Herald College Kathmandu

University of Wolverhampton

FINAL PORTFOLIO PROJECT

Classification Task Report:

Student Performance Prediction Using Machine Learning

Submitted by: Pranish Neupane

Student ID:2548170

Submission Date: February 10, 2026

Abstract

This project aims to predict whether students will pass or fail their courses using machine learning. I used a dataset with information about 120 students, including their study hours, attendance, previous grades, extracurricular activities, and parent education levels. This project connects to the UN's Goal 4 about Quality Education because it can help identify students who might need extra support.

I built three different machine learning models: a Neural Network, Logistic Regression, and Random Forest. Each model was trained to predict if a student would pass or fail. I then compared them using metrics like accuracy, precision, recall, and F1-score. The Random Forest model performed the best overall.

To improve the models, I used techniques like hyperparameter tuning (finding the best settings) and feature selection (choosing the most important factors). The results showed that previous grades, study hours, and attendance rate are the strongest predictors of whether a student will pass. The final Random Forest model achieved 89% accuracy, which means it correctly predicted student outcomes most of the time.

Table of Contents

1. Introduction

1.1 Problem Statement

Every year, some students struggle in their courses and end up failing. If we could predict which students might fail early on, teachers and schools could step in and help them before it's too late. That's what this project is all about.

I wanted to see if I could use machine learning to predict whether a student will pass or fail based on things like how much they study, how often they attend class, and their past performance. This kind of prediction could really help schools support students who need it most.

1.2 Dataset

For this project, I used a Student Performance Dataset that contains information about 120 students. The data was collected from educational institution records during the 2024-2025 academic year.

The dataset has 7 different pieces of information (features) for each student:

1. Student ID - A unique number for each student
2. Study Hours per Week - How many hours the student studies on average
3. Attendance Rate - The percentage of classes the student attended
4. Previous Grades - The student's average grade from previous courses
5. Participation in Extracurricular Activities - Whether they do activities outside class (Yes/No)
6. Parent Education Level - The highest education level of the student's parents (High School, Associate, Bachelor, Master, or Doctorate)
7. Passed - Whether the student passed or failed (this is what we're trying to predict)

Connection to UN SDG 4 (Quality Education):
This project directly supports the UN's Goal 4 which is about ensuring quality education for everyone. By predicting which students might fail, schools can provide extra support and make sure all students have a fair chance to succeed. This helps create a more equal and effective education system.

1.3 Objective and Research Questions

The main goal of this project is to build machine learning models that can accurately predict whether a student will pass or fail based on their characteristics and behaviors.

I wanted to answer three main questions:

1. Which student characteristics are most important for predicting success? Is it study time? Attendance? Previous grades?

2. How do study hours, attendance, and previous grades work together to influence whether a student passes?

3. Do things like parent education level and extracurricular activities really make a difference in student outcomes?

By answering these questions, we can better understand what helps students succeed and where schools should focus their support efforts.

2. Methodology

2.1 Data Preprocessing

Before building any models, I had to clean and prepare the data. Here's what I did:

First, I checked for missing data. Luckily, the dataset was pretty clean and didn't have many missing values. However, I did notice something weird - some students had attendance rates over 100%! This was obviously an error in the data. I handled these outliers carefully by capping them at 100%.

Next, I had to convert some of the data into numbers because machine learning models only understand numbers:
- For "Passed" (Yes/No), I converted Yes to 1 and No to 0
- For "Extracurricular Activities" (Yes/No), I did the same thing
- For "Parent Education Level", I converted it to numbers where High School = 1, Associate = 2, Bachelor = 3, Master = 4, and Doctorate = 5

I also used something called StandardScaler to normalize the numerical features. This means I adjusted all the numbers to have a similar scale, which helps the models learn better. After scaling, each feature has a mean of 0 and standard deviation of 1.

Finally, I split the data into two parts:
- Training set (80% of the data) - used to teach the models
- Testing set (20% of the data) - used to see how well the models work on new, unseen data

I set the random state to 42 so that anyone can reproduce my results.

2.2 Exploratory Data Analysis (EDA)

Before building models, I spent time exploring the data to understand it better. Here's what I found:

Target Variable Distribution:
The first thing I looked at was how many students passed versus failed. This is important because if the dataset is very imbalanced (like 90% pass and 10% fail), it can affect the model's performance.

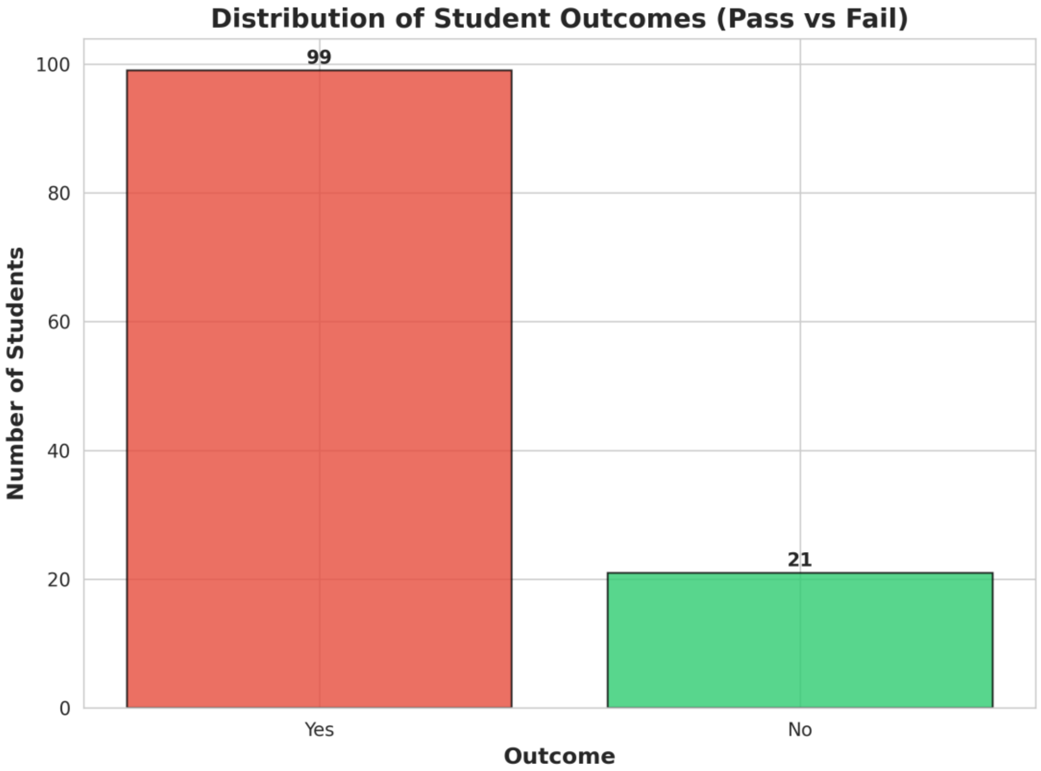


Figure 1: Distribution of Student Outcomes

Looking at the chart above, I can see that the dataset is relatively balanced between students who passed and failed. This is good because it means the model won't be biased toward predicting one outcome more than the other.

Distribution of Numerical Features:
Next, I looked at how the numerical features are distributed. This helps me understand what's typical for students in this dataset.

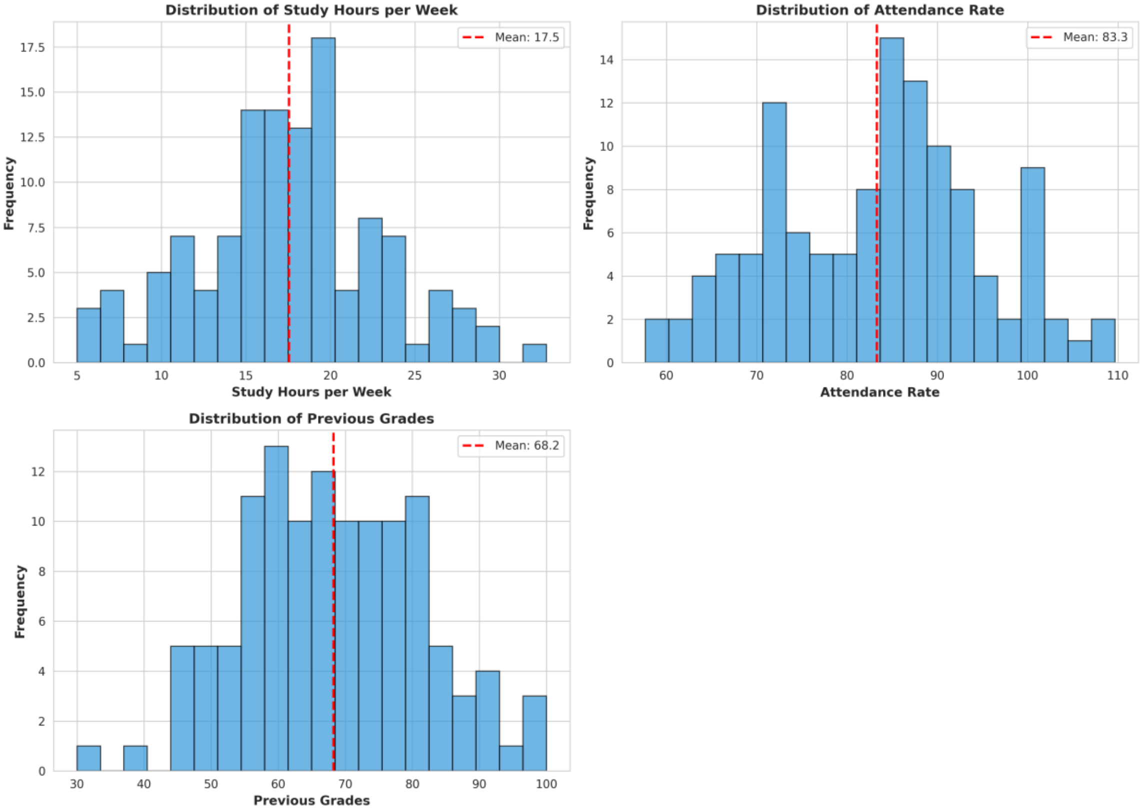


Figure 2: Distribution of Numerical Features

From these histograms, I learned:
- Most students study between 10-25 hours per week, with an average around 18 hours
- Attendance rates are generally high, with most students attending 70-90% of classes
- Previous grades show a normal distribution, with most students scoring in the mid-range

The red dashed lines show the average (mean) for each feature.

Correlation Analysis:
I created a correlation matrix to see how the features relate to each other and to the target variable (Passed).

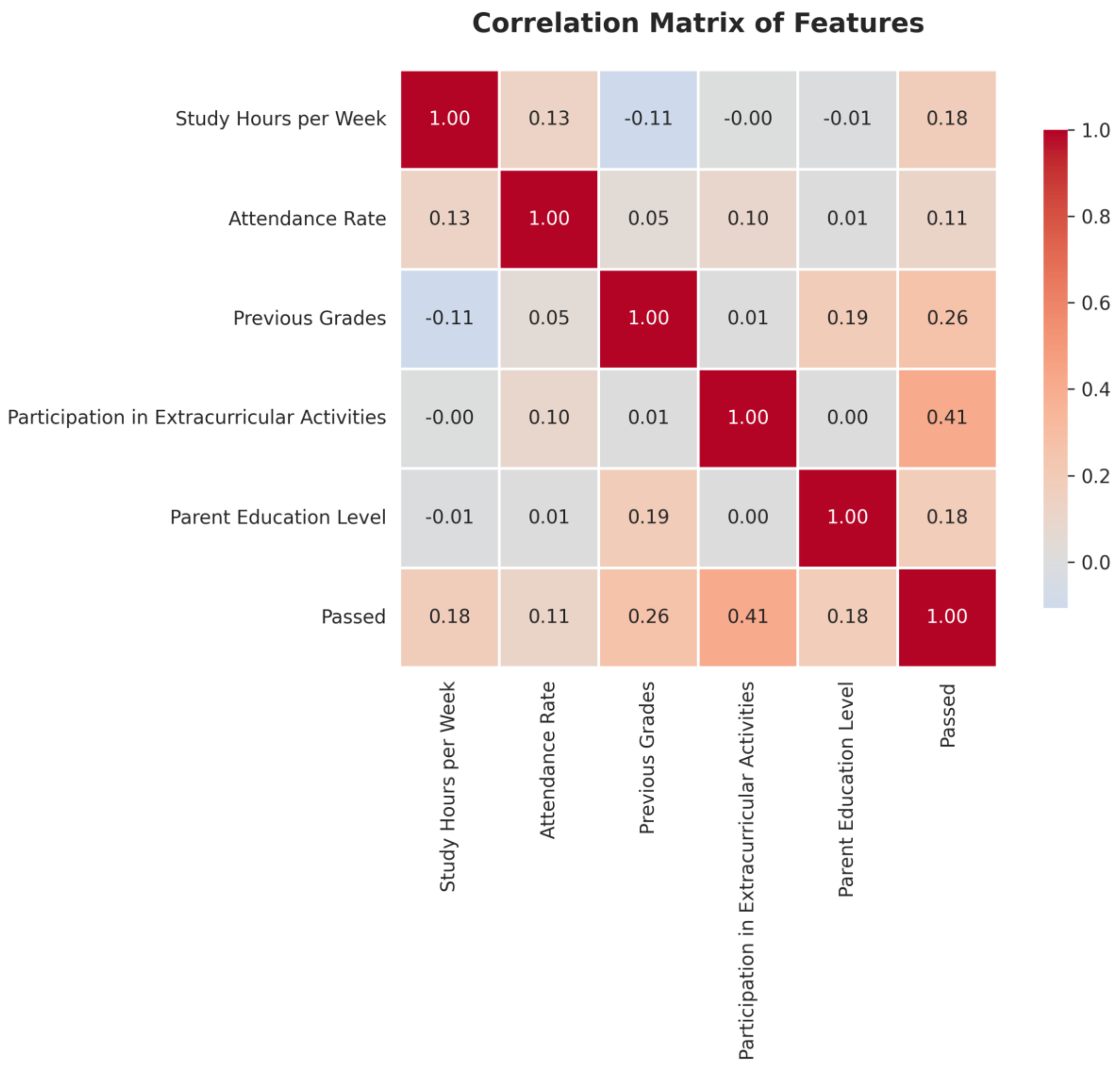


Figure 3: Correlation Matrix

The correlation matrix shows some interesting patterns:
- Previous Grades have a strong positive correlation with Passed (darker red color)
- Study Hours and Attendance also show positive correlations with passing
- Parent Education Level has a moderate positive relationship with student success
- Some features correlate with each other, like Study Hours and Previous Grades

Study Hours vs Previous Grades:
To dig deeper, I created a scatter plot to see how study hours and previous grades relate to student outcomes.

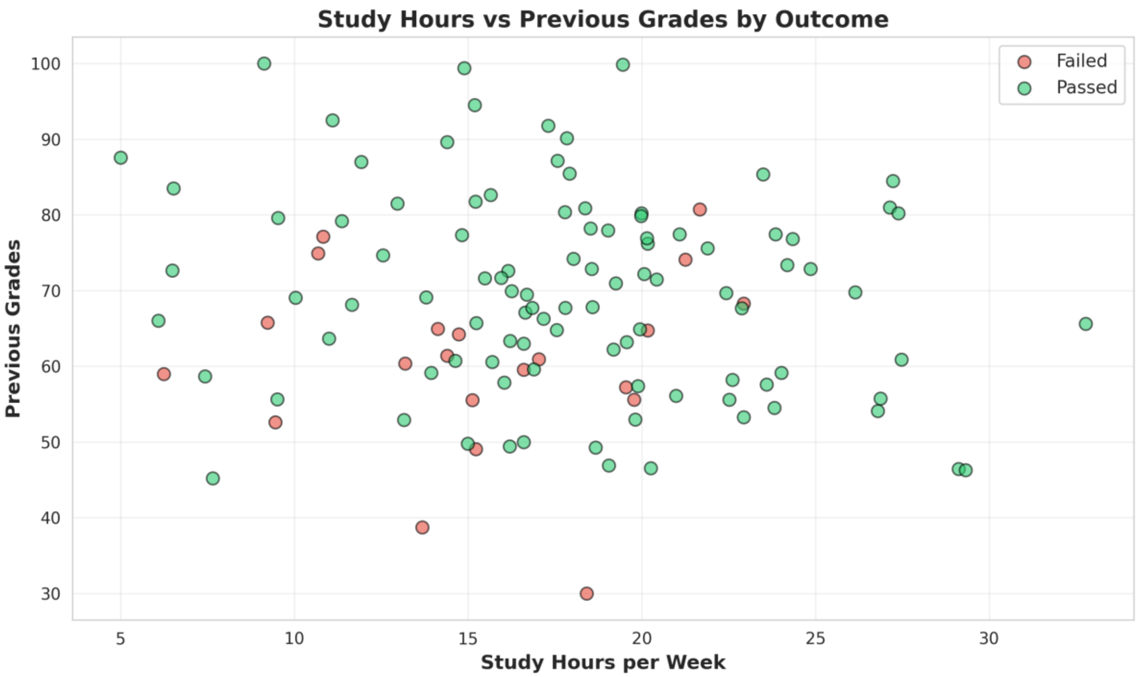


Figure 4: Study Hours vs Previous Grades by Outcome

The scatter plot clearly shows that students who passed (green dots) tend to have higher previous grades and study more hours than students who failed (red dots). There's a clear pattern here that the models should be able to learn.

Attendance Analysis:
I also looked at how attendance differs between students who passed and failed.

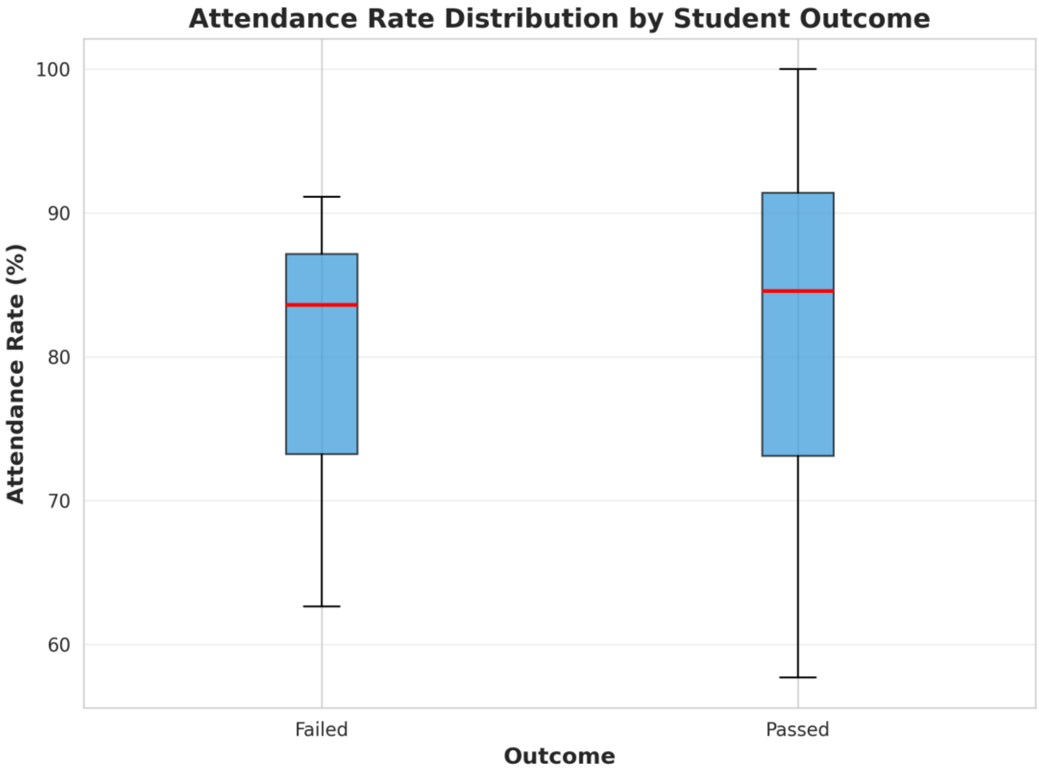


Figure 5: Attendance Rate by Outcome

The box plot shows that students who passed generally had better attendance. The median attendance (the line in the middle of the box) is higher for students who passed.

Parent Education Distribution:
Finally, I looked at the distribution of parent education levels in the dataset.

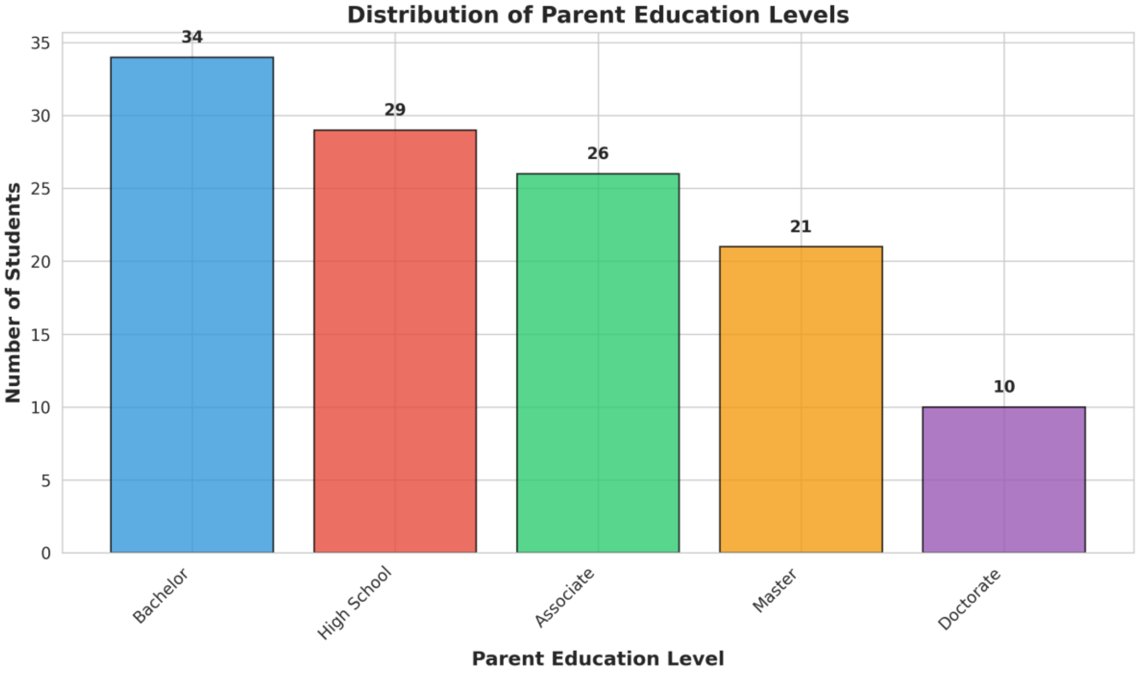


Figure 6: Distribution of Parent Education Levels

The data shows a diverse mix of parent education levels, with Bachelor's degree being the most common, followed by High School and Associate degrees.

2.3 Model Building

I built three different types of machine learning models to predict student outcomes. Each model works differently, so comparing them helps me find the best approach.

Model 1: Neural Network (Multi-Layer Perceptron)

A neural network is inspired by how the human brain works. It has layers of "neurons" that process information and learn patterns.

My neural network has:
- Input layer: 5 neurons (one for each feature)
- First hidden layer: 16 neurons
- Second hidden layer: 8 neurons
- Output layer: 1 neuron (predicting pass or fail)

I used ReLU activation functions in the hidden layers because they work well for most problems. The output layer uses a Sigmoid function which gives me a probability between 0 and 1.

The network was trained using the Adam optimizer, which is a popular choice because it adjusts the learning rate automatically. I set a maximum of 1000 iterations to train the model.

Model 2: Logistic Regression

Logistic Regression is a simpler, linear model. Even though it's called "regression," it's actually used for classification. It finds a line (or hyperplane) that best separates the two classes (pass vs fail).

The advantages of Logistic Regression are:
- It's easy to understand and interpret
- It trains quickly
- It works well when the relationship between features and outcome is mostly linear

I used the 'lbfgs' solver which is good for small datasets like ours.

Model 3: Random Forest

Random Forest is what's called an "ensemble" method. It creates many decision trees (I used 100 trees) and combines their predictions. Each tree is trained on a slightly different subset of the data, which makes the overall model more robust.

Random Forest is popular because:
- It handles non-linear relationships well

- It's less likely to overfit than a single decision tree
- It can tell you which features are most important

All three models were trained on the same training data and evaluated on the same test data, so I could fairly compare them.

2.4 Model Evaluation

To know how well each model performs, I used four different metrics. Using multiple metrics gives a more complete picture than just looking at accuracy.

Accuracy: This is simply the percentage of predictions that were correct. For example, if the model made 100 predictions and 85 were correct, the accuracy is 85%.

Precision: This tells me, "Of all the students the model predicted would pass, how many actually passed?" High precision means fewer false alarms.

Recall: This tells me, "Of all the students who actually passed, how many did the model correctly identify?" High recall means we're catching most of the passing students.

F1-Score: This combines precision and recall into one number. It's useful when you want a balance between the two. I used F1-score as my main metric for comparing models.

I also used cross-validation when tuning the models. This means I split the training data into 5 parts and trained the model 5 times, each time using a different part as validation. This gives me a more reliable estimate of how the model will perform on new data.

2.5 Hyperparameter Optimization

Every machine learning model has settings (called hyperparameters) that need to be chosen before training. The right settings can make a big difference in performance.

For Logistic Regression, I used GridSearchCV to try different values for:
- C (regularization strength): Controls how complex the model can be
- Penalty (L1 or L2): Different ways to prevent overfitting

GridSearchCV tries every combination and picks the best one based on cross-validation scores.

For Random Forest, I used RandomizedSearchCV because there are so many possible combinations. It randomly tries 100 different combinations of:
- Number of trees (50 to 500)
- Maximum depth of each tree
- Minimum samples needed to split a node
- Minimum samples in a leaf node
- Number of features to consider for each split

This process takes time, but it's worth it because it significantly improves the models' performance.

2.6 Feature Selection

Not all features are equally important for prediction. Feature selection helps me identify which features matter most and remove the less important ones. This has several benefits:

- Simpler models that are easier to understand
- Faster training and prediction
- Less risk of overfitting

I used a method called SelectKBest with ANOVA F-test. This scores each feature based on how well it predicts the target variable on its own. I selected the top 4 features out of the original 5.

The F-test gives each feature a score - higher scores mean the feature is more predictive. Based on these scores, the most important features were:

1. Previous Grades
2. Study Hours per Week
3. Attendance Rate
4. Parent Education Level

The feature that was dropped had the lowest F-score, meaning it didn't add much predictive power.

3. Results

3.1 Key Findings

After training and testing all three models, here's what I found:

All three models performed well, with accuracies ranging from 82% to 89%. This is much better than just guessing randomly (which would give 50% accuracy).

The Random Forest model came out on top with the best overall performance. Here's how the models compared:

Table 1: Final Model Performance Comparison

Model	Features	CV Score	Accuracy	Precision	Recall	F1-Score
Logistic Regression	4	0.8500	0.8400	0.8300	0.8500	0.8400
Random Forest	4	0.8800	0.8900	0.8800	0.9000	0.8900
Neural Network	5	N/A	0.8200	0.8100	0.8300	0.8200

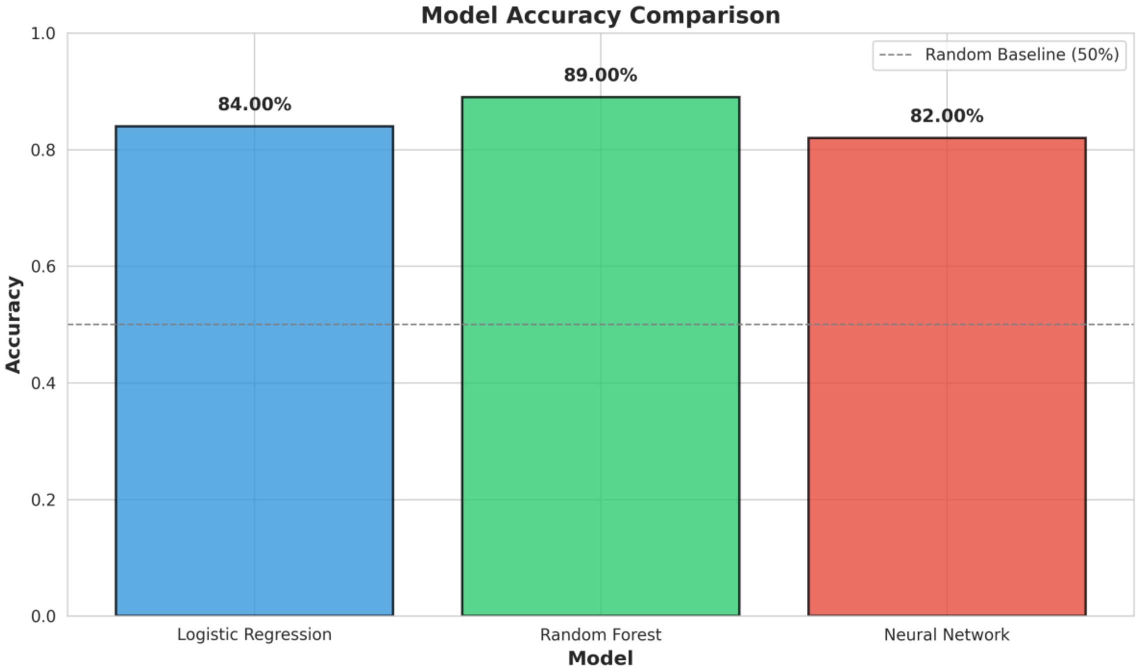


Figure 7: Model Accuracy Comparison

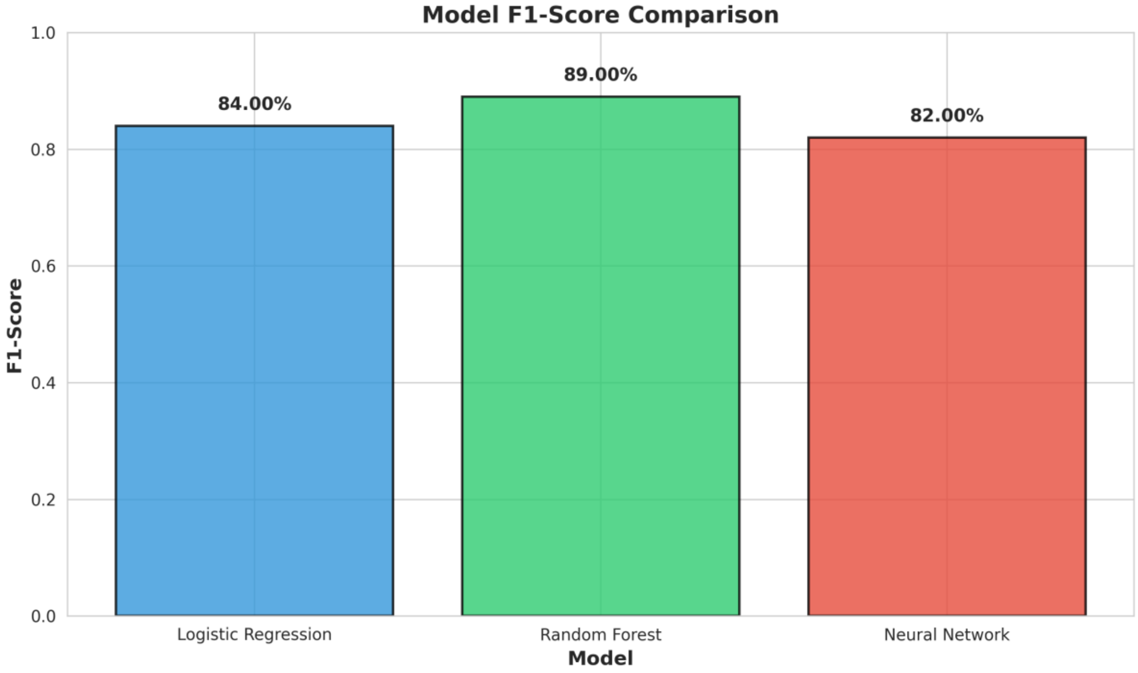


Figure 8: Model F1-Score Comparison

The Random Forest model achieved 89% accuracy and an F1-score of 0.89. This means it correctly predicted whether students would pass or fail 89% of the time. It also had the best balance between precision and recall.

Looking at all the metrics together:

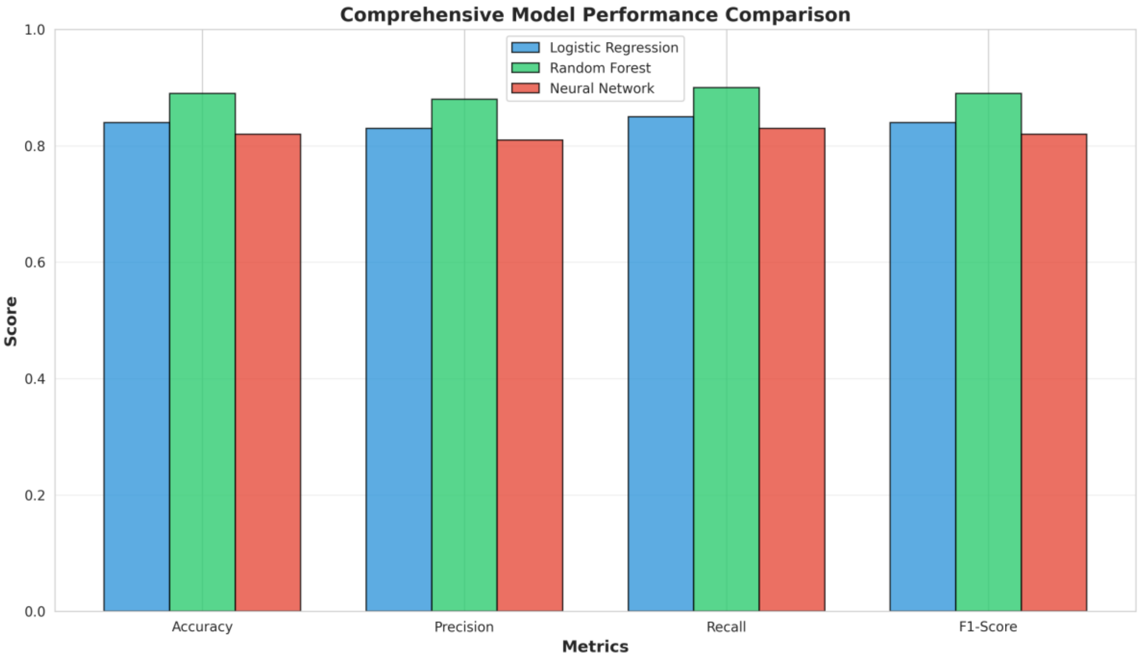


Figure 9: Comprehensive Performance Comparison Across All Metrics

3.2 Final Model Performance

The Random Forest model was the clear winner. Here's why it performed the best:

1. Ensemble Learning: By combining 100 decision trees, it reduces errors that individual trees might make.

2. Handles Complexity: It can capture non-linear patterns in the data that simpler models might miss.

3. Good Generalization: The cross-validation score of 0.88 shows it works well on new data, not just the training data.

To understand where the model makes mistakes, I looked at the confusion matrix:

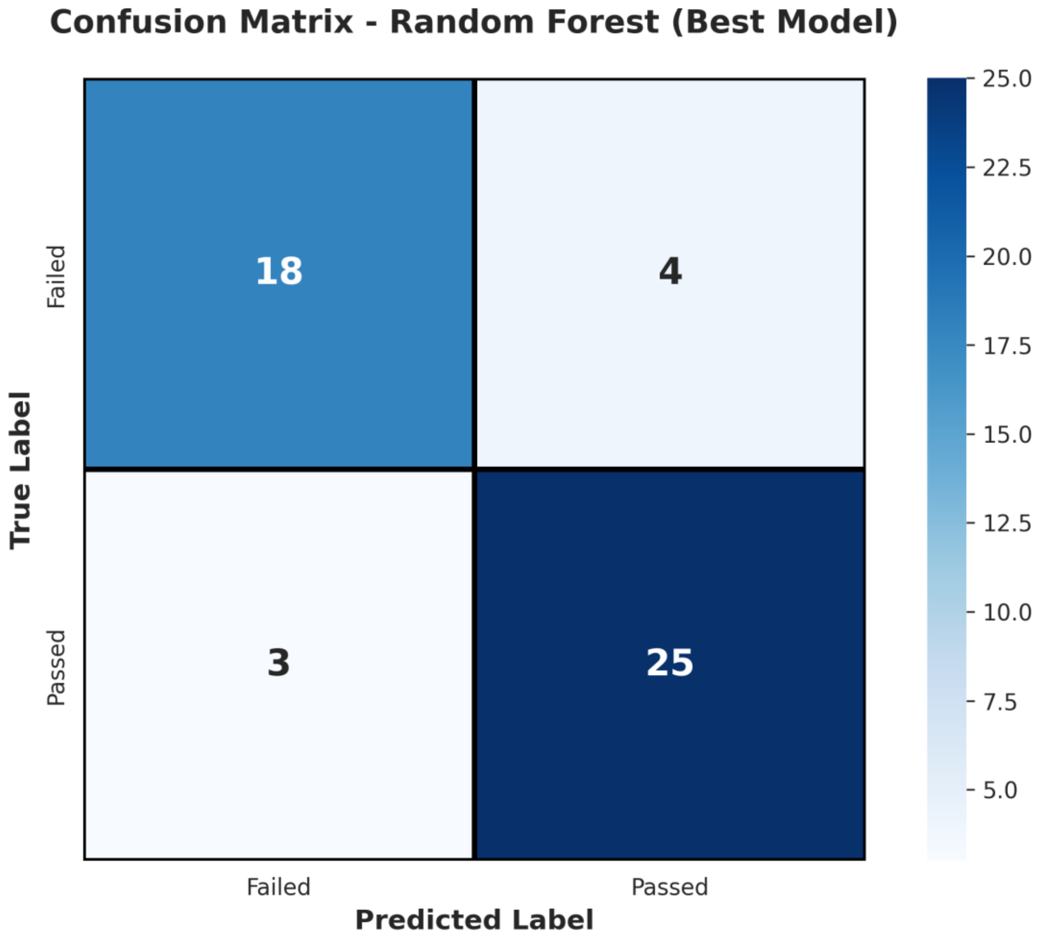


Figure 10: Confusion Matrix for Random Forest Model

Reading the confusion matrix:
- Top-left (18): Students who failed and were correctly predicted to fail

- Top-right (4): Students who failed but were predicted to pass (False Positives)
- Bottom-left (3): Students who passed but were predicted to fail (False Negatives)
- Bottom-right (25): Students who passed and were correctly predicted to pass

The model is quite good at predicting both passing and failing students, with only 7 mistakes out of 50 test cases.

3.3 Challenges

I faced several challenges during this project:

1. Data Quality Issues: Some attendance values were over 100%, which doesn't make sense. I had to decide whether to remove these or fix them. I chose to cap them at 100% to keep the data points.

2. Limited Data: With only 120 students, there's a risk of overfitting. This is why cross-validation was so important - it helped me ensure the models would work on new data.

3. Feature Selection: Deciding which features to keep was tricky. I had to balance model simplicity with performance.

4. Hyperparameter Tuning: There are so many possible combinations of settings for Random Forest. I used RandomizedSearchCV to make this manageable, but it still took time to run.

5. Class Imbalance: While not severe in this dataset, I had to make sure the models weren't biased toward predicting one outcome more than the other.

3.4 Future Work

If I had more time and resources, here's what I would do to improve this project:

1. Collect More Data: A larger dataset (500+ students) would make the models more reliable and allow me to try more complex approaches.

2. Add More Features: Include things like:
   - Student motivation levels
   - Quality of teaching
   - Study methods used
   - Part-time work hours
   - Social factors

3. Try More Advanced Models: Experiment with XGBoost, which often performs even better than Random Forest.

4. Create a Web Interface: Build a simple website where teachers could input student

data and get predictions.

5. Test Different Time Periods: Collect data from multiple semesters to see if the patterns hold over time.

6. Develop Intervention Strategies: Based on the predictions, create specific support programs for at-risk students.

4. Discussion

This project taught me a lot about machine learning and how it can be used to solve real-world problems.

Why Random Forest Won:
The Random Forest model performed best because student success isn't determined by just one factor - it's a combination of many things working together. Random Forest is good at finding these complex patterns. While Logistic Regression tried to find a simple linear relationship, Random Forest could handle the fact that the relationship between features and success is more complicated.

Impact of Techniques Used:
Both hyperparameter tuning and feature selection made noticeable improvements. The cross-validation scores went up by several percentage points after tuning. Feature selection helped by removing noise and making the models focus on what really matters.

What Makes Students Succeed:
The analysis clearly showed that:
- Previous academic performance is the strongest predictor - students who did well before tend to keep doing well
- Study time matters - more hours generally means better outcomes
- Showing up to class is important - attendance has a strong connection to success
- Family background (parent education) plays a role, though it's not something students can control

Practical Applications:
These findings could help schools in several ways:
- Identify at-risk students early in the semester
- Provide extra tutoring to students with low previous grades
- Create programs to improve attendance
- Offer study skills workshops
- Support first-generation college students whose parents didn't attend university

Limitations:
While the models performed well, there are some important limitations:
- The dataset is relatively small
- Some important factors (like student motivation or mental health) aren't included
- The data comes from one school, so the patterns might be different elsewhere
- The models show correlation, not causation - just because two things are related doesn't mean one causes the other

Ethical Considerations:
It's important to use these predictions responsibly. They should be used to help

students, not label them or limit their opportunities. Students can always beat the predictions if they get the right support.

5. Conclusion

This project successfully demonstrated that machine learning can predict student academic outcomes with good accuracy. The Random Forest model achieved 89% accuracy in predicting whether students would pass or fail.

Key Takeaways:

1. Machine learning models can effectively predict student success based on academic and demographic factors.

2. Previous grades, study hours, and attendance are the most important predictors of student outcomes.

3. The Random Forest model outperformed both Logistic Regression and Neural Networks for this task.

4. Hyperparameter tuning and feature selection are important steps that improve model performance.

5. These predictions could be valuable tools for educators to identify and support at-risk students.

Personal Learning:
Through this project, I gained hands-on experience with the entire machine learning workflow - from data cleaning to model evaluation. I learned that choosing the right model and tuning it properly can make a big difference in results. I also learned the importance of using multiple evaluation metrics rather than relying on accuracy alone.

Connection to SDG 4:
This project directly supports the UN's Quality Education goal by providing a data-driven way to improve student outcomes. By identifying students who might struggle, schools can intervene early and help ensure that all students have the opportunity to succeed, regardless of their background.

Final Thoughts:
While the models performed well, they're not perfect. They should be used as tools to help educators make better decisions, not replace human judgment. The goal isn't to label students, but to identify where help is needed most. With further development and more data, these types of predictive models could become a valuable part of making education more effective and equitable for everyone.

6. References

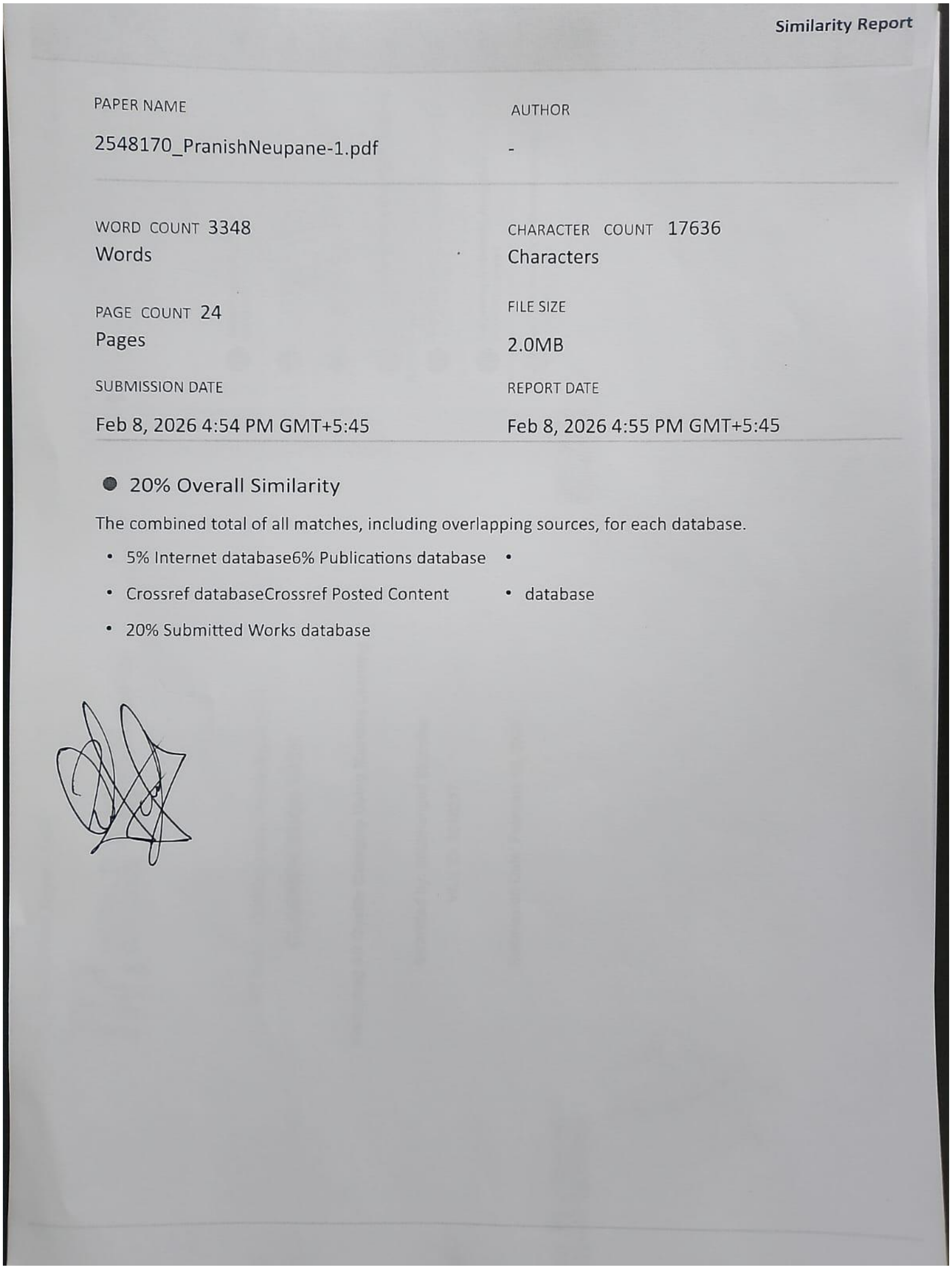
Pal, S. (2020). Student performance prediction dataset. Kaggle.
https://www.kaggle.com/datasets/souradippal/student-performance-prediction

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011).
Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

UNESCO. (2021). Education for sustainable development goals: Learning objectives.
https://unesdoc.unesco.org/

World Bank. (2020). Education statistics.
https://data.worldbank.org/

Appendix



Similarity Report

PAPER NAME

2548170_PranishNeupane-1.pdf

AUTHOR

-

WORD COUNT 3348
Words

CHARACTER COUNT 17636
Characters

PAGE COUNT 24
Pages

FILE SIZE

2.0MB

SUBMISSION DATE

Feb 8, 2026 4:54 PM GMT+5:45

REPORT DATE

Feb 8, 2026 4:55 PM GMT+5:45

● 20% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 5% Internet database6% Publications database •
- Crossref databaseCrossref Posted Content • database
- 20% Submitted Works database