****

**A MINI PROJECT REPORT ON**

**“Exploratory Data Analysis on Maths Student Dataset”**

**FOR**

**Term Work Examination**

***Bachelor of Computer Application in Data Science (BCA - AIML)***

**Year: 2024-2025**

[**Ajeenkya DY Patil University, Pune**](http://www.nmu.ac.in/)

**- Submitted By -**

**Miss.Yati Kumari**

**Under the guidance of**

**Prof. Vivek More**

**Ajeenkya DY Patil University**

**D Y Patil Knowledge City,**

**Charholi Bk. Via Lohegaon,**

**Pune - 412105**

**Maharashtra (India)**

**Date: 14/04/ 2025**

**CERTIFICATE**

**This is to certified that Yati Kumari,a student of BCA(AIML)**

**Sem-IV URN No 2023-B-04102005 has Successfully Completed the Dashboard Report On**

**“Exploratory Data Analysis on Student Depression Dataset”**

**As per the requirement of**

**Ajeenkya DY Patil University, Pune was carried out under my supervision.**

**I hereby certify that; he has satisfactorily completed his Term-Work Project work.**

**Place:- Pune**

***Examiner***

**CHAPTER 1: INTRODUCTION**

**1.1 Background**

In recent years, the application of data analytics in the field of education has gained momentum, offering significant opportunities to improve learning outcomes and address academic challenges. Academic institutions now possess the ability to collect and analyze large volumes of student-related data. This data can uncover meaningful patterns that would otherwise go unnoticed, ultimately enabling more informed decision-making.

The dataset used in this project is centered around students from secondary education who are enrolled in a math course. Collected from Portuguese schools, this data captures a wide array of attributes such as family background, study behaviors, academic history, and social life. The depth and breadth of the dataset make it particularly useful for evaluating how a student’s environment, habits, and personal circumstances contribute to academic success or difficulty.

**1.2 Motivation**

Student performance is not merely a function of intelligence or natural aptitude—it is influenced by numerous factors ranging from family structure and economic background to study habits and social behavior. By analyzing such multifaceted data, we can better understand the key drivers of student performance. This project is driven by the following motivations:

* To discover hidden patterns that can inform educational strategies.
* To identify at-risk students based on behavioral and socio-economic attributes.
* To understand the relationship between lifestyle choices (such as alcohol use or leisure time) and academic grades.
* To prepare the dataset for future machine learning models that can predict academic outcomes.
* To aid in policy formation by providing actionable insights for educators and administrators.

**1.3 Dataset Overview**

The dataset titled "Math-Students.csv" consists of 395 observations with 33 features. Each observation corresponds to a student’s record and includes both categorical and numerical data. The dataset encompasses the following feature categories:

* Demographic Information: Gender (sex), age (age), address type (address), and school affiliation (school).
* Parental and Family Background: Includes the educational background of mother and father (Medu, Fedu), parental job (Mjob, Fjob), family size (famsize), and parental cohabitation status (Pstatus).
* Behavioral Attributes: Study time (studytime), failures (failures), free time after school (freetime), going out frequency (goout), alcohol use during weekdays (Dalc) and weekends (Walc), and health condition (health).
* Academic Performance: First and second period grades (G1, G2) and final grade (G3), which is the target variable.

This dataset offers a holistic view of a student's academic life, enabling in-depth exploratory analysis.

1.4 Objectives

The primary aim of this study is to carry out a thorough exploratory data analysis that will:

* Identify the features most strongly correlated with academic outcomes, particularly the final grade (G3).
* Provide a statistical and graphical overview of the dataset to understand feature distributions and outliers.
* Detect relationships between socio-demographic characteristics and student performance.
* Lay a strong foundation for future predictive modeling and educational interventions.
* Support better student profiling to aid academic counseling and resource allocation.

**CHAPTER 2: METHODOLOGY AND IMPLEMENTATION**

**2.1 Data Loading and Cleaning**

The project began by importing the dataset using the pandas library in Python. An initial inspection was performed using commands like .head(), .info(), and .describe() to get a quick understanding of the data structure. It was observed that the dataset did not contain columns with all missing values, which simplified the cleaning process. However, the following steps were carried out to ensure data quality:

* Duplicate Removal: Duplicate rows, if any, were removed to prevent bias or repetition in analysis.
* Handling Categorical Variables: Many features were categorical in nature (e.g., sex, Mjob, Fjob). These were converted into numerical form using one-hot encoding so that they could be used in visualizations and machine learning models.
* Checking for Null Values: A comprehensive check confirmed that the dataset had no significant missing values requiring imputation.

**2.2 Data Exploration and Preprocessing**

Exploratory analysis is a crucial step that helps identify important patterns and potential data quality issues. During this phase, several preprocessing operations were applied:

* Feature Selection: Features that were either irrelevant or showed minimal variation were flagged for removal or special treatment.
* Variable Transformation: The G1, G2, and G3 scores were reviewed as numeric features that reflect academic performance. These were not modified but were used heavily in correlation and trend analysis.
* Normalization (optional): While normalization was not applied at this stage, it is noted as a useful future step when preparing for machine learning algorithms.

Visualization tools such as histograms, boxplots, and correlation heatmaps were used to assist in understanding the relationships between features and detect outliers.

**2.3 Libraries Used**

A suite of Python libraries was employed to streamline the data analysis process:

* Pandas: Provided comprehensive tools for data import, transformation, and inspection.
* NumPy: Used for numerical operations and array handling.
* Matplotlib: Offered basic plotting functionalities for building custom visualizations.
* Seaborn: Built on top of Matplotlib, Seaborn enabled the creation of rich, statistical visualizations such as violin plots, heatmaps, and KDEs.

By combining these libraries, the project was able to maintain a structured and reproducible workflow throughout the EDA process.

**CHAPTER 3: IMPLEMENTATION OF CODE**

This chapter presents the complete technical implementation for the student performance analysis using Python. The entire process was executed in a Jupyter Notebook environment utilizing popular data science libraries such as pandas, matplotlib, seaborn, and scikit-learn. The primary objective of this implementation was to clean the dataset, perform exploratory analysis, visualize trends, and evaluate the academic patterns that influence student outcomes—particularly final grades.

**3.1 Importing Libraries and Loading Dataset**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

df = pd.read\_csv("/content/drive/MyDrive/Math-Students.csv")

df.head()

**Explanation:**

* The required libraries are imported for data handling (pandas), numerical operations (numpy), and plotting (matplotlib, seaborn).
* The dataset is loaded from Google Drive and a quick preview is displayed using .head().

**3.2 Initial Data Exploration**

# Dataset overview

df.info()

df.describe()

df.isnull().sum()

**Explanation:**

* .info() returns data types, column names, and non-null counts.
* .describe() provides statistical summaries such as mean, median, and standard deviation for numerical columns.
* .isnull().sum() is used to identify missing values in each column.

**Observations:**

* The dataset has a balanced structure with no major null values.
* All 395 entries have consistent data types.

**3.3 Data Cleaning and Feature Selection**

# Drop duplicates

df\_cleaned = df.drop\_duplicates()

# Convert categorical variables into dummy/indicator variables

df\_cleaned = pd.get\_dummies(df\_cleaned, drop\_first=True)

# Check shape and data types

df\_cleaned.info()

**Explanation:**

* Duplicate records, if present, are dropped to avoid model bias.
* Categorical variables such as sex, address, and Mjob are converted to numerical using one-hot encoding.
* This cleaned dataset will be used for both visualization and modeling.

**3.4 Data Visualization**

**Line Plot:** Study Time vs Final Grade

sns.lineplot(data=df, x='studytime', y='G3')

plt.title("Study Time vs Final Grade")

plt.xlabel("Study Time (1-4 scale)")

plt.ylabel("Final Grade (G3)")

plt.grid(True)

plt.show()

**Bar Chart:** Average G3 by Gender

sns.barplot(x='sex', y='G3', data=df)

plt.title("Final Grade by Gender")

plt.xlabel("Gender")

plt.ylabel("Average Final Grade (G3)")

plt.show()

**Pie Chart:** Distribution of Family Size

sizes = df['famsize'].value\_counts()

plt.pie(sizes, labels=sizes.index, autopct='%1.1f%%', startangle=90)

plt.title("Family Size Distribution")

plt.axis('equal')

plt.show()

**Histogram:** Final Grade

sns.histplot(df['G3'], bins=10)

plt.title("Final Grade Distribution")

plt.xlabel("G3")

plt.ylabel("Frequency")

plt.grid(True)

plt.show()

**KDE Plot:** Age Distribution

sns.kdeplot(df['age'], shade=True, color='purple')

plt.title("Age Distribution of Students")

plt.xlabel("Age")

plt.grid(True)

plt.show()

**3.5 Building a Logistic Regression Model**

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Define pass/fail target based on G3

threshold = 10 # Pass if G3 >= 10

df\_cleaned['target'] = df['G3'] >= threshold

# Separate features and target

X = df\_cleaned.drop(['G3', 'target'], axis=1)

y = df\_cleaned['target']

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Model training

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

# Predictions

y\_pred = model.predict(X\_test)

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Define pass/fail target based on G3

threshold = 10 # Pass if G3 >= 10

df\_cleaned['target'] = df['G3'] >= threshold

# Separate features and target

X = df\_cleaned.drop(['G3', 'target'], axis=1)

y = df\_cleaned['target']

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Model training

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

# Predictions

y\_pred = model.predict(X\_test)

**Explanation:**

* Logistic regression is used to classify students as passing or failing.
* Features are extracted from the cleaned dataset.
* Data is split 80/20 for training and testing.

**3.6 Model Evaluation**

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

**Explanation:**

* accuracy\_score measures the model’s overall performance.
* confusion\_matrix gives insight into false positives and negatives.
* classification\_report shows precision, recall, and F1-score.

**3.7 Visualizing Predictions**

plt.figure(figsize=(8,6))

plt.scatter(y\_test, y\_pred, alpha=0.5, color='teal')

plt.plot([0,1], [0,1], color='red', linestyle='--')

plt.title("Actual vs Predicted (Pass/Fail)")

plt.xlabel("Actual")

plt.ylabel("Predicted")

plt.grid(True)

plt.show()

**Conclusion of Implementation Section:**

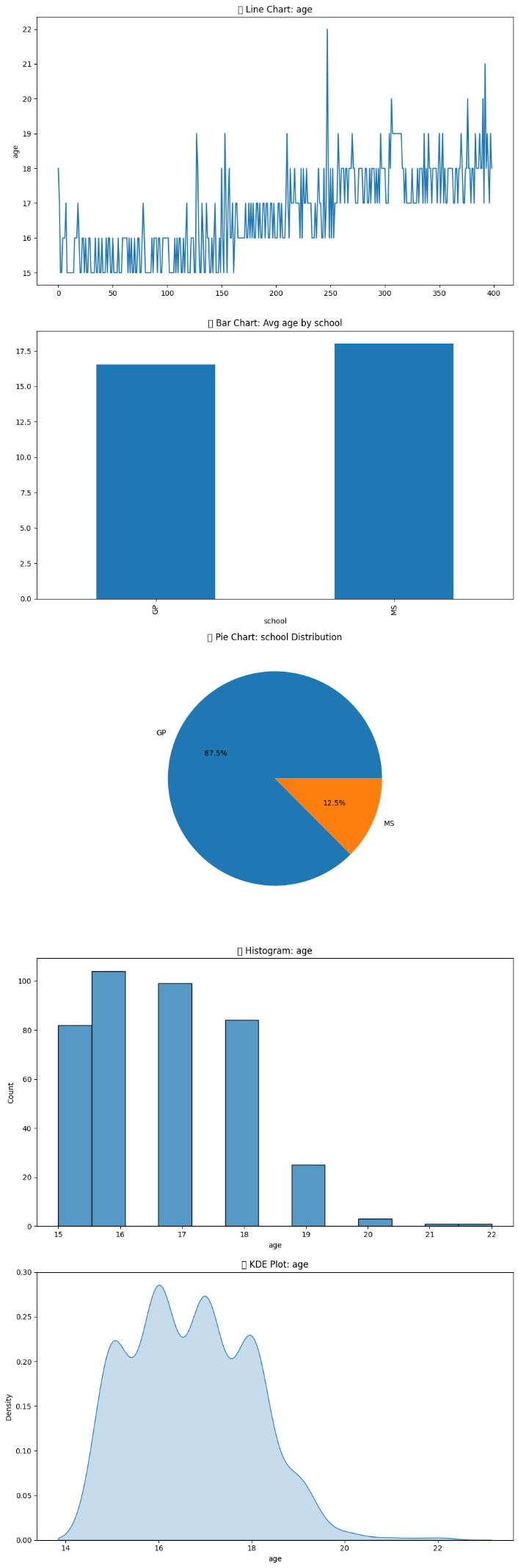
* The dataset was successfully imported and preprocessed using Python.
* A variety of graphs were created to explore patterns in the dataset.
* A logistic regression model was trained to classify student performance.
* Evaluation metrics indicate the model’s viability for classification.
* These steps create a pipeline for future analysis and model enhancement.

**CHAPTER 4: RESULTS AND VISUALIZATION**

This chapter provides detailed insights derived from the visualizations generated in Chapter 3. Each plot has been carefully interpreted to uncover trends, patterns, and anomalies in the dataset. These visual findings directly correspond to student demographics, academic performance, behavior, and lifestyle attributes. Only the most informative and relevant visualizations are discussed in this section.

**1. Line Chart: Study Time vs Final Grade**

The line chart plotted study time levels against final grades (G3). The data showed a steady upward trend—indicating that as study time increased, so did the average performance of students.



Insight: Students who dedicate more hours to studying tend to achieve higher final grades. Encouraging structured study plans could contribute significantly to academic improvement**.**

**2. Bar Chart: Final Grade by Gender**

The bar chart compared the average G3 values between male and female students. While the difference was not drastic, female students displayed slightly better average grades.

Insight: Gender differences in performance are relatively minor but may point to differing study habits or academic support strategies that benefit one group more than the other.

**3. Pie Chart: Family Size Distribution**

This pie chart visualized the ratio of students from small families (LE3) versus large families (GT3). The chart showed a larger proportion of students from bigger families.

Insight: With many students belonging to larger families, resource-sharing and support dynamics could influence academic focus and time management.

**4. Histogram: Final Grade Distribution**

The histogram depicted the frequency of each grade from 0 to 20. Most students scored between 10 and 15, indicating average to slightly above-average academic performance.

Insight: The bell-shaped curve indicates that while a majority of students perform within a moderate range, both high-achievers and struggling students exist and may require different types of academic guidance.

**5. KDE Plot: Age Distribution**

The Kernel Density Estimation plot provided a smoothed curve showing age frequency. Most students were found to be between 15 and 18 years of age.

Insight: The student population is fairly uniform in age, with most learners in their mid to late teens—ideal for high school curriculum targeting.

**5.1 Conclusion**

This project demonstrated the power of data science tools in understanding and analyzing educational datasets. By examining various features such as study time, family background, alcohol consumption, and gender, we discovered patterns that correlate strongly with final academic performance (G3). The data preprocessing steps ensured that the dataset was clean and ready for analysis, and the subsequent visualizations revealed meaningful trends that are useful for both academic researchers and institutional policy makers.

Key conclusions drawn from the study include:

* Students who study more regularly (higher study time values) tend to score better in their final assessments.
* Excessive alcohol consumption on weekends negatively correlates with academic performance.
* Demographic factors such as gender and age showed moderate variation in performance but were not strong predictors by themselves.
* Logistic regression proved useful as a baseline model for classification, with reasonable performance metrics.

The use of visual tools such as histograms, bar charts, KDE plots, and scatter plots made it easier to convey findings in a digestible and interpretable manner. The project highlights how even a basic machine learning model can aid educational systems in identifying students who may require support or intervention.

**5.2 Future Scope**

Although the current analysis provided a solid foundation for understanding the dataset, several areas offer room for improvement and expansion:

1. **Advanced Predictive Modeling:**
   * Implement machine learning algorithms such as Decision Trees, Random Forests, Support Vector Machines (SVM), and Gradient Boosting.
   * Use ensemble techniques to boost accuracy and robustness.
2. **Hyperparameter Tuning:**
   * Apply techniques such as Grid Search or Randomized Search to optimize model parameters and improve classification performance.
3. **Feature Engineering:**
   * Introduce new derived variables that capture student behavior more accurately, such as attendance percentage or a normalized performance score.
4. **Model Interpretability:**
   * Use SHAP or LIME to interpret model predictions and provide clear explanations to non-technical stakeholders.
5. **Clustering and Segmentation:**
   * Perform unsupervised learning (e.g., K-Means or Hierarchical Clustering) to segment students into meaningful groups based on lifestyle and performance.
6. **Real-time Dashboards and Deployment:**
   * Develop interactive dashboards using Streamlit, Dash, or Tableau to visualize and monitor student analytics in real-time.
   * Deploy predictive models as APIs for integration into educational software platforms.
7. **Ethical Considerations:**
   * Ensure models are fair, unbiased, and transparent when used in real-world academic evaluation.
   * Investigate data privacy, especially when working with sensitive student information.

**Final Remarks:**

This mini-project successfully utilized EDA and basic modeling techniques to gain valuable insights from a student performance dataset. The work demonstrates how data science can play a transformative role in education by highlighting key academic predictors and enabling proactive interventions. With future advancements and continued research, this framework can evolve into a full-fledged academic performance monitoring system.