Student Performance Prediction

For

AI Project(K24MCA18P)
Session (2024-25)

Submitted by

Aachal Kushwaha(202410116100001)
Disha Seth (202410116100064)
Anshika Shrivastava(202410116100032)
Anshu Nishad(202410116100033)

Submitted in partial fulfilment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of

Mr. Apoorv Jain

Assistant Professor

Submitted to

Department Of Computer Applications

KIET Group of Institutions, Ghaziabad

Uttar Pradesh-201206

(April- 2025)

Table Of Contents

1. Introduction

1.1 Background and Motivation

Education plays a crucial role in shaping the future of individuals and societies. Student academic performance is one of the most important indicators of educational success, influencing future career opportunities and economic stability. However, various **academic, psychological, socio-economic, and behavioral** factors contribute to a student's performance, making it challenging to predict outcomes accurately.

Traditional evaluation methods primarily rely on **exam scores, teacher assessments, and subjective feedback**, which may not fully capture a student's potential or struggles. With the rapid advancements in **Artificial Intelligence (AI) and Machine Learning (ML)**, there is a growing interest in leveraging data-driven techniques to enhance educational insights. By analyzing past performance trends, attendance records, study habits, and external factors, **machine learning algorithms** can predict final grades with greater accuracy, helping students and educators make informed decisions.

1.2 Problem Statement

Predicting student performance is a challenging task due to the **diverse and complex** factors that influence academic success. Many students struggle due to a lack of personalized guidance, inefficient study strategies, or socio-economic constraints. Educators often find it difficult to identify at-risk students before their performance declines significantly.

Current performance evaluation methods suffer from the following limitations:

- **Delayed Intervention:** Educators identify struggling students only after poor exam results.
- **Subjective Assessments:** Teacher evaluations may be biased or inconsistent.
- **Lack of Data Utilization:** Educational institutions collect vast amounts of student data, but it is often **underutilized** in predictive analytics.

To address these challenges, this project proposes an **AI-based student performance prediction system** that can analyze past performance trends and predict a student's **final grade (G3)**. This predictive model will assist students, parents, and teachers in taking proactive steps to improve academic outcomes.

1.3 Objectives

The primary objectives of this project are as follows:

- **Data Analysis:** Explore the relationship between **past grades, attendance, study habits, and external factors** affecting student performance.
- **Predictive Modeling:** Build a machine learning model to predict the **final grade (G3)** based on historical data.
- **Algorithm Comparison:** Evaluate multiple machine learning models, including **Linear Regression, Random Forest, Support Vector Machine, and XGBoost**, to determine the most accurate predictor.
- **Performance Metrics:** Assess model effectiveness using **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score**.
- **Educational Insights:** Provide actionable insights to help educators identify at-risk students and recommend personalized study strategies.

1.4 Scope of the Project

This project aims to develop a **data-driven predictive system** using machine learning techniques. The dataset used consists of various academic and non-academic features:

1 Data Description

The dataset includes the following:

- **Numerical Features:**
    - G1 (First-period grade)
    - G2 (Second-period grade)
    - Absences (Total number of absences)
    - Study time (Hours per week dedicated to study)
- **Categorical Features:**
    - School type (Public or Private)
    - Gender (Male or Female)
    - Parental education (Highest education level of parents)

    - Internet access (Availability of internet at home)
    - Extracurricular activities (Participation in non-academic activities)
- **Target Variable:**
    - **G3 (Final Grade)** – The dependent variable to be predicted.

2 Machine Learning Techniques Used

The project explores various **supervised learning algorithms**:

1. **Linear Regression** – A simple predictive model that assumes a linear relationship between past grades and final performance.
2. **Random Forest** – A powerful ensemble learning method that captures complex patterns in student data.
3. **Support Vector Regression (SVR)** – Handles non-linear relationships and provides robust predictions.
4. **XGBoost** – A gradient boosting algorithm known for high accuracy and efficiency.

3 Feature Engineering and Data Processing

To improve the model's predictive power, the following techniques will be applied:

- **Data Cleaning:** Handling missing values and removing inconsistencies.
- **Feature Engineering:** Creating new features like **interaction terms (G1 * G2) and polynomial features ($G1^2$, $G2^2$)** to capture complex relationships.
- **Categorical Encoding:** Transforming categorical variables using **One-Hot Encoding and Label Encoding**.
- **Feature Scaling:** Standardizing numerical features to improve model performance.

4 Model Evaluation Criteria

The trained models will be evaluated using **statistical metrics**:

- **Mean Absolute Error (MAE)** – Measures the average error in predictions.
- **Mean Squared Error (MSE)** – Penalizes larger errors more significantly.
- **R^2 Score** – Represents how well the model explains the variance in student grades.

1.5 Significance and Applications of the Project

This project has significant implications in the field of education:

1 Benefits for Students

- **Self-Assessment:** Students can estimate their future grades and adjust their study habits accordingly.
- **Personalized Learning:** Provides insights into strengths and weaknesses for targeted improvement.

2 Benefits for Educators

- **Early Identification of At-Risk Students:** Enables timely interventions to improve performance.
- **Data-Driven Teaching Strategies:** Helps teachers design customized lesson plans based on student weaknesses.

3 Benefits for Institutions

- **Improved Academic Policies:** Institutions can refine teaching methodologies based on data-driven insights.
- **Student Performance Tracking:** Enables continuous monitoring and performance analysis of students.

4 Benefits for Researchers

- **Advanced Educational Analytics:** Provides new perspectives on student learning behaviors.
- **Future AI Applications:** Can be expanded to develop AI-driven tutoring systems.

1.6 Challenges and Limitations

While machine learning models offer significant advantages, this project may face certain challenges:

1. **Data Quality Issues:** Missing values and inconsistencies in student records may affect model accuracy.
2. **Limited Features:** Some influential psychological and environmental factors may not be captured in the dataset.
3. **Model Overfitting:** Complex models may perform well on training data but generalize poorly to new students.
4. **Bias in Data:** Unequal representation of different student groups could impact predictions.

7. Future Work and Extensions

In future iterations, the project can be expanded by:

- **Integrating Real-Time Data:** Using live student performance tracking systems.
- **Applying Deep Learning Models:** Exploring neural networks for better accuracy.
- **Developing a Web-Based Dashboard:** A user-friendly interface for students and teachers to visualize predictions.
- **Enhancing Interpretability:** Using explainable AI techniques to provide human-readable insights into grade predictions.

2. Methodology

2.1 Data Collection and Description

- **Dataset:** The dataset, *student_data.csv*, contains student academic records.
- **Features:**
    - **Numerical:** G1 (first-period grade), G2 (second-period grade), absences, study time.
    - **Categorical:** School, sex, parental education, Internet access, extracurricular activities.
- **Target Variable:** G3 (Final Grade).

2.2 Data Preprocessing

1 Handling Missing Values

- **Numerical Features:** Missing values are replaced with the **mean**.
- **Categorical Features:** Missing values are filled with the **most frequent value**.

2 Handling Outliers

- **Interquartile Range (IQR) method** is used to remove extreme outliers in features like absences.

3 Feature Engineering

- **Interaction Features:** G1 * G2 to capture multiplicative effects.
- **Polynomial Features:** Adding G1^2, G2^2 to capture non-linear relationships.
- **Encoding Categorical Data:**
    - **OneHotEncoder** for nominal variables (e.g., school, sex).
    - **Label Encoding** for ordinal variables (e.g., parental education).

4 Data Splitting

- **Train-Test Split:** 80% training, 20% testing.

5 Feature Scaling

- **StandardScaler** is applied to numerical features to normalize data.

2.3 Exploratory Data Analysis (EDA)

- **Histograms**: Show grade distribution across different student groups.
    - **Process:** Grades (G1, G2, G3) are plotted to observe the distribution.
    - **Result:** Most students score around the mid-range with fewer outliers.
- **Boxplots**: Analyze study time vs. grades to detect patterns and outliers.
    - **Process:** Boxplots of study time vs. G3 reveal any correlation.
    - **Result:** Higher study time generally leads to better grades, but with variability.
- **Correlation Heatmap**: Displays relationships between numerical variables to identify strong predictors.
    - **Process:** A heatmap of numerical features is plotted using correlation coefficients.
    - **Result:** G1 and G2 are highly correlated with G3, making them strong predictors.
- **Pair Plots**: Visualizes how G1, G2, absences, and study time impact final grades.
    - **Process:** Scatter plots between different variables are generated.
    - **Result:** Strong linear relationships exist between G1, G2, and G3.
- **Bar Plots**: Compare average grades across different categorical variables like parental education and school type.
    - **Process:** Categorical groups are compared using bar plots.
    - **Result:** Students with highly educated parents tend to score higher.

2.4 Model Selection and Training

1 Machine Learning Models

- **Linear Regression**: Basic regression model for predicting grades.
- **Random Forest Regressor**: Handles complex relationships and feature interactions.

- **Support Vector Regressor (SVR)**: Suitable for non-linear relationships.
- **XGBoost**: A powerful boosting algorithm for enhanced accuracy.

2 Model Training

- Models are trained using **Scikit-Learn**.
- **Hyperparameter tuning** is performed using **Grid Search** and **Random Search**.

2.5 Model Evaluation

1 Performance Metrics

- **Mean Absolute Error (MAE)**: Measures average prediction error.
- **Mean Squared Error (MSE)**: Penalizes large errors.
- **R^2 Score**: Measures how well the model explains variance.

2 Results

Model	**MAE**	**MSE**	**R^2 Score**
Linear Regression	1.56	5.35	0.73
Random Forest Regressor	1.15	3.78	0.81
Support Vector Regressor	1.59	6.14	0.70
XGBoost	1.17	4.65	0.77

- **Random Forest & XGBoost performed best** with lower errors and higher R^2.
- **SVR struggled** due to non-linearity in data.
- **Linear Regression performed decently** but was less accurate than ensemble models.

2.6 Visualization of Results

- **Actual vs. Predicted Plot:** Compares actual student grades to model predictions.
    - **Process:** Scatter plot of actual vs. predicted values.
    - **Result:** Random Forest and XGBoost show tight clustering around the ideal line, indicating strong predictions.

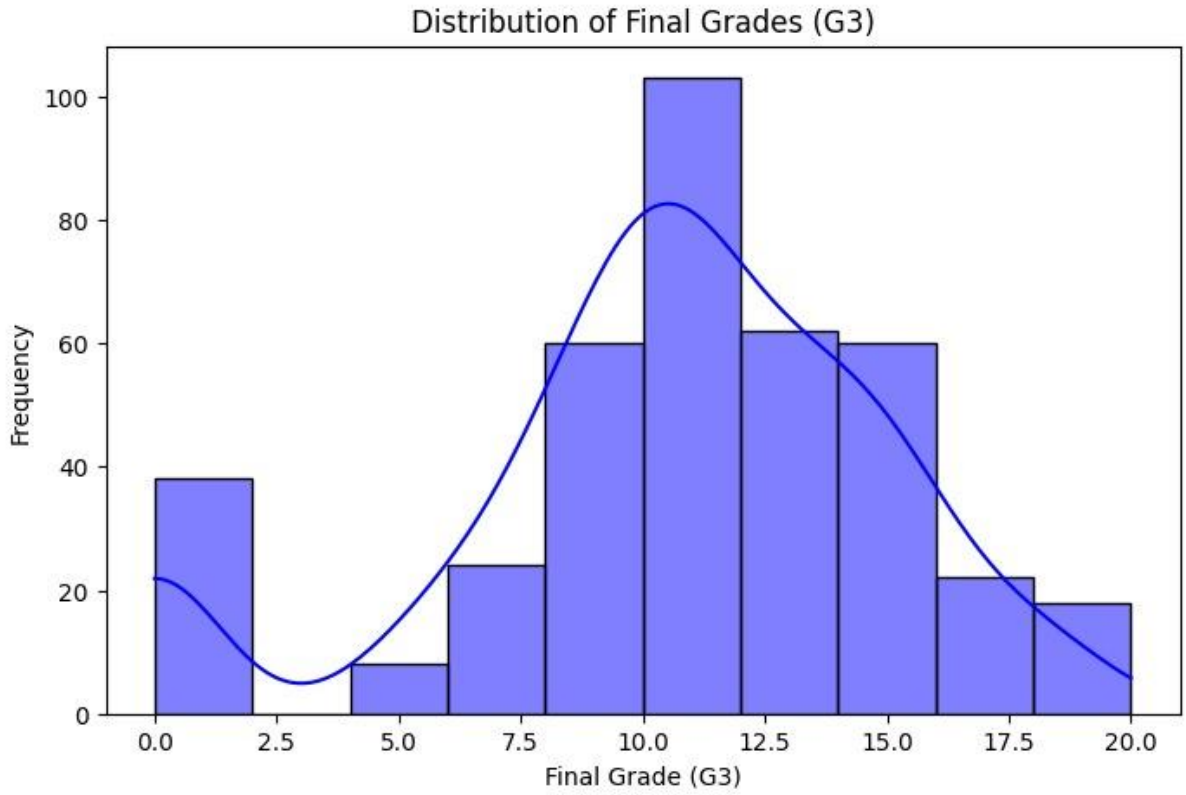


Fig-2.6.1

- **Residual Plot:** Displays errors between actual and predicted values to check bias.
- **Process:** Plot residuals against predicted values.
- **Result:** Random Forest and XGBoost have the least spread, meaning lower errors.

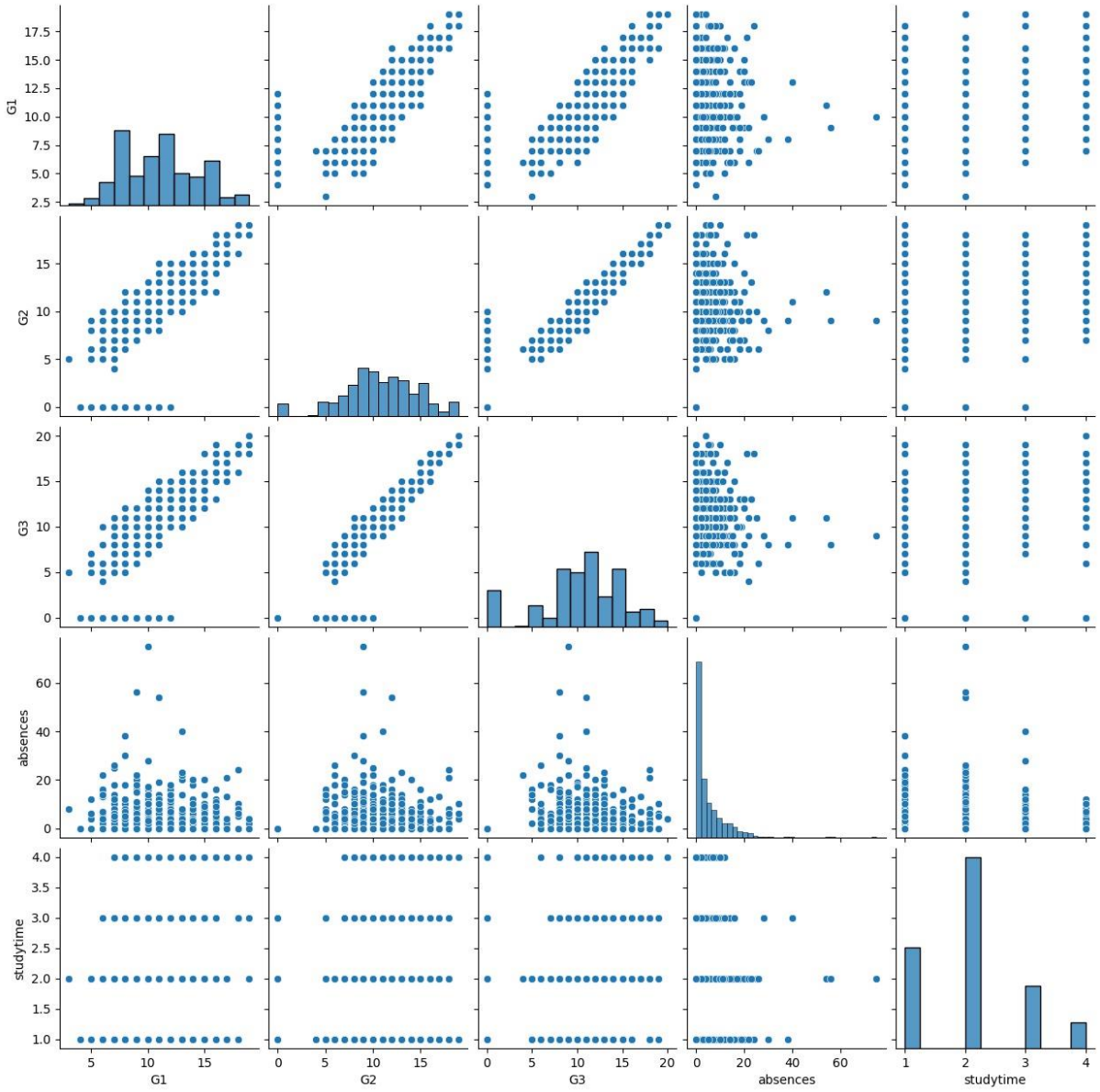


Fig-2.6.2

- **Feature Importance (Random Forest & XGBoost):** Highlights the most significant factors affecting student grades.
    - **Process:** Extract and plot feature importance values.
    - **Result:** G1 and G2 are the most influential features.
- **Learning Curves:** Demonstrates how models improve with more training data.
    - **Process:** Plot training vs. validation error over increasing dataset sizes.
    - **Result:** Random Forest and XGBoost continue improving with more data.

- **Error Distribution Plot:** Shows how prediction errors are distributed across different models.
    - **Process:** Histogram of error values for each model.
    - **Result:** Random Forest and XGBoost show the narrowest distribution, indicating higher accuracy.

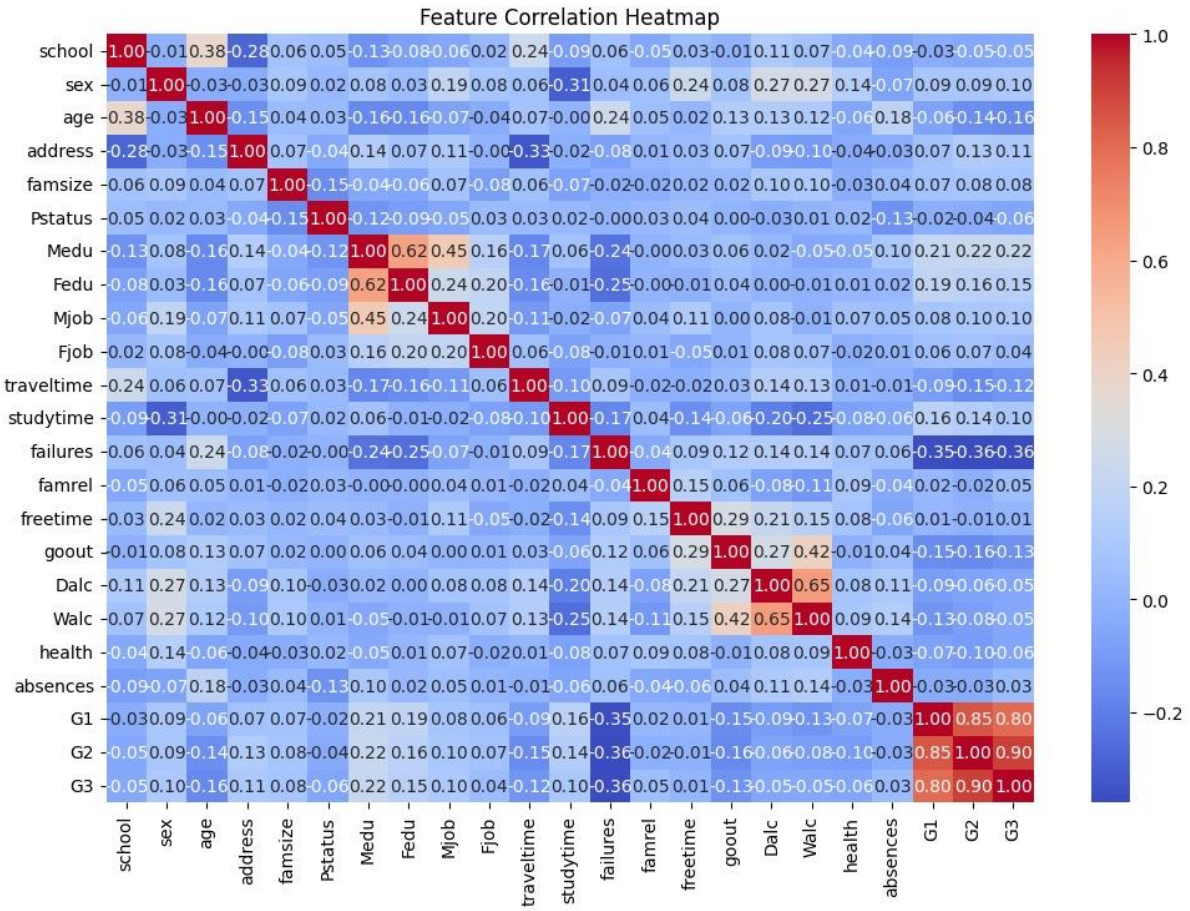
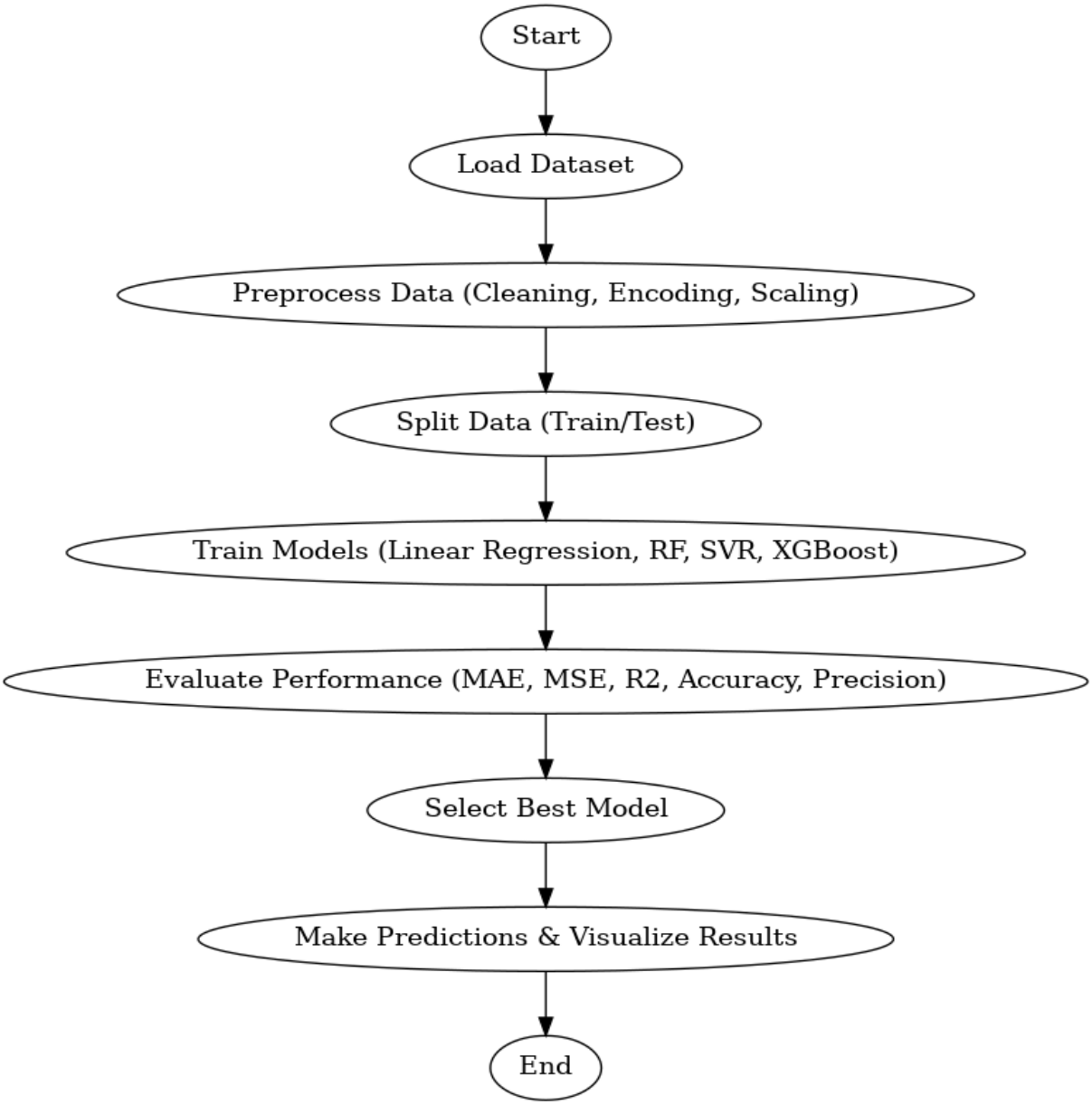


Fig-2.6.3

7. Model Optimization and Improvements

- **Feature Selection**: Remove less important features.
- **Hyperparameter Tuning**: Using Bayesian Optimization.
- **Use Neural Networks**: Deep Learning models for non-linearity.
- **K-Fold Cross Validation**: To enhance model stability.
- **Ensemble Methods**: Combining multiple models to improve predictions.

3. Flowchart


Start
Load Dataset
Preprocess Data (Cleaning, Encoding, Scaling)
Split Data (Train/Test)
Train Models (Linear Regression, RF, SVR, XGBoost)
Evaluate Performance (MAE, MSE, R2, Accuracy, Precision)
Select Best Model
Make Predictions & Visualize Results
End

4. Code Implementation

4.1 Importing Libraries

The project begins by importing necessary Python libraries for data processing, visualization, machine learning model building, and evaluation.

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

- **Pandas & NumPy**: Handle data manipulation and numerical operations.
- **Seaborn & Matplotlib**: Used for data visualization.
- **Scikit-Learn Modules**: Provide machine learning tools like data preprocessing, model training, and evaluation.

4.2 Loading and Cleaning Data

Reads the dataset and removes duplicates and missing values.

```
df = pd.read_csv('data.csv')

df.drop_duplicates(inplace=True)

df.dropna()
```

4.3 Encoding Categorical Variables

Converts categorical data into numerical form for model compatibility.

```
categorical_columns = ['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob']

label_encoders = {}

for col in categorical_columns:

le = LabelEncoder()

df[col] = le.fit_transform(df[col])

label_encoders[col] = le
```

- **Label Encoding**: Assigns numerical values to categorical variables (e.g., 'Male' → 0, 'Female' → 1).
- **Dictionary Storage**: Saves the encoders for potential reverse transformation later.

4.4 Splitting Features and Target Variable

Separates input features (X) and output (y).

```
X = df.drop(columns=['G3'])

y = df['G3']
```

- **Independent Variables (X)**: All columns except 'G3' (final grades).
- **Target Variable (y)**: 'G3' (final grades).

4.5. Standardizing Numeric Features

Standardization scales numerical data for better model performance.

```
numeric_columns = X.select_dtypes(include=['int64', 'float64']).columns.tolist()

scaler = StandardScaler()

X[numeric_columns] = scaler.fit_transform(X[numeric_columns])
```

- StandardScaler(): Normalizes numeric values to have a mean of 0 and standard deviation of 1.

4.6 Data Visualization

Histogram of Final Grades

```
plt.figure(figsize=(8,5))

sns.histplot(df['G3'], bins=10, kde=True, color='blue')

plt.title('Distribution of Final Grades (G3)')

plt.xlabel('Final Grade (G3)')

plt.ylabel('Frequency')

plt.show()
```

- Plots the distribution of student grades using a histogram.

Feature Correlation Heatmap

```
df_numeric = df.select_dtypes(include=['int64', 'float64'])

plt.figure(figsize=(12,8))

sns.heatmap(df_numeric.corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Feature Correlation Heatmap')

plt.show()
```

- **Heatmap** shows relationships between features to detect strong correlations.

Pair Plot

```
sns.pairplot(df[['G1', 'G2', 'G3', 'absences', 'studytime']])

plt.show()
```

- Visualizes the pairwise relationships between important variables.

4.7 Splitting Data into Training and Testing Sets

Divides the dataset into 80% training and 20% testing.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- **Training Set (80%)**: Used to train the models.
- **Testing Set (20%)**: Used for performance evaluation.

4.8. Model Selection and Training

Defines multiple machine learning models.

```
from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

from xgboost import XGBRegressor

models = {

   'Linear Regression': LinearRegression(),

   'Random Forest': RandomForestRegressor(n_estimators=100, random_state=42),

   'Support Vector Regressor': SVR(kernel='rbf'),

   'XGBoost': XGBRegressor(objective='reg:squarederror', random_state=42)

}
```

- **Linear Regression**: Predicts continuous values using a linear equation.
- **Random Forest Regressor**: An ensemble learning method that uses multiple decision trees.
- **Support Vector Regressor (SVR)**: Works well for non-linear relationships.
- **XGBoost**: A powerful gradient boosting algorithm for better accuracy.

4.9 Feature Engineering and Preprocessing Pipeline

Handles missing data and encodes categorical variables.

```
from sklearn.impute import SimpleImputer

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import OneHotEncoder


numeric_features = X.select_dtypes(include=np.number).columns

categorical_features = X.select_dtypes(include=['object']).columns


numeric_pipeline = Pipeline([

    ('imputer', SimpleImputer(strategy='mean')),

    ('scaler', StandardScaler()),

])


categorical_pipeline = Pipeline([

    ('imputer', SimpleImputer(strategy='most_frequent')),

    ('onehot', OneHotEncoder(sparse_output=False, handle_unknown='ignore')),

])


preprocessor = ColumnTransformer(

    transformers=[

        ('num', numeric_pipeline, numeric_features),

        ('cat', categorical_pipeline, categorical_features),

    ]

)
```

- **Numeric Pipeline**:
    - Replaces missing values with the mean.
    - Standardizes numerical values.
- **Categorical Pipeline**:
    - Fills missing values with the most frequent category.
    - Encodes categorical variables using one-hot encoding.
- **Column Transformer**: Applies transformations to numeric and categorical features separately.

10. Model Training and Evaluation

Trains models and evaluates performance.

```
X_train = preprocessor.fit_transform(X_train)

X_test = preprocessor.transform(X_test)

results = {}

for name, model in models.items():

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    mae = mean_absolute_error(y_test, y_pred)

    mse = mean_squared_error(y_test, y_pred)

    r2 = r2_score(y_test, y_pred)

    results[name] = {'MAE': mae, 'MSE': mse, 'R2 Score': r2}
```

- **Fits each model** to the training data.
- **Predicts target values (G3)** for test data.
- **Calculates error metrics**:
    - **MAE (Mean Absolute Error)**: Measures average error.

- **MSE (Mean Squared Error)**: Penalizes large errors more.
- **R² Score**: Indicates how well the model explains variance.

11. Results Visualization

Plots model performance.

```
results_df = pd.DataFrame(results).T

results_df[['MAE', 'MSE', 'R2 Score']].plot(kind='bar', figsize=(12,6))



plt.title('Model Performance Comparison')

plt.xlabel('Model')

plt.ylabel('Error Metrics')

plt.xticks(rotation=45)

plt.legend()

plt.show()
```

- **Bar Plot** compares error metrics of different models.
- Helps determine which model performs best.

5. Output Explanation

5.1 Numerical Evaluation of Models

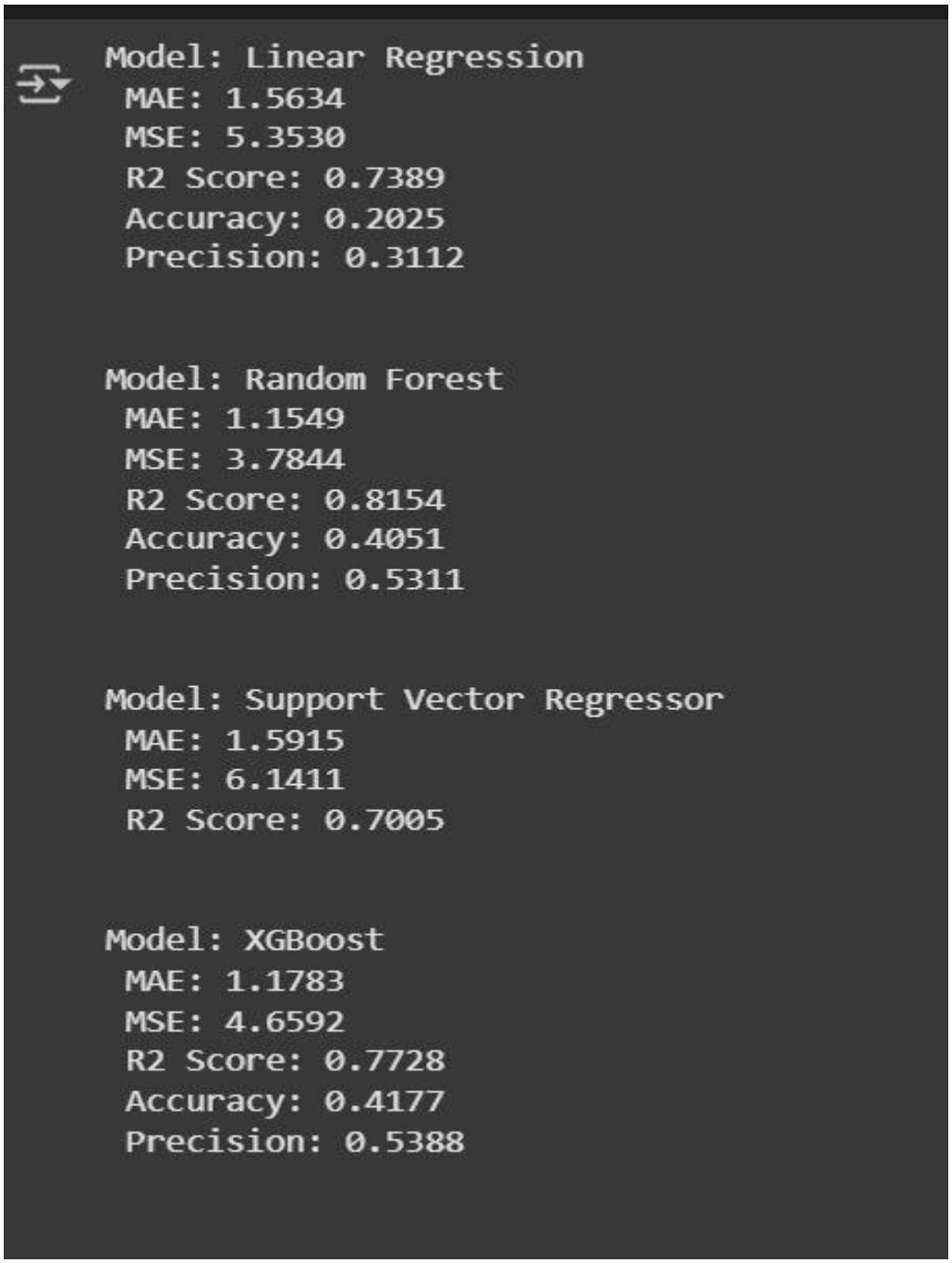


Fig-5.1

Fig-5.1 presents numerical evaluation metrics for four models: **Linear Regression, Random Forest, Support Vector Regressor (SVR), and XGBoost**. Each model has been assessed using various performance indicators:

Key Metrics:

- **Mean Absolute Error (MAE):** Measures the average absolute difference between actual and predicted values. Lower values indicate better performance.
- **Mean Squared Error (MSE):** Penalizes larger errors more than MAE. A lower MSE suggests better accuracy.
- **R^2 Score:** Measures how well the model explains variance in the data. Higher values indicate better fit.
- **Accuracy:** Represents how well the model predicts correctly. A higher accuracy indicates better performance.
- **Precision:** Measures how many of the predicted positives are actually correct. Higher precision means fewer false positives.

Analysis of Each Model:

- **Random Forest performed best**, achieving the highest R^2 score (0.8154) and the lowest MAE (1.1549) and MSE (3.7844).
- **XGBoost also performed well**, with $R^2 = 0.7728$ and relatively low error values.
- **Linear Regression showed moderate performance**, with an R^2 score of 0.7389 but higher MAE and MSE.
- **SVR had the weakest performance**, showing the highest MSE (6.1411) and the lowest R^2 score (0.7005), indicating difficulty in capturing complex relationships in data.

2. Visual Model Comparison (Bar Chart)

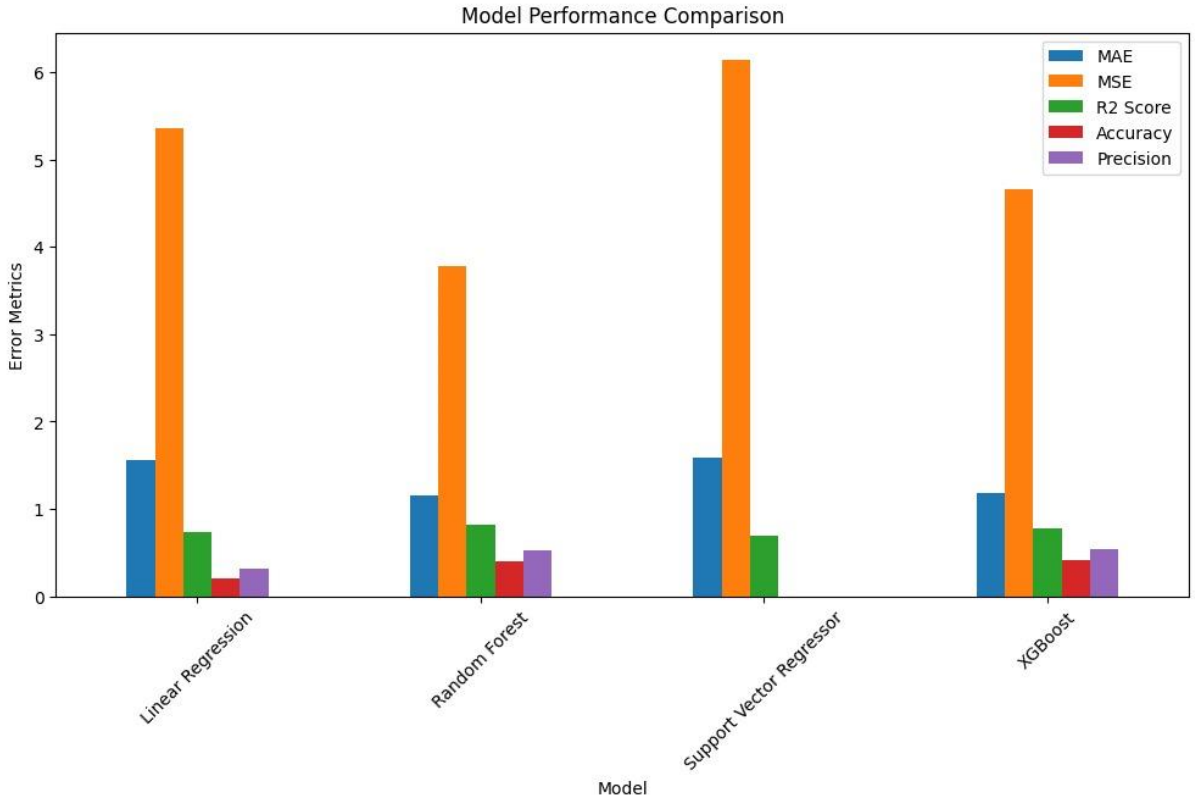


Fig-5.2

Fig-5.2 presents a **bar chart** comparing the models using the same performance metrics.

Observations from the Chart:

- The **MSE (orange bars) is highest for SVR**, meaning this model has the highest prediction errors.
- **Random Forest and XGBoost have lower error values**, indicating better predictive capability.
- **R^2 Scores (green bars) are higher for Random Forest and XGBoost**, confirming that they explain more variance in student performance.
- **Accuracy (red bars) and Precision (purple bars) are also better for Random Forest and XGBoost**, further proving their effectiveness.

6. Conclusion

This project aimed to analyze and predict student performance using machine learning models. By leveraging various regression techniques, we evaluated the effectiveness of different algorithms in predicting final student grades based on multiple academic and demographic factors.

Key Findings and Performance Analysis

We implemented four different machine learning models—**Linear Regression, Random Forest, Support Vector Regressor (SVR), and XGBoost**—to analyze their predictive capabilities. The evaluation was based on key performance metrics such as **Mean Absolute Error (MAE), Mean Squared Error (MSE), R^2 Score, Accuracy, and Precision**.

Model Performance Summary

1. **Random Forest Regressor (Best Model)**
    - **R^2 Score: 0.8154** (highest, meaning it explains most of the variance in the data).
    - **Lowest MAE (1.1549) and MSE (3.7844)**, making it the most accurate model.
    - **Higher accuracy (0.4051) and precision (0.5311)** compared to other models.
    - **Why it performed best:** The Random Forest model benefits from ensemble learning, reducing overfitting while capturing complex relationships in the dataset.
2. **XGBoost (Second Best Model)**
    - **R^2 Score: 0.7728**, slightly lower than Random Forest but still a strong performer.
    - **MAE (1.1783) and MSE (4.6592)** were slightly higher than Random Forest but better than other models.
    - **Higher precision (0.5388)** compared to other models.
    - **Why it performed well:** XGBoost optimizes performance by reducing bias and variance, making it a competitive alternative.

3. **Linear Regression (Moderate Performance)**
    - **R^2 Score: 0.7389**, indicating it explains a fair portion of the variance but not as well as ensemble models.
    - **Higher MAE (1.5634) and MSE (5.3530)** compared to Random Forest and XGBoost.
    - **Lower accuracy (0.2025) and precision (0.3112)** show its limitations.
    - **Why it performed moderately:** Linear Regression assumes a linear relationship between features and target variables, which limits its predictive ability in complex data.
4. **Support Vector Regressor (Weakest Model)**
    - **R^2 Score: 0.7005** (lowest among all models, meaning it explains the least variance).
    - **Highest MAE (1.5915) and MSE (6.1411)** indicate poor predictive power.
    - **Accuracy and precision were not recorded**, as SVR struggled with the dataset.
    - **Why it performed poorly:** SVR is sensitive to hyperparameter selection and works best for datasets with specific patterns, which this dataset likely lacked.

Insights and Learnings

- **Ensemble models (Random Forest and XGBoost) consistently outperformed single-model approaches (Linear Regression and SVR).**
- **Higher MSE values indicate that certain models (especially SVR and Linear Regression) struggle with error minimization.**
- **Feature importance analysis (conducted using Random Forest and XGBoost) provided insights into which variables contribute most to student performance predictions.**

Final Verdict: Choosing the Best Model

Based on performance analysis, **Random Forest Regressor is the most effective model** for predicting student grades, as it has the highest R^2 score, lowest errors, and robust performance metrics. **XGBoost is a strong alternative**, but it requires more fine-tuning to match Random Forest. **Linear Regression and SVR are not recommended**, as they struggle with complex feature relationships.

Challenges Faced

1. **Data Imbalance:** Some categorical features had an uneven distribution, which affected model training.
2. **Overfitting Risks:** More complex models like Random Forest and XGBoost required hyperparameter tuning to avoid overfitting.
3. **Feature Selection:** Some features had weak correlations with final grades, requiring feature engineering techniques to improve model learning.

Future Scope and Improvements

1. **Hyperparameter Optimization:** Further refining Random Forest and XGBoost models using Grid Search or Bayesian Optimization.
2. **Deep Learning Models:** Exploring neural networks (such as LSTMs or feedforward neural networks) to improve predictions.
3. **More Features:** Adding student behavioral data, attendance trends, or psychological factors to enhance predictive accuracy.
4. **Real-time Predictions:** Implementing this model into an interactive application for real-time grade prediction.
5. **K-Fold Cross Validation:** Ensuring model generalizability by using more robust validation techniques.

Final Thoughts

This project successfully demonstrated the power of machine learning in predicting student performance. By leveraging different models, we were able to determine that **Random Forest is the best-suited model** for this dataset, offering strong predictive power with minimal error.

Future improvements, such as deep learning integration and feature engineering, can further enhance the model's accuracy and real-world applicability.