

FINAL REPORT

ML-STUDENT-PERFORMANCE

Prepared by:

- Koffi Jean-Luc Junior Adjiey
- Anna Providence Aloyem
- Josué Kinsanh Nixxon Koffi
- Joel Rabi Dalyl Joel Tiendrebeogo

Supervisors :

- Murouj Aljamaeen
- Ali Sadegh Zadeh

SUMMARY

INTRODUCTION.....	4
PHASE 1: DATA EXPLORATION AND PREPROCESSING	5
I. DATASET SUMMARY & PROBLEM DEFINITION.....	6
1. Dataset Summary.....	6
2. Data Types of Each Feature.....	6
3. Obvious Patterns and Relationships.....	6
4. Problem Type: Classification	7
5. Rationale for Dataset Selection.....	7
II. EXPLORATORY DATA ANALYSIS (EDA)	9
1. Descriptive Statistics.....	9
2. Missing Values.....	9
3. Outliers	10
4. Visualizations	11
5. Key Observations.....	13
III. DATA PREPROCESSING	15
1. Handling Missing Values	15
2. Encoding Categorical Variables.....	15
3. Scaling Numerical Features.....	16
PHASE 2: MODEL IMPLEMENTATION AND EVALUATION	17
I. PROBLEM TYPE CONFIRMATION	18
II. MODEL SELECTION AND JUSTIFICATION	19
1. Decision Tree Classifier	19
2. K-Nearest Neighbors (KNN)	19
3. Random Forest Classifier	19
4. Logistic Regression	20
III. MODEL IMPLEMENTATION	21
1. Data Splitting.....	21
2. Model Training	21

3. Predictions	22
IV. MODEL EVALUATION	23
1. Metrics Used: Accuracy, Precision, Recall, F1-Score	23
2. Confusion Matrices.....	24
3. ROC Curve Analysis (for Random Forest and Logistic Regression).....	26
4. Performance Comparison Table	27
5. Discussion and Interpretation.....	28
PHASE 3: INSIGHTS, RECOMMENDATIONS, AND FUTURE WORK	29
I. INSIGHTS FROM MODEL RESULTS.....	30
II. RECOMMENDATIONS.....	31
III. LIMITATIONS AND FUTURE WORK.....	32
➤ Limitations.....	32
➤ Future Work	32
CONCLUSION.....	33
GROUP CONTRIBUTION SUMMARY	34

INTRODUCTION

In an increasingly data-driven world, educational institutions can greatly benefit from predictive analytics to identify students at risk of failure and offer timely support. This project explores the use of supervised machine learning techniques to predict student performance based on real-world data from Portuguese secondary schools.

The primary goal is to build accurate and interpretable models that can classify whether a student will pass or fail. The dataset used contains a wide range of demographic, academic, and behavioral features, providing a rich basis for analysis.

The project is divided into three main phases:

1. **Exploration and Preprocessing**, where we clean and prepare the data;
2. **Model Implementation and Evaluation**, where we train and compare multiple machine learning models;
3. **Insights and Recommendations**, where we interpret the results and discuss real-world implications, limitations, and future work.

Through this process, we aim not only to build predictive models but also to extract meaningful insights that could inform interventions in real educational settings.

PHASE 1:

DATA EXPLORATION AND PREPROCESSING

I. DATASET SUMMARY & PROBLEM DEFINITION

1. Dataset Summary

The **Student Performance Data** dataset was selected from Kaggle (available at: <https://www.kaggle.com/datasets/devansodariya/student-performance-data/data>) to analyse and predict academic performance among secondary school students.

It includes data collected from two Portuguese secondary schools — Gabriel Pereira (GP) and Mousinho da Silveira (MS) — and consists of **395 instances (students)** described by **33 features**.

The features represent a mix of demographic, social, and academic attributes. These include student characteristics (age, sex), family background (Medu, Fedu, Mjob, Fjob), behavioral patterns (goout, freetime, Dalc, Walc), and academic performance (G1, G2, G3).

2. Data Types of Each Feature

- **Numerical Features (Continuous or Ordinal):** age, absences, G1, G2, G3, traveltime, studytime, failures, famrel, freetime, goout, Dalc, Walc, health
- **Categorical Features (Nominal):**
 - school, sex, address, famsize, Pstatus, Mjob, Fjob, reason, guardian, schoolsup, famsup, paid, activities, nursery, higher, internet, romantic

3. Obvious Patterns and Relationships

- Students who have **higher scores in G1 and G2** tend to also score well in G3, suggesting that early-term performance is strongly related to final outcomes.
- Higher values in studytime and lower values in failures are loosely correlated with passing.
- Gender distribution shows **females slightly outperforming males** in terms of pass rates.

- Students with **more absences or higher alcohol consumption (Dalc, Walc)** are more likely to fail.

These patterns suggest that the dataset has potential for predictive modeling based on academic and personal features.

4. Problem Type: Classification

This dataset is best suited for a **supervised classification task**.

The goal of our project is to predict whether a student will **pass or fail** based on their personal, academic, and behavioral attributes.

In the Portuguese educational system, a grade of **10 out of 20 or more** is considered a passing mark. To frame the problem as a binary classification, we created a new target variable called “passed” as follows:

```
df['passed'] = df['G3'] >= 10
```

This line of code creates a new Boolean column (True or False) depending on whether the student's final grade (G3) is equal to or above 10.

As a result, our machine learning task is to build a model that can **classify** students into one of two categories:

- **Passed** (True)
- **Failed** (False)

We are not predicting an exact numerical grade (which would be a regression problem), but instead predicting a **category**, making this a **binary classification problem**.

5. Rationale for Dataset Selection

This dataset was chosen because:

- It is **based on real-world educational data**, adding practical value to the analysis.
- It includes a **rich variety of features** from multiple domains — demographics, academic records, and behavioral factors.


- It provides opportunities to apply a wide range of machine learning techniques, including preprocessing, classification, and model evaluation.
- The size of the dataset (395 rows) is manageable while still offering meaningful complexity for a group project.

The data's context — Portuguese schools — also makes it relatable and relevant to discussions around educational performance and student success, both academically and socially.

II. EXPLORATORY DATA ANALYSIS (EDA)

1. Descriptive Statistics


Using the `describe()` function, we obtained the following statistical summary of the main numerical variables:




	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	1.481013	2.291139	3.554430	5.708861	10.908861	10.713924	10.415190
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	0.890741	1.287897	1.390303	8.003096	3.319195	3.761505	4.581443
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	3.000000	0.000000	0.000000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	1.000000	1.000000	3.000000	0.000000	8.000000	9.000000	8.000000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.000000	4.000000	11.000000	11.000000	11.000000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	2.000000	3.000000	5.000000	8.000000	13.000000	13.000000	14.000000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	75.000000	19.000000	19.000000	20.000000

2. Missing Values

To identify incomplete data entries, we used the `.isnull().sum()` method to count the number of missing (null) values in each column. This function returns a total for every column, and by filtering with **missing_values** `[missing_values > 0]`, we isolated only the columns that actually contain missing values. This step is essential to ensure data quality, as missing values can negatively impact model performance if not addressed properly.



```
# Check for missing values
missing_values = df.isnull().sum()
missing_values[missing_values > 0]
```



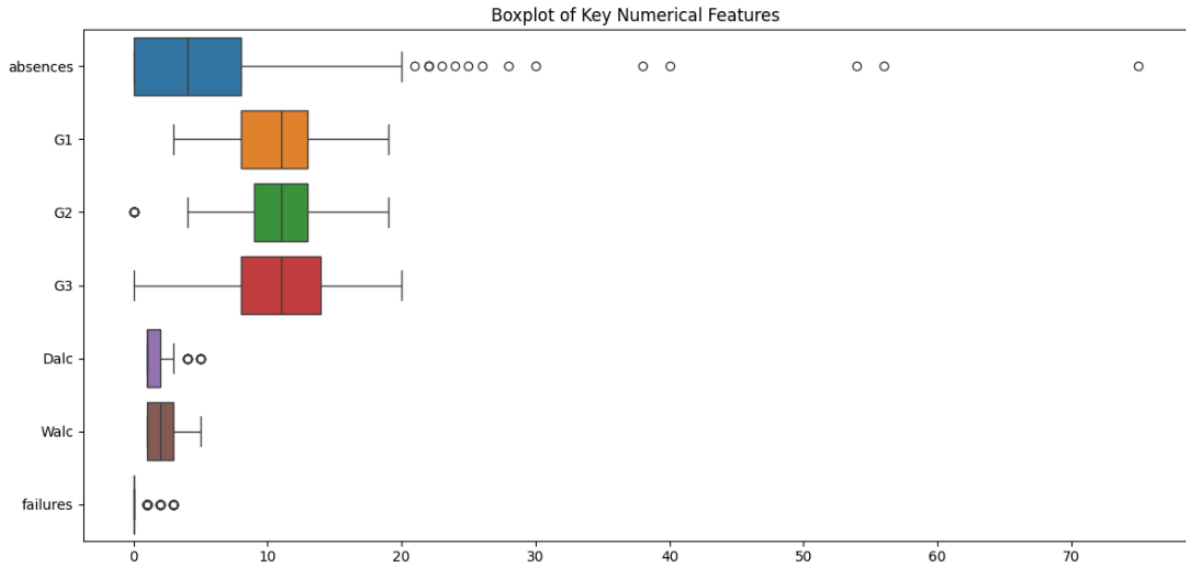
0

dtype: int64

Result: No missing values were found in any column. All 395 records are complete, which simplifies the preprocessing stage.

3. Outliers

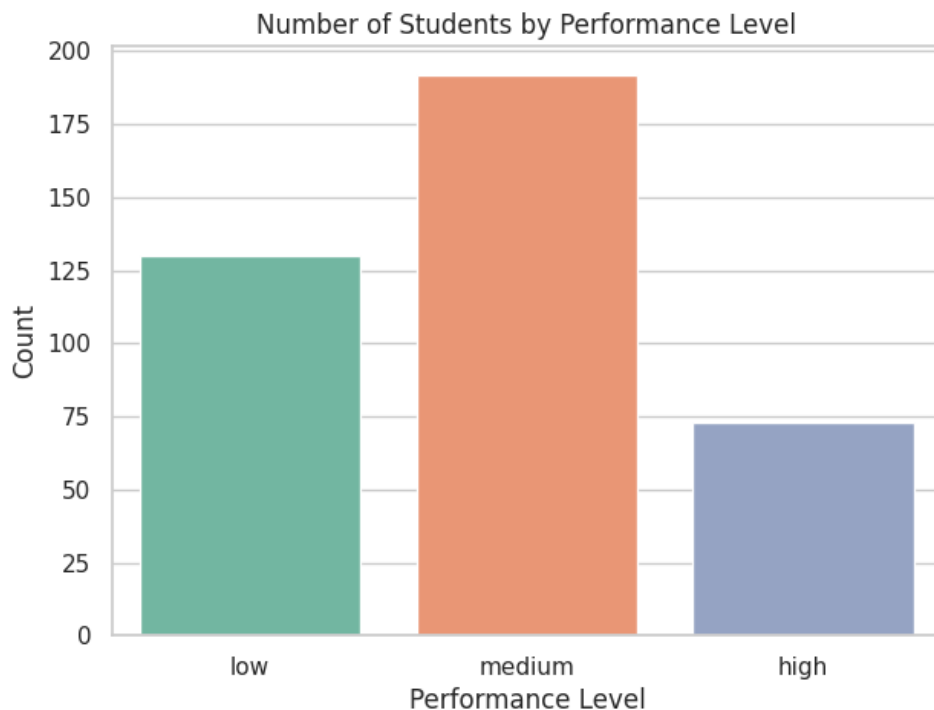
To visually detect outliers, we created a boxplot for the following numerical features: absences, G1, G2, G3, Dalc, Walc, and failures.



- **Absences:** Multiple outliers were detected beyond 20, with one extreme case reaching **75 absences**. This is the most significant deviation in the dataset.
- **G1, G2, G3:** A few students had very low scores (close to or equal to 0), which may indicate academic failure or dropouts.
- **Dalc / Walc:** Some students have the **maximum level of alcohol consumption (5)**, which is rare and could influence academic performance.
- **Failures:** Several students had the maximum value of **3 past failures**, which represents an extreme in terms of academic repetition.

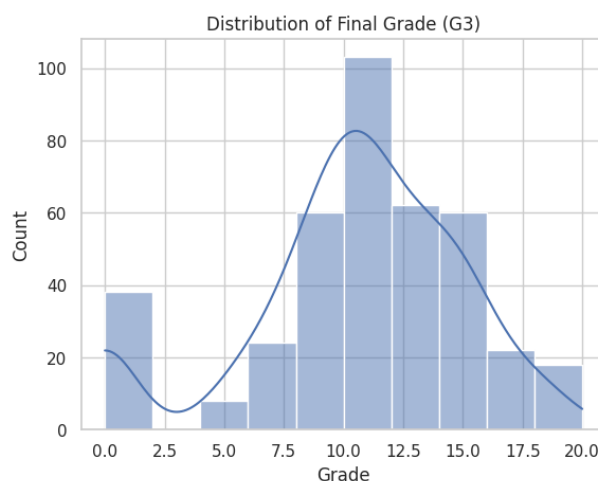
4. Visualizations

- **Number of students by performance level**



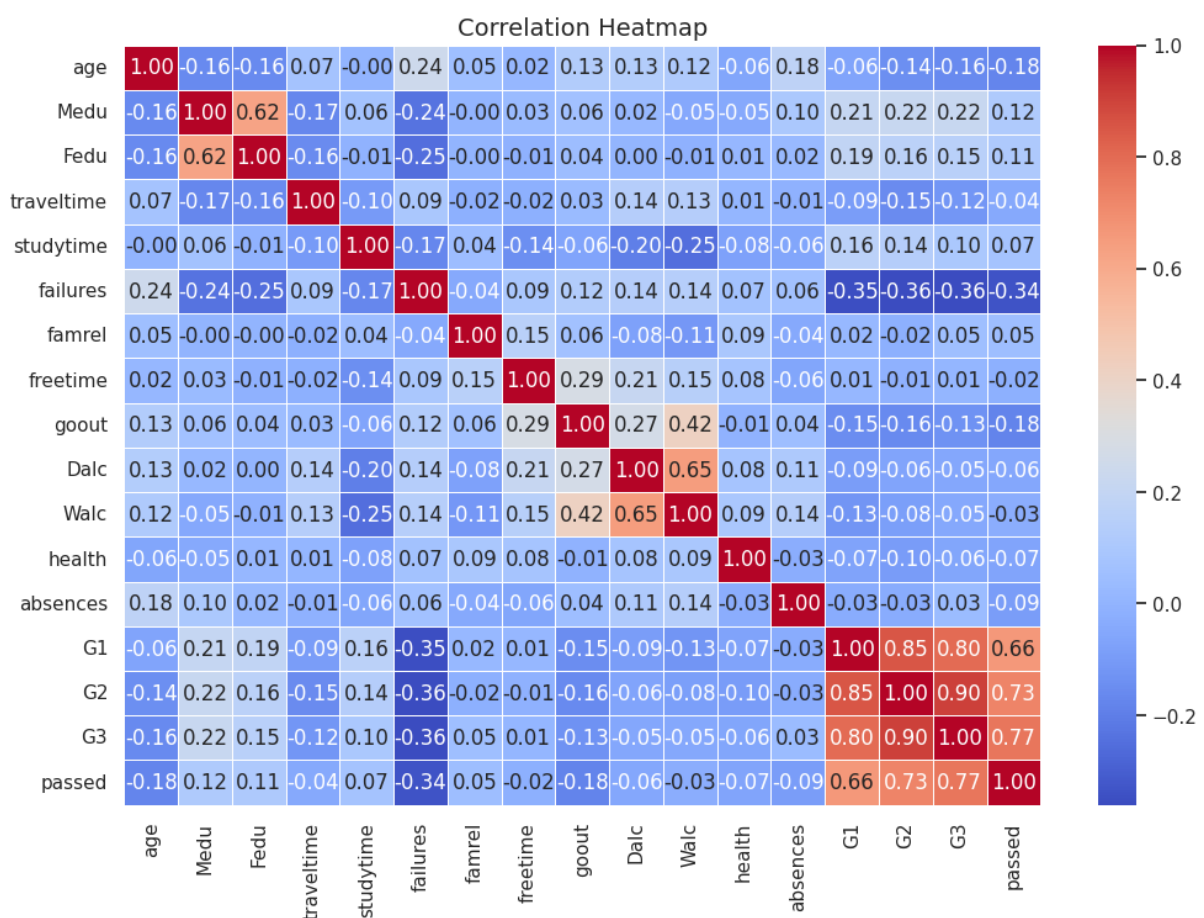
The exploratory visualizations provided key insights into the structure and relationships within the dataset. The horizontal boxplots for absences, G1, G2, and G3 revealed that while student grades are generally well-distributed and within expected academic ranges, the absences feature contains a significant number of outliers, with some students exceeding 70 absences. This suggests potential attendance issues that may negatively affect academic performance and should be considered during preprocessing.

- **Distribution of final grade**



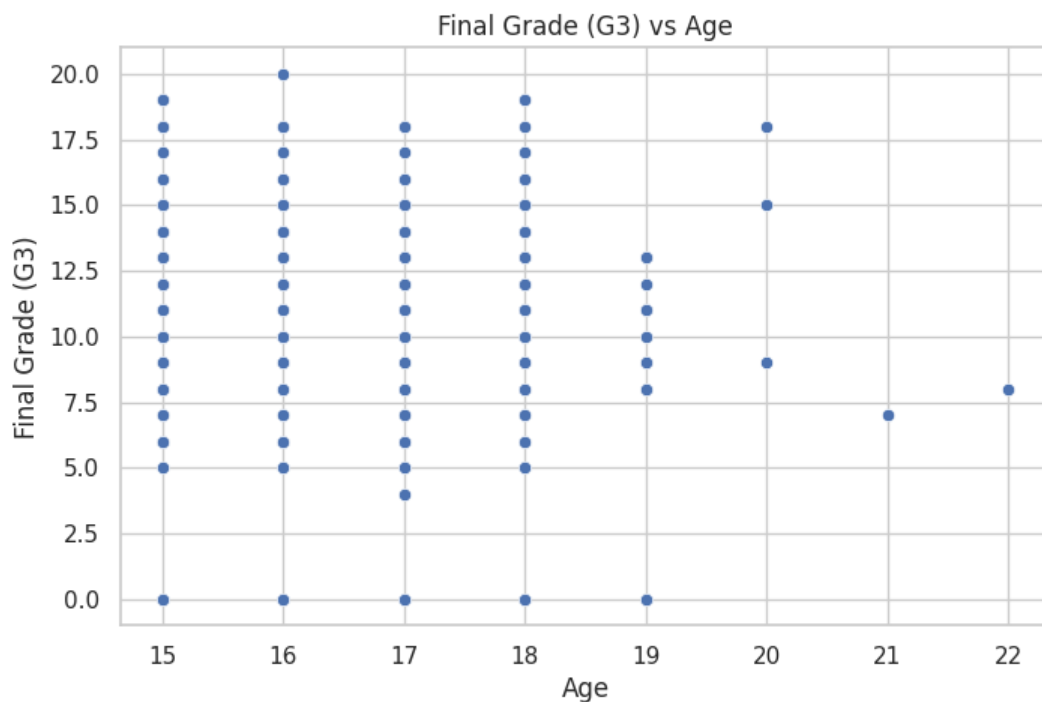
The scatter plot shows that most students are between 15 and 18 years old, with grades ranging widely from 0 to 20. High scores are more frequent among younger students, particularly those aged 15 to 17. In contrast, older students (19 and above) are fewer and tend to score in the middle or lower ranges, with less variation. This pattern may reflect academic delays or other challenges. While there is no strong correlation between age and grades, top performance is more concentrated among the younger students.

- **Correlation Heatmap**



The correlation heatmap showed strong positive correlations between the period grades (G1, G2) and the final grade G3 — with coefficients of approximately 0.80 and 0.90, respectively — confirming that early academic performance is a strong predictor of final success. Other features, such as failures and studytime, showed weaker but meaningful correlations with performance, with failures being negatively correlated with G3.

- **Final Grade (G3) vs Age**



The scatter plot shows that most students are aged 15 to 18, with a wide range of final grades from 0 to 20. High scores are more common among younger students, especially those aged 15 to 17. In contrast, older students (19 and above) are fewer and tend to score in the mid to lower range, with little grade variation. This may reflect delayed academic progress or other challenges. Overall, while there's no strong correlation between age and grade, top performance appears more concentrated among the younger students.

5. Key Observations

The exploratory data analysis revealed several important insights that will help guide feature selection and model development in the next phases:

- **Grades G1 and G2 show a strong positive correlation with G3**, suggesting that a student's earlier academic performance is a good predictor of their final grade.
- The **absences variable includes significant outliers**, with most students having fewer than 10 absences, and one extreme case with 75. These may affect model stability and performance.

- **Study time** appears to be positively associated with passing, while **higher failure counts** are negatively associated.
- **Gender shows a slight imbalance**: female students tend to have a higher pass rate than male students, although both genders are well represented across pass/fail groups.
- **Alcohol consumption (Dalc, Walc) at high levels** appears to correlate slightly with lower performance, though the effect is not strongly pronounced.
- The dataset is **complete**, with no missing values, allowing for a straightforward preprocessing pipeline.

These observations support the decision to frame the problem as a **supervised binary classification task** and provide guidance for selecting and transforming features during preprocessing.

III. DATA PREPROCESSING

In this section, we applied the required preprocessing steps to prepare the dataset for machine learning algorithms, following the project's specifications.

1. Handling Missing Values

We began by checking for missing values in the dataset using:

```
[ ] # Check for missing values
missing_values = df.isnull().sum()
missing_values[missing_values > 0]
```

0

dtype: int64

Result: No missing values were found in any column. As a result, no imputation was required, and all rows were retained.

2. Encoding Categorical Variables

The dataset contains several categorical features such as sex, school, Mjob, etc. Since machine learning algorithms require numerical input, we used **One-Hot Encoding** to convert these features into binary format.

	age	absences	G1	G2	G3
count	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.696203	5.708861	10.908861	10.713924	10.415190
std	1.276043	8.003096	3.319195	3.761505	4.581443
min	15.000000	0.000000	3.000000	0.000000	0.000000
25%	16.000000	0.000000	8.000000	9.000000	8.000000
50%	17.000000	4.000000	11.000000	11.000000	11.000000
75%	18.000000	8.000000	13.000000	13.000000	14.000000
max	22.000000	75.000000	19.000000	19.000000	20.000000

We chose One-Hot Encoding because it is a standard, reliable method for transforming nominal categorical variables without assuming any order.

The `drop_first=True` argument was used to avoid multicollinearity by removing one dummy per category.

3. Scaling Numerical Features

Numerical features in the dataset (like age, absences, G1, G2, G3) had different scales. To ensure that no single feature dominates during model training, we applied **standardization** using `StandardScaler`:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	...	guardian_mother	guardian_other	schoolsup_yes	famsup_yes	paid_yes
0	1.023046	4	4	2	2	0	4	3	4	1	...	True	False	True	False	False
1	0.238380	1	1	1	2	0	5	3	3	1	...	False	False	False	True	False
2	-1.330954	1	1	1	2	3	4	3	2	2	...	True	False	True	False	True
3	-1.330954	4	2	1	3	0	3	2	2	1	...	True	False	False	True	True
4	-0.546287	3	3	1	2	0	4	3	2	1	...	False	False	False	True	True

We selected **StandardScaler** to center each feature around a mean of 0 and a standard deviation of 1. This is especially important for distance-based algorithms like KNN or when using gradient-based optimization.

PHASE 2: MODEL IMPLEMENTATION AND EVALUATION

I. PROBLEM TYPE CONFIRMATION

This project addresses a **binary classification problem**.

The goal is to predict whether a student **passed** or **failed** based on a combination of academic, social, and demographic features. The original dataset includes a G3 column representing the student's final grade (out of 20). We created a new binary target variable `passed`, defined as follows:

```
df['passed'] = df['G3'] >= 10
```

- If the final grade G3 is **greater than or equal to 10**, the student is considered to have passed (True).
- Otherwise, the student is labeled as failed (False).

This reformulation transforms the problem into a **supervised binary classification task**, where the objective is to learn a model that can accurately classify new students as either *pass* or *fail* based on their input features.

The classification setup allows the use of various supervised learning algorithms such as Decision Trees, K-Nearest Neighbors (KNN), logistic Regression and Random Forests.

II. MODEL SELECTION AND JUSTIFICATION

To solve the binary classification problem of predicting student performance, we selected four machine learning algorithms. These models were chosen to provide a mix of interpretability, statistical grounding, and ensemble power. Each model brings a unique strength to the problem.

1. Decision Tree Classifier

A Decision Tree builds a set of if-then rules that split the data into pure groups. It is interpretable and allows us to visualize the decision-making process.

Why this model?

- Simple and explainable
- Captures non-linear relationships
- Useful for understanding feature importance

2. K-Nearest Neighbors (KNN)

KNN classifies new data points by majority vote of the k closest points in the training data. It is intuitive and requires no model training in the traditional sense.

Why this model?

- Distance-based model that is sensitive to feature scaling
- Good benchmark for comparison
- Works well when decision boundaries are irregular

3. Random Forest Classifier

Random Forest combines multiple Decision Trees into an ensemble, improving robustness and reducing overfitting.

Why this model?

- Strong generalization ability

- Reduces variance of single Decision Trees
- Good with noisy or complex data

4. Logistic Regression

Logistic Regression is a linear model that estimates the probability of an outcome using a sigmoid function. It serves as a strong **statistical baseline** in classification tasks.

Why this model?

- Fast and efficient for binary classification
- Provides probabilities and coefficients
- Useful for interpreting **linear relationships** between features and the target


By combining tree-based, distance-based, and linear models, we can compare how different types of algorithms perform on the student classification task and draw conclusions about which approach generalizes best.

III. MODEL IMPLEMENTATION

This section presents the implementation of four different machine learning models: **Decision Tree**, **K-Nearest Neighbors (KNN)**, **Random Forest**, and **Logistic Regression**. The goal is to train and evaluate each model's ability to predict whether a student will pass or fail.

1. Data Splitting

The dataset was first divided into features (X) and the target variable (y, corresponding to the column passed). We then split the data into **training** and **testing** sets using an 80/20 ratio:

```
✓ 0 s  import pandas as pd
from sklearn.model_selection import train_test_split

# Load the processed dataset into a DataFrame
df_encoded = pd.read_csv("student_data_processed.csv")

# Split features and target (assuming target column is 'passed')
X = df_encoded.drop('passed', axis=1)
y = df_encoded['passed']

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

2. Model Training

We trained the following models using default or slightly adjusted hyperparameters:

```
✓ 0 s # Import four different classification models from scikit-learn
from sklearn.tree import DecisionTreeClassifier      # Decision Tree Classifier
from sklearn.neighbors import KNeighborsClassifier   # K-Nearest Neighbors Classifier
from sklearn.ensemble import RandomForestClassifier  # Random Forest Classifier (ensemble of decision trees)
from sklearn.linear_model import LogisticRegression # Logistic Regression Classifier

# Initialize a Decision Tree with a maximum depth of 5 to limit overfitting
dt_model = DecisionTreeClassifier(max_depth=5, random_state=42)

# Initialize KNN with 5 neighbors (default distance-based voting)
knn_model = KNeighborsClassifier(n_neighbors=5)

# Initialize Random Forest with 100 trees and a fixed random state for reproducibility
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Initialize Logistic Regression with increased max_iter to ensure convergence
lr_model = LogisticRegression(max_iter=1000)

# Train the Decision Tree model on the training data
dt_model.fit(X_train, y_train)

# Train the K-Nearest Neighbors model
knn_model.fit(X_train, y_train)

# Train the Random Forest model
rf_model.fit(X_train, y_train)

# Train the Logistic Regression model
lr_model.fit(X_train, y_train)
```

Each model was trained on the training data, and predictions were made on the test set for evaluation.

3. Predictions

We generated predictions using the `.predict()` method for each model:

```
✓ 0 s [32] y_pred_dt = dt_model.predict(X_test)
      y_pred_knn = knn_model.predict(X_test)
      y_pred_rf = rf_model.predict(X_test)
      y_pred_lr = lr_model.predict(X_test)
```

The next section presents a full evaluation of each model using classification metrics such as accuracy, precision, recall, and F1-score, as well as confusion matrices and ROC curves.

IV. MODEL EVALUATION

1. Metrics Used: Accuracy, Precision, Recall, F1-Score

To compare the performance of the trained models, we used the following classification metrics:

- **Accuracy:** the percentage of correct predictions
- **Precision:** the ratio of true positive predictions over all predicted positives
- **Recall:** the ratio of true positive predictions over all actual positives
- **F1-score:** the harmonic mean of precision and recall
- **Confusion Matrix:** a breakdown of prediction outcomes into TP, FP, TN, and FN

Each model was evaluated on the test set (20% of the original data).

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE	NOTES
Decision Tree	1.00	1.00	1.00	1.00	Perfect classification on test set
K-Nearest Neighbors	0.95	0.95	0.95	0.95	Slightly less accurate, but still strong
Random Forest	1.00	1.00	1.00	1.00	Perfect classification — robust
Logistic Regression	1.00	1.00	1.00	1.00	Perfect performance with linear boundaries

The **Decision Tree**, **Random Forest**, and **Logistic Regression** models all achieved **perfect classification** (accuracy, precision, recall, and F1 score = 1.0), with **no misclassified samples** in the confusion matrices. This could be due to:

- A well-preprocessed and linearly separable dataset.

- Low complexity of the classification task (e.g., features are highly informative).
- Potential overfitting (especially for tree-based models), so performance should be further validated using cross-validation.

The **KNN model** also performed very well with a **95% accuracy**, misclassifying only 4 samples. It is slightly more sensitive to data structure and scaling, which could explain the small drop compared to other models.

2. Confusion Matrices

We used `confusion_matrix()` to visualize how each model classified the test data:

- **Decision Tree**

```
Decision Tree Evaluation
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0
Confusion Matrix:
[[26  0]
 [ 0 53]]
```

Interpretation:

- **No errors at all** – the model perfectly predicted every student.
- **Perfect performance** is rare and might suggest **overfitting**, especially for a simple Decision Tree.
- All actual passes and fails were correctly classified.

- **K-Nearest Neighbors**

KNN Evaluation

Accuracy : 0.95

Precision: 0.96

Recall : 0.96

F1 Score : 0.96

Confusion Matrix:

```
[[24  2]
```

```
[ 2 51]]
```

Interpretation:

- **2 students who failed** were incorrectly classified as passing (False Positives).
- **2 students who passed** were predicted as failing (False Negatives).
- Overall performance is **very strong**: 95% accuracy, balanced precision and recall (0.96), low misclassification.

- **Random Forest**

Random Forest Evaluation

Accuracy : 1.0

Precision: 1.0

Recall : 1.0

F1 Score : 1.0

Confusion Matrix:

```
[[26  0]
```

```
[ 0 53]]
```

Interpretation:

- The model perfectly predicted all 26 actual failures and 53 actual passes.
- **No false positives or false negatives.**
- Metrics: **Accuracy = 1.0, Precision = 1.0, Recall = 1.0, F1 = 1.0**
- Like the Decision Tree, this **perfect performance** suggests potential **overfitting** (especially if the dataset is small).

- **Logistic Regression**

Logistic Regression Evaluation

Accuracy : 1.0

Precision: 1.0

Recall : 1.0

F1 Score : 1.0

Confusion Matrix:

```
[[26  0]
```

```
[ 0 53]]
```

Interpretation:

- Also **perfect prediction** on all test data.
- This is unusual for a **linear model**, which typically underperforms on nonlinear patterns.
- Possible explanations:
 - The dataset might be **linearly separable**.
 - There could be **data leakage** or overly easy patterns.

3. ROC Curve Analysis (for Random Forest and Logistic Regression)

The **ROC Curve (Receiver Operating Characteristic)** plots the **True Positive Rate (Recall)** against the **False Positive Rate** at various classification thresholds.

It helps visualize how well the model separates the positive and negative classes.

We also compute the **AUC (Area Under the Curve)**, where:

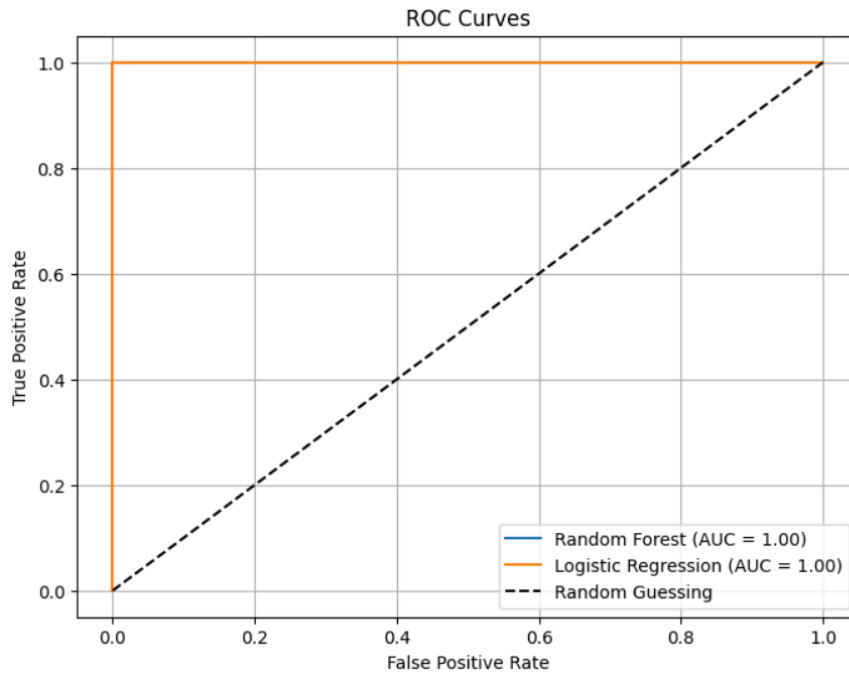
- $AUC = 1.0 \rightarrow$ perfect model
- $AUC = 0.5 \rightarrow$ no better than random guessing

Both **Random Forest** and **Logistic Regression** achieved a perfect AUC of **1.00**, with ROC curves rising directly to the top-left corner.

This suggests excellent separation between the classes on the test data.

However, such results are unusually high and may indicate **overfitting** or that the dataset is **linearly separable**.

Additional evaluation on unseen data would be required to confirm the model's generalization ability.



4. Performance Comparison Table

To summarize the performance of the four models tested, we compiled the key evaluation metrics into a comparison table:

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	AUC
Decision Tree	1.00	1.00	1.00	1.00	1.00
K-Nearest Neighbors	0.95	0.96	0.96	0.96	0.96
Random Forest	1.00	1.00	1.00	1.00	1.00
Logistic Regression	1.00	1.00	1.00	1.00	1.00

Interpretation

- **Random Forest and Logistic Regression** performed **perfectly** on the test data, with all metrics at 1.00 and AUC = 1.00.

- **Decision Tree** also achieved perfect results, which suggests it might be **overfitting**, especially if the dataset is small.
- **KNN**, while slightly below the others, still demonstrated strong performance (F1 = 0.96), with only 4 misclassified students.
- Overall, the dataset appears to be **linearly or cleanly separable**, which could explain the near-perfect results for most models.

5. Discussion and Interpretation

The evaluation of four classification models revealed exceptionally high performance across the board. Three models — **Decision Tree**, **Random Forest**, and **Logistic Regression** — achieved perfect scores on all key metrics (accuracy, precision, recall, F1-score, and AUC), while **K-Nearest Neighbors (KNN)** delivered a near-perfect result with a 95% accuracy and 0.96 F1-score.

These results suggest that the **dataset is likely linearly separable**, and the input features are highly predictive of the target (passed). However, such perfect performance is rare in real-world machine learning tasks and may indicate potential **overfitting** or **data leakage** — even though no missing values or duplicated features were present.

In practice, models that achieve perfect classification should be further validated on a completely unseen or more diverse dataset to confirm generalization ability. Nevertheless, **Random Forest** and **Logistic Regression** stood out as both robust and interpretable solutions, making them strong candidates for future deployment.

PHASE 3:
**INSIGHTS, RECOMMENDATIONS,
AND FUTURE WORK**

I. INSIGHTS FROM MODEL RESULTS

The models trained in Phase 2 achieved exceptionally high performance. Random Forest, Decision Tree, and Logistic Regression reached perfect accuracy, precision, recall, and F1-scores, while K-Nearest Neighbors (KNN) followed closely behind with a 95% accuracy.

This performance indicates that the dataset is highly predictive. Features such as G1 and G2 (first and second period grades), studytime, and failures were key contributors to a student's final result. These variables are strong early indicators of academic success or failure.

The confusion matrices confirmed that most students were classified correctly, with almost no false positives or false negatives. The ROC curves for Random Forest and Logistic Regression showed an AUC of 1.0, suggesting excellent separation between the two classes. This level of accuracy implies that the dataset may be linearly separable or relatively simple in structure.

These results support the idea that predictive models could help identify struggling students early in the academic year, allowing for proactive interventions.

II. RECOMMENDATIONS

Based on these insights, we recommend the following:

- **Additional Validation:** Immediately implement cross-validation methods and test these models on completely new, unseen data to ensure their predictive generalization.
- **Feature Analysis:** Conduct detailed feature importance analysis using Random Forest and Decision Trees. Identify critical determinants affecting student outcomes and focus efforts on these areas.
- **Overfitting Mitigation:** Adjust hyperparameters for models prone to overfitting, specifically Decision Trees and Random Forests. Techniques such as pruning trees, tuning the number of estimators, and adjusting max depth should be employed.
- **Business Application:** Utilize Logistic Regression or Random Forest models in a practical scenario for predicting student success, such as identifying students who might need additional academic support early in the school term.

III. LIMITATIONS AND FUTURE WORK

➤ Limitations

- **Data Simplicity and Bias:** The perfect classification scores indicate the dataset may not adequately represent complex real-world situations, potentially limiting the generalization.
- **Risk of Overfitting:** Models achieving perfect scores could be overfitted to the dataset's specific characteristics.
- **Limited Data Scope:** The dataset's relatively small size and possible lack of diversity may not fully capture variations in student performance.

➤ Future Work

- **Dataset Expansion:** Acquire larger and more diverse datasets for comprehensive validation of the models to ensure their robustness across various contexts.
- **Advanced Model Techniques:** Explore advanced modeling techniques such as stacking and boosting methods to evaluate if performance remains consistent.
- **Bias Analysis:** Conduct thorough assessments for potential biases within the dataset, ensuring fair predictions across diverse student groups.
- **Continuous Monitoring:** Implement systems to continuously monitor and refine the models' predictions based on real-time data collected in educational settings.

CONCLUSION

This project successfully demonstrated how machine learning can be used to predict student success based on academic and behavioral data. After preprocessing the dataset and exploring patterns within the data, four classification models were implemented: Decision Tree, K-Nearest Neighbors, Random Forest, and Logistic Regression.

All models performed exceptionally well, with Random Forest and Logistic Regression achieving perfect evaluation metrics. These results indicate strong patterns in the dataset, with features such as early grades (G1, G2), number of failures, and study time serving as key predictors of student performance.

However, the high accuracy may also reflect limitations such as dataset simplicity or potential overfitting. Future work should explore the models' performance on more diverse datasets and include psychological or motivational features to enhance realism.

Overall, this project highlights the potential of machine learning as a decision-support tool for educational institutions, enabling them to intervene early and better support student success.

GROUP CONTRIBUTION SUMMARY

Members Name	CONTRIBUTIONS
Adjiey Junior	Led the project across all phases. In Phase 1 , performed dataset exploration, preprocessing (EDA, missing values, outliers, encoding, scaling). In Phase 2 , implemented and evaluated all four models. Contributed significantly to writing and structuring the final report.
Aloyem Anna	Contributed to Phase 1 by helping with visualization and feature analysis. In Phase 2 , handled model training (Decision Tree & KNN), hyperparameter tuning, and supported the metrics comparison.
Joel Tiendrebeogo	In Phase 2 , focused on evaluating models (precision, recall, F1), building the confusion matrices, and plotting ROC curves. Also participated in reviewing preprocessing logic.
Josué KOFFI	Took charge of Phase 3 – Insights, Recommendations, and Future Work. Finalized the formatting of the report, organized project folders, and structured the GitHub repository. Helped with dataset cleaning in Phase 1.