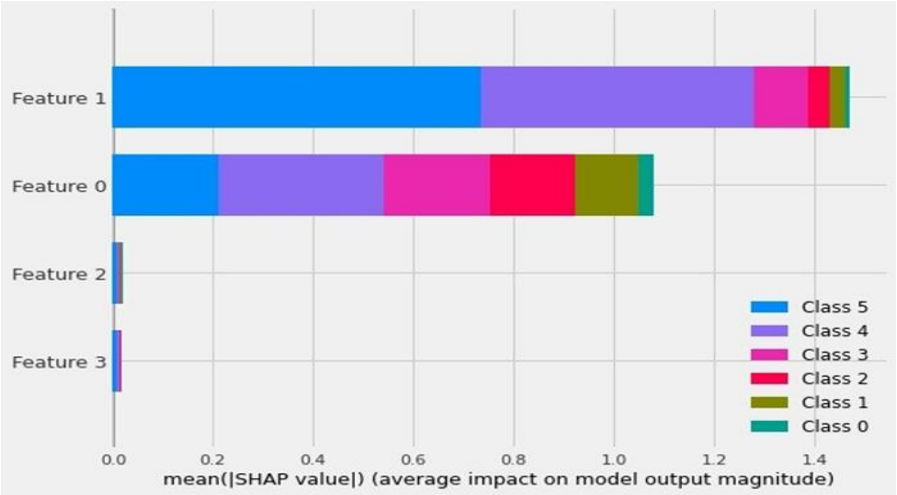

Figure 35: SHAP feature importance for the random forest model for predicting grades

Feature 0: Percentage

Feature 1: Status

Feature 2: Lunch

Feature 3: Test Preparation Course

As seen from the graph, "Feature 1" has the most significant influence on the model's predictions, with the highest average SHAP value. "Feature 0" shows considerable influence from multiple classes, indicating it plays a multifaceted role in grade determination. "Feature 2" and "Feature 3" have relatively lower SHAP values, suggesting their lesser impact on the model's predictions.

11. Limitation

Using race or ethnicity to determine student grades is tricky. Even within one racial or ethnic group, there are vast differences in backgrounds, beliefs, and experiences. Just because two students share the same race doesn't mean they've had the same opportunities or challenges. So, when we try to tie grades to race or ethnicity, our results might not be precise because it's such a broad and varied category.

The second limitation that occurred in our assignment is that our target variable is not the best dependent variable to measure student performance. This is because we only took math score, reading score, and also

writing score as our y variable. To measure the performance of a student, the test scores are far from enough. Not all students can perform well in the study, however, some perform better in physical activities like sports, arts, music, etcetera. Therefore, the final analysis may not be that accurate. We've relied heavily on traditional testing methods like written exams. However, these might not be the best measure for all students. Some might excel in project-based assessments, presentations, or group work. Our analysis might overlook these alternative skill sets. There are countless external factors affecting a student's grades: quality of teaching, classroom environment, family support, personal health, and even daily nutrition. Our analysis doesn't consider these factors, and they can have a big influence on a student's performance. In the future, our team would like to investigate and analyze the dataset's characteristics in greater depth in order to extract the best features and improve accuracy. Furthermore, we would like to experiment with different algorithms as well as design and come up with our own algorithm that can reliably forecast for this dataset.

12. Conclusion

The goal of this task was to determine if students' performance in exams could be predicted based on the attributes and data provided by the Kaggle "Student Performance in Exams" dataset. Using the score method for validation, the accuracy of the model obtained is 82.8% for logistic regression, 87.6% for post-tuned logistic regression, and 100% for random forest. Therefore, it can be concluded that random forest is the best model for predicting student performance. From our work, there are limitations that can be seen. For example, we have only used two models to examine the attributes of the data. In the future, other models such as artificial neural network could be used to study the attributes in depth. Moreover, the attributes in the dataset we have chosen are limited, when in reality there are many more factors which may contribute to a student's performance in exams such as learning environment, interactivity during classes, and individual personality. Our model can be said to have achieved its goal to a certain extent. The data mining techniques that we have used have allowed us to predict students' performance on tests. However, further improvements can be made in order to build a more complete model which includes many more attributes.

13. References

[1] Elijah, A.V., Abdullah, A., Jhanjhi, N.Z., Supramaniam, M. & Balogun, A.O. "Ensemble and Deep-Learning Methods for Two-Class and Multi-Attack Anomaly Intrusion Detection: An Empirical Study" International Journal of Advanced Computer Science and Applications(IJACSA), **10**(9), http://dx.doi.org/10.14569/IJACSA.2019.0100969 (2019).

[2] Alalawi, K., Athauda, R. & Chiong, R. Contextualizing the current state of research on the use of machine learning for student performance prediction: A systematic literature review. *Engineering Reports*. https://doi.org/10.1002/eng2.12699 (2023).

[3] Alex, S. A., Ponkamali, S., Andrew, T. R., Jhanjhi, N. Z., & Tayyab, M. Machine Learning-Based wearable Devices for smart healthcare application with risk factor monitoring. In *IGI Global eBooks* (pp. 174–185). https://doi.org/10.4018/978-1-7998-9201-4.ch009 (2022).

[4] Ali, U. *Using UX design principles for comprehensive data visualisation*. DIVA. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1802066&dswid=-9243 (2023).

[5] Alieva, I. How American media framed 2016 presidential election using data visualization: the case study of The New York Times and The Washington Post. *Journalism Practice*, **17**(4), 814–840. https://doi.org/10.1080/17512786.2021.1930573 (2021).

[6] Alsamman, A. M. et al. AlignStatPlot: An R package and online tool for robust sequence alignment statistics and innovative visualization of big data. *PLOS ONE*, **18**(9), e0291204. https://doi.org/10.1371/journal.pone.0291204 (2023).

[7] Bharadiya, J. P. (2023, July 6). *Leveraging machine learning for enhanced business intelligence*. https://ijcst.com.pk/IJCST/article/view/234 (2023).

[8] Chaudhary, M., Gaur, L., Jhanjhi, N. Z., Masud, M., & Aljahdali, S. Envisaging employee churn using MCDM and machine learning. *Intelligent Automation and Soft Computing*, **33**(2), 1009–1024. https://doi.org/10.32604/iasc.2022.023417 (2022).

[9] Cruz, T., Jiménez, F., Bravo, A. R. Q., & Ander, E. DataXploreFines: Generalized Data for Informed Decision, Making, An Interactive Shiny Application for Data Analysis and Visualization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2307.11056 (2023).

[10] Dhamala, E., Yeo, B. T. T., & Holmes, A. J. One size does not fit all: Methodological Considerations for Brain-Based Predictive Modeling in Psychiatry. *Biological Psychiatry*, **93**(8), 717–728. https://doi.org/10.1016/j.biopsych.2022.09.024 (2023).

[11] Dou, B. et al. (2023). Machine learning methods for small data challenges in molecular science. *Chemical Reviews*, **123**(13), 8736–8780. https://doi.org/10.1021/acs.chemrev.3c00189

[12] Zahra, F., Jhanjhi, N. Z., Brohi, S. N., & Malik, N. A. Proposing a Rank and Wormhole Attack Detection Framework using Machine Learning. *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. https://doi.org/10.1109/macs48846.2019.9024821 (2019).

[13] Gaur, L. et al. Disposition of youth in predicting sustainable development goals using the neuro-fuzzy and random forest algorithms. Human-Centric Computing and Information Sciences, **11,** NA (2021).

[14] Gaur, L. et al. Capitalizing on big data and revolutionary 5G technology: Extracting and visualizing ratings and reviews of global chain hotels. Computers and Electrical Engineering, **95,** 107374 (2021).

[15] Gouda, W., Sama, N. U., Al-Waakid, G., Humayun, M., & Jhanjhi, N. Z. (2022, June). Detection of skin cancer based on skin lesion images using deep learning. In Healthcare **10**(7), 1183, MDPI (2022).

[16] George, D., Dr.V.Sujatha, George, A., & Dr. T.Baskar. Bringing Light to Dark Data: A framework for unlocking hidden business value. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.8262384 (2023).

[17] Gobena, G. A. Family Socio-economic Status Effect on Students' Academic Achievement at College of Education and Behavioral Sciences, Haramaya University, Eastern Ethiopia. Journal of Teacher Education and Educators , **7**(3) , 207-222 . Retrieved from https://dergipark.org.tr/en/pub/jtee/issue/43443/530195 (2018).

[18] Gobena, G.A. Family Socio-economic Status Effect on Students' Academic Achievement at College of Education and Behavioral Sciences, Haramaya University, Eastern Ethiopia. Journal of Teacher Education andEducators, [online] **7**(3), 207–222. Available at: https://dergipark.org.tr/en/pub/jtee/issue/43443/530195 (2018).

[19] Hamadani, A., Ganai, N. A., & Bashir, J. Artificial neural networks for data mining in animal sciences. *Bulletin of the National Research Centre*, **47**(1). https://doi.org/10.1186/s42269-023-01042-9 (2023).

[20] Hofer-Pottala, S. (2023, May 5). *Approaches to Data Visualization in Technical Communication Research: A Systematic Literature review*. https://conservancy.umn.edu/handle/11299/254252 (2023).

[21] Hussain, K., Hussain, S. J., Jhanjhi, N. Z., & Humayun, M. (2019, April). SYN flood attack detection based on bayes estimator (SFADBE) for MANET. In 2019 International Conference on Computer and Information Sciences (ICCIS), 1- 4, IEEE (2019).

[22] Jain, P., Tripathi, V., Malladi, R., & Khang, A. Data-Driven Artificial Intelligence (AI) models in the workforce development planning. In *CRC Press eBooks*, 159–176. https://doi.org/10.1201/9781003357070-10 (2023).

[23] Khan, S., & Shaheen, M. From data mining to wisdom mining. *Journal of Information Science*, **49**(4), 952–975. https://doi.org/10.1177/0165551521103087 (2021).

[24] Kitchenham, B., et al. Systematic literature reviews in software engineering–a tertiary study. Information and Software Technology, **52**(8), 792-805 (2010).

[25] Kok, S., Azween, A., & Jhanjhi, N. Z. Evaluation metric for crypto-ransomware detection using machine learning. *Journal of Information Security and Applications*, **55,** 102646. https://doi.org/10.1016/j.jisa.2020.102646 (2020).

[26] Kufel, J. et al. What is machine Learning, artificial neural networks and Deep Learning?—Examples of Practical applications in medicine. *Diagnostics*, **13**(15), 2582. https://doi.org/10.3390/diagnostics13152582 (2023).

[27] Kumar, T., Pandey,B., Mussavi, S.H.A. & Jhanjhi, N.Z. "CTHS based energy efficient thermal aware image ALU design on FPGA." Wireless Personal Communications **85,** 671-696 (2015).

[28] Lee, S., Abdullah, A., Jhanjhi, N. Z., & Kok, S. Honeypot Coupled Machine Learning Model for Botnet Detection and Classification in IoT Smart Factory – An Investigation. *MATEC Web of Conferences*, **335,** 04003. https://doi.org/10.1051/matecconf/202133504003 (2021).

[29] Lim, M., Abdullah, A., Jhanjhi, N.Z. & Supramaniam, M. "Hidden link prediction in criminal networks using the deep reinforcement learning technique." Computers **8**(1) (2019).

[30] Lim, M., Abdullah, A., & Jhanjhi, N. Z. Performance optimization of criminal network hidden link prediction model with deep reinforcement learning. Journal of King Saud University-Computer and Information Sciences, **33**(10), 1202-1210 (2021).

[31] Li, H., & Xiong, Y. The relationship between test preparation and state test performance: Evidence from the Measure of Effective Teaching (MET) project. *Education Policy Analysis Archives*, **26,** 64. https://doi.org/10.14507/epaa.26.3530 (2018).

[32] Li, Z., & Qiu, Z. How does family background affect children's educational achievement? Evidence from Contemporary China. *The Journal of Chinese Sociology*, **5**(1). https://doi.org/10.1186/s40711-018-0083-8 (2018).

[33] Ma, D., Li, X., Lin, B., Zhu, Y., & Yue, S. A dynamic intelligent building retrofit decision-making model in response to climate change. *Energy and Buildings*, **284,** 112832. https://doi.org/10.1016/j.enbuild.2023.112832 (2023).

[34] Masini, A. et al. Mediterranean diet, physical activity, and family characteristics associated with cognitive performance in Italian primary school children: analysis of the I-MOVE project. *European Journal of Pediatrics*, **182**(2), 917–927. https://doi.org/10.1007/s00431-022-04756-6 (2022).

[35] Melitoshevich, V. A. (2023, February 6). *Development by a graphic user interface - programs in the Tkinter package using modern pedagogical technologies in the field of medicine*. https://miastoprzyszlosci.com.pl/index.php/mp/article/view/1081 (2023).

[36] Menon, S. et al. Blockchain and machine learning inspired secure smart home communication network. *Sensors*, **23**(13), 6132. https://doi.org/10.3390/s23136132 (2023).

[37] Moerland, T. M., Broekens, J., Plaat, A., & Jonker, C. M. Model-based Reinforcement Learning: a survey. *Foundations and Trends in Machine Learning*, **16**(1), 1–118. https://doi.org/10.1561/2200000086 (2023).

[38] Mundargi, Z. K. et al. Plotplay: An Automated Data Visualization Website using Python and Plotly. *2023 International Conference for Advancement in Technology (ICONAT)*. https://doi.org/10.1109/iconat57137.2023.10079977 (2023).

[39] Nanglia, S., Ahmad, M., Khan, F.A. & Jhanjhi, N.Z. "An enhanced Predictive heterogeneous ensemble model for breast cancer prediction." Biomedical Signal Processing and Control **72,** 103279 (2022).

[40] Omar, I., Khan, M. A., & Starr, A. Suitability analysis of machine learning algorithms for crack growth prediction based on dynamic response data. *Sensors*, **23**(3), 1074. https://doi.org/10.3390/s23031074 (2023).

[41] Pandey, S. et al. A review of current perspective and propensity in Reinforcement Learning (RL) in an orderly manner. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 206–227. https://doi.org/10.32628/cseit2390147 (2023).

[42] Priya, S., Kumar, A., Singh, D. B., Jain, P., & Tripathi, G. Machine learning approaches and their applications in drug discovery and design. *Chemical Biology & Drug Design*, **100**(1), 136–153. https://doi.org/10.1111/cbdd.14057 (2022).

[43] Rani, V., Nabi, S. T., Kumar, M., Mittal, A., & Kumar, K. Self-supervised Learning: A succinct review. *Archives of Computational Methods in Engineering*, **30**(4), 2761–2775. https://doi.org/10.1007/s11831-023-09884-2 (2023).

[44] Razaque, A. et al. Quality of Service Generalization using Parallel Turing Integration Paradigm to Support Machine Learning. *Electronics*, **12**(5), 1129. https://doi.org/10.3390/electronics12051129 (2023).

[45] Saeed, S., Jhanjhi, N. Z., Naqvi, M., Ponnusamy, V., & Humayun, M. Analysis of climate prediction and climate change in Pakistan using data mining techniques. In *Advances in computer and electrical engineering book series*, 321–338. https://doi.org/10.4018/978-1-7998-2803-7.ch016 (2020).

[46] Saha, E., & Rathore, P. Discovering hidden patterns among medicines prescribed to patients using Association Rule Mining Technique. *International Journal of Healthcare Management*, **16**(2), 277–286. https://doi.org/10.1080/20479700.2022.2099335 (2022).

[47] Saleh, M., Jhanjhi, N. Z., Abdullah, A., & Saher, R. IoTES (A Machine learning model) Design dependent encryption selection for IoT devices. *2022 24th International Conference on Advanced Communication Technology (ICACT)*. https://doi.org/10.23919/icact53585.2022.9728960 (2022).

[48] Sennan, S. et al. Energy efficient optimal parent selection based routing protocol for Internet of Things using firefly optimization algorithm. Transactions on Emerging Telecommunications Technologies, **32**(8), e4171 (2021).

[49] Shafiq, M. et al. Robust Cluster-Based Routing Protocol for IoT-Assisted Smart Devices in WSN. Computers, Materials & Continua, **67**(3) (2021).

[50] Verma, S. et al. “Intelligent Framework Using IoT-Based WSNs for Wildfire Detection," in IEEE Access, **9,** 48185-48196, doi: 10.1109/ACCESS.2021.3060549 (2021).

[51] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. Machine Learning Approaches for Load Balancing in Cloud Computing Services. *2021 National Computing Colleges Conference (NCCC)*. https://doi.org/10.1109/nccc49330.2021.9428825 (2021).

[52] Sharifani, K. *Machine Learning and Deep Learning: A review of Methods and applications*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4458723 (2023).

[53] Shouman, M. New Weather Forecasting Applications. *Alexandria Journal of Managerial Research and Information Systems*, **1**(1), 45–70. https://doi.org/10.21608/ajmris.2023.317449 (2023).

[54] Simon, C. G. K., Jhanjhi, N. Z., Wei, G. W., & Sukumaran, S. Applications of Machine Learning in Knowledge Management System: A Comprehensive Review. *Journal of Information & Knowledge Management*, **21**(02). https://doi.org/10.1142/s0219649222500174 (2022).

[55] Singh, M., Kanroo, M. S., Kawoosa, H. S., & Goyal, P. Towards accessible chart visualizations for the non-visuals: Research, applications and gaps. *Computer Science Review*, **48,** 100555. https://doi.org/10.1016/j.cosrev.2023.100555 (2023).

[56] Sujatha, R., Sl, A., Chatterjee, J. M., Alaboudi, A. A., & Jhanjhi, N. Z. A machine learning way to classify autism spectrum Disorder. *International Journal of Emerging Technologies in Learning (Ijet)*, **16**(06), 182. https://doi.org/10.3991/ijet.v16i06.19559 (2021).

[57] Tsui, K., Chen, V., Jiang, W., Yang, F., & Chen, K. (2023). Data mining methods and applications. In *Springer handbooks* (pp. 797–816). https://doi.org/10.1007/978-1-4471-7503-2_38

[58] Wang, Y., Wallmersperger, T., & Ehrenhofer, A. Application of back propagation neural networks and random forest algorithms in material research of hydrogels. *Proceedings in Applied Mathematics & Mechanics*, **23**(1). https://doi.org/10.1002/pamm.202200278 (2023).

[59] Wassan, S., Chen, X., Shen, T., Waqar, M., & Jhanjhi, N. Z. Amazon Product Sentiment Analysis using Machine Learning Techniques. *International Journal of Early Childhood Special Education (INT-JECSE)*, **30**(1), 695. https://doi.org/10.24205/03276716.2020.2065 (2021).

[60] Yang, X., Klein, B., Li, G., & Gopaluni, R. B. Evaluation of logistic regression and support vector machine approaches for XRF based particle sorting for a copper ore. *Minerals Engineering*, **192,** 108003. https://doi.org/10.1016/j.mineng.2023.108003 (2023).

[61] Zaheer, A., Tahir, S., Humayun, M., Almufareh, M. F., & Jhanjhi, N. Z. A novel Machine learning technique for fake smart watches advertisement detection. *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. https://doi.org/10.1109/macs56771.2022.10023151 (2022).

[62] Zahra, F. T. et al. Rank and wormhole attack detection model for RPL-Based internet of things using Machine learning. *Sensors*, **22**(18), 6765. https://doi.org/10.3390/s22186765 (2022).

[63] Zahra, F. T. et al. Protocol-Specific and sensor Network-Inherited attack detection in IoT using machine learning. *Applied Sciences*, **12**(22), 11598. https://doi.org/10.3390/app122211598 (2022).

[64] Zeineddine, H., Braendle, U. C., & Farah, A. Enhancing prediction of student success: Automated machine learning approach. *Computers & Electrical Engineering*, **89,** 106903. https://doi.org/10.1016/j.compeleceng.2020.106903 (2021).

[65] Zhang, H., Xu, H., Zhao, S., & Zhou, Q. Learning discriminative representations and decision boundaries for open intent detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **31,** 1611–1623. https://doi.org/10.1109/taslp.2023.3265203 (2023).

Appendix

```
#import the required libraries to read the dataset
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#import dataset
#reading the data into the dataframe into the object data
df = x = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Student Performance/StudentsPerformance.csv", header=0)

#Feature Overview of our dataset
#Finding number of rows and columns
print("Number of rows and columns: ",df.shape)

#List all the column
df.info

#Basic Information of each column using info() function
df.info()

#Basic Statistics of each column for quantitative data
df.describe().transpose()

#First five row
df.head()

#finding the data types of each column and checking for null
null_ = df.isna().any()
dtypes = df.dtypes
sum_na_ = df.isna().sum()
info = pd.concat([null_,sum_na_,dtypes],axis = 1,keys = ['isNullExist','NullSum','type'])
info

#Frequency of Parental level of Education
df1=df['parental level of education'].value_counts()
type(df1)

dfs = pd.DataFrame(df1)
```

```
dfs

#import matplotlib library as plt
import matplotlib.pyplot as plt
import numpy as np
# Create graph for the frequency distribution of parental level of education
dfs['parental level of education'].plot.bar()
plt.xticks(np.arange(6))
plt.xlabel('Education')
plt.ylabel('Count')
plt.title('Frequency of Parental Level of Education')
plt.show();


#Frequency distribution of lunch type
df1=df['lunch'].value_counts()
type(df1)

dfs = pd.DataFrame(df1)
dfs

#import matplotlib library as plt
import matplotlib.pyplot as plt
import numpy as np
# Create graph for the frequency distribution of lunch type
dfs['lunch'].plot.bar()
plt.xticks(np.arange(2))
plt.xlabel('lunch')
plt.ylabel('Count')
plt.title('lunch type')
plt.show();

#Frequency distribution of gender
df1=df['gender'].value_counts()
type(df1)

dfs = pd.DataFrame(df1)
dfs

#import matplotlib library as plt
import matplotlib.pyplot as plt
```

```
import numpy as np
# Create graph for the frequency distribution of gender
dfs['gender'].plot.bar()
plt.xticks(np.arange(2))
plt.xlabel('gender')
plt.ylabel('Count')
plt.title('Gender')
plt.show();

df1=df['race/ethnicity'].value_counts()
type(df1)
dfs = pd.DataFrame(df1)
dfs

#import matplotlib library as plt
import matplotlib.pyplot as plt
import numpy as np
# Create graph for the frequency distribution of race/ethnicity
dfs['race/ethnicity'].plot.bar()
plt.xticks(np.arange(5))
plt.xlabel('race/ethnicity')
plt.ylabel('Count')
plt.title('Race/ethnicity')
plt.show();

df1=df['test preparation course'].value_counts()
type(df1)
dfs = pd.DataFrame(df1)
dfs

#import matplotlib library as plt
import matplotlib.pyplot as plt
import numpy as np
# Create graph for the frequency distribution of test preparation course
dfs['test preparation course'].plot.bar()
plt.xticks(np.arange(6))
plt.xlabel('test preparation course')
plt.ylabel('Count')
plt.title('Test preparation course')
plt.show();
```

```
#Frequency distribution for quantitative data
!pip install dabl
# comparison of all other attributes with respect to Math score
import dabl
plt.rcParams['figure.figsize'] = (18, 6)
plt.style.use('fivethirtyeight')
dabl.plot(df, target_col = 'math score')

# comparison of all other attributes with respect to Reading score
import dabl
plt.rcParams['figure.figsize'] = (18, 6)
plt.style.use('fivethirtyeight')
dabl.plot(df, target_col = 'reading score')


# we set the pass mark at 40 for the three subjects.
passmarks = 40

# creating a new column pass_math, to identify either the students pass or fail
df['pass_math'] = np.where(df['math score']< passmarks, 'Fail', 'Pass')
df['pass_math'].value_counts().plot.pie(colors = ['lightblue', 'lightgreen'])

plt.title('Pass/Fail in Maths', fontweight = 30, fontsize = 20)
plt.xlabel('status')
plt.ylabel('count')
plt.show()

# creating a new column pass_reading, this column will tell us whether the students pass or fail

df['pass_reading'] = np.where(df['reading score']< passmarks, 'Fail', 'Pass')
df['pass_reading'].value_counts(dropna = False).plot.pie(colors = ['pink', 'yellow'])

plt.title('Pass/Fail in Reading', fontweight = 30, fontsize = 20)
plt.xlabel('status')
plt.ylabel('count')
plt.show()

# creating a new column pass_writing, this column will tell us whether the students pass or fail

df['pass_writing'] = np.where(df['writing score']< passmarks, 'Fail', 'Pass')
df['pass_writing'].value_counts(dropna = False).plot.pie(colors = ['orange', 'gray'])
```

```
plt.title('Pass/Fail in Writing', fontweight = 30, fontsize = 20)
plt.xlabel('status')
plt.ylabel('count')
plt.show()

#Comparision of total score of all the student
import warnings
warnings.filterwarnings('ignore')

df['total_score'] = df['math score'] + df['reading score'] + df['writing score']

sns.distplot(df['total_score'], color = 'magenta')

plt.title('comparison of total score of all the students', fontweight = 30, fontsize = 20)
plt.xlabel('total score scored by the students')
plt.ylabel('count')
plt.show()

# computing percentage for each of the students
# importing math library to use ceil
from math import *
import warnings
warnings.filterwarnings('ignore')

df['percentage'] = df['total_score']/3

for i in range(0, 1000):
    df['percentage'][i] = ceil(df['percentage'][i])

plt.rcParams['figure.figsize'] = (15, 9)
sns.distplot(df['percentage'], color = 'orange')

plt.title('Comparison of percentage scored by all the students', fontweight = 30, fontsize = 20)
plt.xlabel('Percentage scored')
plt.ylabel('Count')
plt.show()


# checking how many students have failed overall

df['status'] = df.apply(lambda x : 'Fail' if x['pass_math'] == 'Fail' or
```

```
                x['pass_reading'] == 'Fail' or x['pass_writing'] == 'Fail'
                else 'pass', axis = 1)

df['status'].value_counts(dropna = False).plot.pie(colors = ['grey', 'crimson'])
plt.title('overall results', fontweight = 30, fontsize = 20)
plt.xlabel('status')
plt.ylabel('count')
plt.show()

def getgrade(percentage, status):
  if status == 'Fail':
    return 'E'
  if(percentage >= 90):
    return 'A*'
  if(percentage >= 80):
    return 'A'
  if(percentage >= 70):
    return 'B'
  if(percentage >= 60):
    return 'C'
  if(percentage >= 40):
    return 'D'
  else :
    return 'E'

df['grades'] = df.apply(lambda x: getgrade(x['percentage'], x['status']), axis = 1 )

df['grades'].value_counts()

#Grade Pie Chart
df['grades'].value_counts(dropna = False).plot.pie(colors = ['blue','red', 'green','yellow','orange','brown'],
explode=[0.02,0.02,0.02,0.02,0.02,0.02], autopct='%1.1f%%', startangle=90)
plt.title('Grade Pie Chart', fontweight = 40, fontsize = 20)
plt.show()

df.shape

#remove duplicated values
df.drop_duplicates(inplace = True)
df.shape
```

```
#finding the data types of each column and checking for null
null_ = df.isna().any()
dtypes = df.dtypes
sum_na_ = df.isna().sum()
info = pd.concat([null_,sum_na_,dtypes],axis = 1,keys = ['isNullExist','NullSum','type'])
info
df.head(10)
```

#Removing outliers

```
fig, axes = plt.subplots(2, 3, figsize=(18, 10))

sns.boxplot(ax=axes[0, 0], data=df, x='math score')
sns.boxplot(ax=axes[0, 1], data=df, x='writing score')
sns.boxplot(ax=axes[0, 2], data=df, x='reading score')

axes[1,2].remove()

axes[0, 0].set_xlabel('math score', fontsize = 14)
axes[0, 1].set_xlabel('writing score', fontsize = 14)
axes[0, 2].set_xlabel('reading score', fontsize = 14)

plt.tight_layout()
plt.show()
```

#Data transformation using label encoder

```
from sklearn.preprocessing import LabelEncoder

# creating an encoder
le = LabelEncoder()

# label encoding for test preparation course
df['test preparation course'] = le.fit_transform(df['test preparation course'])

# label encoding for lunch
df['lunch'] = le.fit_transform(df['lunch'])

# label encoding for race/ethnicity
# we have to map values to each of the categories
df['race/ethnicity'] = df['race/ethnicity'].replace('group A', 1)
df['race/ethnicity'] = df['race/ethnicity'].replace('group B', 2)
df['race/ethnicity'] = df['race/ethnicity'].replace('group C', 3)
df['race/ethnicity'] = df['race/ethnicity'].replace('group D', 4)
```

```
df['race/ethnicity'] = df['race/ethnicity'].replace('group E', 5)

# label encoding for parental level of education
df['parental level of education'] = le.fit_transform(df['parental level of education'])

#label encoding for gender
df['gender'] = le.fit_transform(df['gender'])

# label encoding for pass_math
df['pass_math'] = le.fit_transform(df['pass_math'])

# label encoding for pass_reading
df['pass_reading'] = le.fit_transform(df['pass_reading'])

# label encoding for pass_writing
df['pass_writing'] = le.fit_transform(df['pass_writing'])

# label encoding for status
df['status'] = le.fit_transform(df['status'])

# label encoding for grades
# we have to map values to each of the categories
df['grades'] = df['grades'].replace('A*', 1)
df['grades'] = df['grades'].replace('A', 2)
df['grades'] = df['grades'].replace('B', 3)
df['grades'] = df['grades'].replace('C', 4)
df['grades'] = df['grades'].replace('D', 5)
df['grades'] = df['grades'].replace('E', 6)

df.head(5)
```

#Checking the skewness of the data

```
plt.subplot(1, 3, 1)
sns.distplot(df['math score'])

plt.subplot(1, 3, 2)
sns.distplot(df['reading score'])

plt.subplot(1, 3, 3)
sns.distplot(df['writing score'])
```

```
plt.suptitle('Checking for Skewness', fontsize = 18)
plt.show()

#Race/Ethnicity’s effect on test scores
df[['race/ethnicity','math score','reading score','writing score']].groupby('race/ethnicity').agg({'math score':'mean', 'reading score':'mean', 'writing score':['mean','count']}).round(1)

df[['race/ethnicity','math score','reading score','writing score']].groupby('race/ethnicity').mean().plot.bar()
plt.xticks(np.arange(6))
plt.title("Race/Ethnicity's effect on test scores")
plt.xlabel("Race/Ethnicity")
plt.ylabel("Test Scores")

#Gender and test preparation effect on math score
df[['gender','test preparation course','math score']].groupby(['gender','test preparation course']).agg(['mean','count']).round(1)

df[['gender','test preparation course','math score']].groupby(['gender','test preparation course']).agg('mean').plot.bar()
plt.title("Gender and Test Preparation course effect on Math score")
plt.xlabel("Gender, Test preparation course")
plt.ylabel("Math Score")


#Lunch and Gender effect on test scores
df[['lunch','gender','math score','writing score','reading score']].groupby(['lunch','gender']).agg('median')

df[['lunch','gender','math score','writing score','reading score']].groupby(['lunch','gender']).agg('median').plot.bar()
plt.title("Lunch and Gender effect on test scores")
plt.xlabel("Lunch, Gender")
plt.ylabel("Test Scores")

#Parental level of education’s effect on test score percentage
df[['parental level of education','percentage']].groupby(['parental level of education']).agg(['mean','count']).round(1)

df[['parental level of education','percentage']].groupby(['parental level of education']).mean().plot.bar()
plt.title("Parental level of education's effect on test score percentage")
plt.xlabel("Parental level of education")
plt.ylabel("Test Score Percentage")
```

```
#Test Scores Heatmap
df4 = df.loc[:,["math score", "reading score", "writing score", "total_score"]]

plt.title('Test Scores Heatmap',fontsize=20,pad=40)
sns.heatmap(df4.corr(),annot=True,linewidths=.4);

#Feature Extraction
df.info()
# splitting the dependent and independent variables

x = df.iloc[:,:14]
y = df.iloc[:,14]

print(x.shape)
print(y.shape)

#Feature Selection
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

## Univariate selection
best_features = SelectKBest(score_func=chi2, k=10)
fit = best_features.fit(x,y)

df_scores = pd.DataFrame(fit.scores_)
df_columns = pd.DataFrame(x.columns)

# Concat the two dataframes above for better visualization
feature_scores = pd.concat([df_columns, df_scores], axis = 1)
feature_scores.columns = ['Feature', 'Score']

# Higher score means the features are more important
feature_scores

# Show top 10 features
print(feature_scores.nlargest(10, 'Score'))

# Create a new dataframe with the important attributes and the target attribute
clean_df = df.loc[:,['total_score', 'writing score', 'math score', 'percentage', 'reading score', 'status', 'lunch',
'pass_math', 'test preparation course', 'pass_writing', 'grades']]
clean_df.head()
```

```
path = "/content/drive/MyDrive/Colab Notebooks/Student Performance/clean_df.csv"
df.to_csv(path, index=False)

# plot a heatmap to show the correlation between each attribute
plt.figure(figsize=(16, 18))

sns.heatmap(clean_df.corr(), vmax=1, vmin=(-1), square=True, annot=True, cmap = 'coolwarm')

plt.title('Correlation Between Each Attribute', fontsize = 16)

plt.show()

# New dataframe is created with the important attributes and the target attribute
clean_df = df.loc[:,[ 'percentage', 'status', 'lunch', 'test preparation course', 'grades']]

# plot a heatmap to show the correlation between each attribute
plt.figure(figsize=(16, 18))

sns.heatmap(clean_df.corr(), vmax=1, vmin=(-1), square=True, annot=True, cmap = 'coolwarm')

plt.title('Correlation Between Each Attribute', fontsize = 16)

plt.show()

clean_df.head()
clean_df["grades"].unique()

x = clean_df.iloc[:, 0:-1]
y = clean_df.iloc[:, -1]

print(x.shape)
print(y.shape)

# Create features and target
x = x
y = y

x.head()

from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 45)

print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)

print(x_train)
print(y_train)

#Logistic Regression Model development

from sklearn.linear_model import  LogisticRegression
# making an instance model
modeldev = LogisticRegression()
modeldev.fit(x_train, y_train)

# to feed the training data to the model
y_pred=modeldev.predict(x_test)

# predicting the multiple observation
modeldev.predict(x_test[0:10])

# calculating the accuracies
print("Training Accuracy :", modeldev.score(x_train, y_train))
print("Testing Accuracy :", modeldev.score(x_test, y_test))

#Result of prediction
print(y_pred)
y_test[0:20]

#Random Forest Model Development
from sklearn.ensemble import RandomForestClassifier

# creating a model
model1 = RandomForestClassifier(n_estimators=100, random_state=111) #we set the number of trees to
100

# feeding the training data to the model
model1.fit(x_train, y_train)
```

```
# predicting the x-test results
y_predict = model1.predict(x_test)
# calculating the accuracies
print("Training Accuracy :", model1.score(x_train, y_train))
print("Testing Accuracy :", model1.score(x_test, y_test))

#Result of prediction
y_predict
y_test[0:10]

#Logistic Regression Confusion Matrix
# printing the confusion matrix

from sklearn.metrics import confusion_matrix

# creating a confusion matrix
cm = confusion_matrix(y_test, y_pred)

# printing the confusion matrix
plt.rcParams['figure.figsize'] = (8, 8)
sns.heatmap(cm, annot = True, cmap = 'Blues')
plt.title('Confusion Matrix for Logistic Regression', fontweight = 30, fontsize = 20)
plt.show()

#Random Forest confusion Matrix
# printing the confusion matrix

from sklearn.metrics import confusion_matrix

# creating a confusion matrix
cm = confusion_matrix(y_test, y_predict)

# printing the confusion matrix
plt.rcParams['figure.figsize'] = (8, 8)
sns.heatmap(cm, annot = True, cmap = 'Greens')
plt.title('Confusion Matrix for Random Forest', fontweight = 30, fontsize = 20)
plt.show()

#Score for Logistic Regression
# Use score method to get accuracy of model
score = modeldev.score(x_test, y_test)
print(score)
```

```
#Score for Random Forest
# Use score method to get accuracy of model
score = model1.score(x_test, y_test)
print(score)

# Improvement using Normalization
from sklearn.preprocessing import StandardScaler

# Create the scaler
ss = StandardScaler()

# Apply the scaler to the DataFrame subset
x_train = ss.fit_transform(x_train)
x_test = ss.fit_transform(x_test)


#Logistic Regression after Normalization
from sklearn.linear_model import  LogisticRegression

# creating a model
modeldev = LogisticRegression()

# feeding the training data to the model
modeldev.fit(x_train, y_train)

# predicting the test set results
y_pred = modeldev.predict(x_train)

# calculating the classification accuracies
print("Training Accuracy :", modeldev.score(x_train, y_train))
print("Testing Accuracy :", modeldev.score(x_test, y_test))

# Use score method to get accuracy of model
score = modeldev.score(x_test, y_test)
print(score)

#Interpreting and visualizing model
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))
```

```
def addlabels(names,results):
    for i in range(len(names)):
        plt.text(i,results[i],results[i])

#Evaluating performance
results = [94.52,100]
names = ["Logistic Regression","Random Forests"]

bar_Compare = plt.bar(names, results, color=['palevioletred','gainsboro'])
plt.ylabel("Accuracy",fontsize=20)
plt.ylim([0, 100])
plt.title("Machine Learning Model Accuracy Comparison",fontsize=25,fontweight="bold")
addlabels(names, results)

#Random Forest Classification Performance
from sklearn import tree

# plot tree number 0
plt.figure(figsize=(150,100))
tree.plot_tree(model1.estimators_[0], filled=True)

plt.show()

# plot tree number 27
plt.figure(figsize=(150,100))
tree.plot_tree(model1.estimators_[27], filled=True)

plt.show()

# plot tree number 70
plt.figure(figsize=(150,100))
tree.plot_tree(model1.estimators_[70], filled=True)

plt.show()

#feature Importance computed with SHAP values
pip install shap
import shap

explainer = shap.TreeExplainer(model1, x)
shap_values = explainer.shap_values(x_test)
```

```
shap.summary_plot(explainer.shap_values(x_test, check_additivity=False), x_test, plot_type="bar", plot_size=(8, 6))
```