



Institute of  
Space Technology

# **Machine Learning**

## **Project Report**

### **Group Members:**

Uzair Rauf	210201094
Malik Muhammad Aashir Awan	210201032
Abdullah Ijaz	210201055

**Section:** CS-02-A

**Submitted to:** Ma'am Ammara Yaseen

# **Project Title: Predicting Student Grades: Feature Importance, Model Performance, and GUI Implementation**

## **1. Executive Summary**

The Student Grade Prediction System is an advanced machine learning-based application developed to predict students' academic performance by evaluating a combination of demographic, academic, and contextual features. This system includes two core prediction functionalities: Grade Classification and CGPA Prediction.

- Grade Classification predicts whether a student falls within a specific grade range based on their academic and contextual data.
- CGPA Prediction forecasts a student's final CGPA score using historical data and performance indicators.

The system provides a comprehensive approach by integrating sophisticated data preprocessing techniques, robust machine learning algorithms, and an interactive user interface for ease of use. The models built in this system not only predict performance but also analyze the importance of different features affecting the final predictions.

The backend of the system utilizes several classification and regression algorithms to predict grades and CGPA. These predictions can help educators and administrators identify at-risk students and recommend interventions, thereby improving overall academic support strategies.

## **2. System Architecture and Technical Framework**

### **2.1 Technology Stack**

- **Programming Language:** Python 3.11.4

- **Web Application Framework:** Streamlit (for real-time, interactive user interface)
- **Machine Learning Libraries:**
  - **scikit-learn:** For model training, preprocessing, and evaluation.
  - **pandas:** Used for data manipulation and transformation.
  - **NumPy:** Performs numerical operations and matrix handling.
  - **joblib:** Efficient serialization and saving of trained models.

2.2 Architectural Components

The architecture follows a modular design approach consisting of the following layers:

1. **Data Input Layer:** Collects the user input features (e.g., gender, age, study hours, etc.).
2. **Preprocessing Module:** Cleans, normalizes, and encodes the input data for use in machine learning models.
3. **Machine Learning Models:** Implements both classification and regression models.
4. **Prediction Engine:** Makes real-time predictions based on the input data.
5. **User Interface:** Built with Streamlit, the front-end interface allows users to input data, select prediction modes, and view results.

3. Feature Engineering and Data Preprocessing

3.1 Input Features

The system uses nine key features that influence student performance, categorized as follows:

Feature Category	Specific Features	Data Type
Demographic	Gender, Age	Categorical, Numerical
Academic History	10th Grade Score, 12th Grade Score	Numerical

Performance Indicators	Attendance Percentage, Study Hours	Numerical
Environmental Factors	Extracurricular Activities, Family Support, Internet Access	Categorical

Each of these features has been identified based on their significant contribution to predicting academic performance.

### 3.2 Preprocessing Techniques

Before feeding data into the models, several preprocessing techniques were applied:

- **Numerical Normalization:** The StandardScaler was used to scale numerical features to the same range, ensuring equal treatment by the models.
- **Categorical Encoding:** LabelEncoder was used to convert categorical features (such as gender or extracurricular participation) into numerical values.
- **Missing Values Handling:** Techniques like mean imputation were used for numerical features with missing values, ensuring complete datasets for model training.
- **Model Persistence:** Trained models and the preprocessing pipeline are saved using joblib, which allows the model to be loaded and reused without retraining.

## 4. Machine Learning Modeling

### 4.1 Model Architectures

The system uses both classification and regression approaches to predict student performance:

1. **Classification Model** (for grade prediction):
  - **Random Forest:** A robust algorithm that aggregates predictions from multiple decision trees.
  - **Support Vector Machines (SVM):** A powerful method for classification tasks, effective for high-dimensional data.

- **Gradient Boosting Classifiers:** An ensemble technique that builds strong models by combining weak learners.

## 2. **Regression Model** (for CGPA prediction):

- **Linear Regression:** A simple model that works well when the relationship between input features and output is linear.
- **Ridge and Lasso Regression:** Regularized versions of linear regression that penalize large coefficients.
- **Decision Tree Regression:** A non-linear model that splits data into subsets to predict continuous values.

## 4.2 Model Evaluation Considerations

The models were evaluated using the following techniques and metrics:

- **Cross-validation** was performed to assess model performance and reduce overfitting.
- **Classification Metrics:** Accuracy, Precision, Recall, F1-Score
- **Regression Metrics:** Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)

Both models underwent thorough evaluation to ensure the most accurate predictions possible.

## 5. User Interface Design

### 5.1 Design Principles

The user interface (UI) was designed with the following principles in mind:

- **Simplicity and Usability:** The interface allows users to easily input data and obtain predictions without technical knowledge.
- **Interactivity:** Real-time feedback on the predictions based on the input data enhances user experience.
- **Responsive Design:** The interface adjusts to different screen sizes, ensuring a seamless experience on desktops and mobile devices.

### 5.2 Key UI Features

- **Onscreen Options Selection:** A sidebar allows the user to choose between grade classification and CGPA prediction modes.

- **Dynamic Input Fields:** Input fields appear based on the prediction mode selected, with real-time validation.
- **Styled Prediction Displays:** Predictions are shown with accompanying explanations and visual elements for clarity.
- **Interactive Visuals:** The UI also provides visual aids such as bar graphs to highlight feature importance, giving users more context around the predictions.

## 6. Performance Insights and Feature Importance

### 6.1 Potential Predictive Factors

The system identifies several key factors that significantly affect grade and CGPA predictions:

- **Academic Performance:** Scores from prior grades (10th and 12th), study hours, and attendance are the top contributors to the predictions.
- **Contextual Influences:** Family support, internet access, and extracurricular involvement also play a vital role.

### 6.2 Feature Importance Analysis

Using techniques like Random Forest feature importance and SHAP (Shapley Additive Explanations), we can analyze the contribution of each feature:

- **10th Grade Score:** Shows a high predictive power, acting as a direct indicator of academic ability.
- **Study Hours:** Plays a moderate to high role in predicting grades, showing the student's commitment to learning.
- **Attendance:** Has a moderate effect, reflecting engagement and presence in class.
- **Family Support:** While moderate, it still contributes as an indirect motivational factor.

## 7. Limitations and Future Enhancements

### 7.1 Current Constraints

- **Model Generalization:** The models rely on pre-trained datasets, which may not generalize well for students from different educational backgrounds.
- **Feature Representation:** Categorical features like family support and extracurricular activities are simplified and may not capture the full complexity.
- **Contextual Limitations:** The system may not fully account for external factors that could affect student performance, such as emotional or psychological factors.

### 7.2 Recommended Improvements

1. **Advanced Feature Engineering:** Future iterations should explore more granular features, like parental involvement and access to learning resources.
2. **Ensemble Methods:** Combining multiple models could improve prediction accuracy by leveraging the strengths of each algorithm.
3. **Granular Categorization:** Instead of predicting broad grade categories, the system could predict performance at a more detailed level, such as specific subject scores.
4. **Contextual Data Integration:** Incorporating factors such as mental health or peer influence could provide more personalized predictions.
5. **Confidence Interval Estimation:** Introducing uncertainty in predictions could help users understand the reliability of the results.

## 8. Ethical Considerations

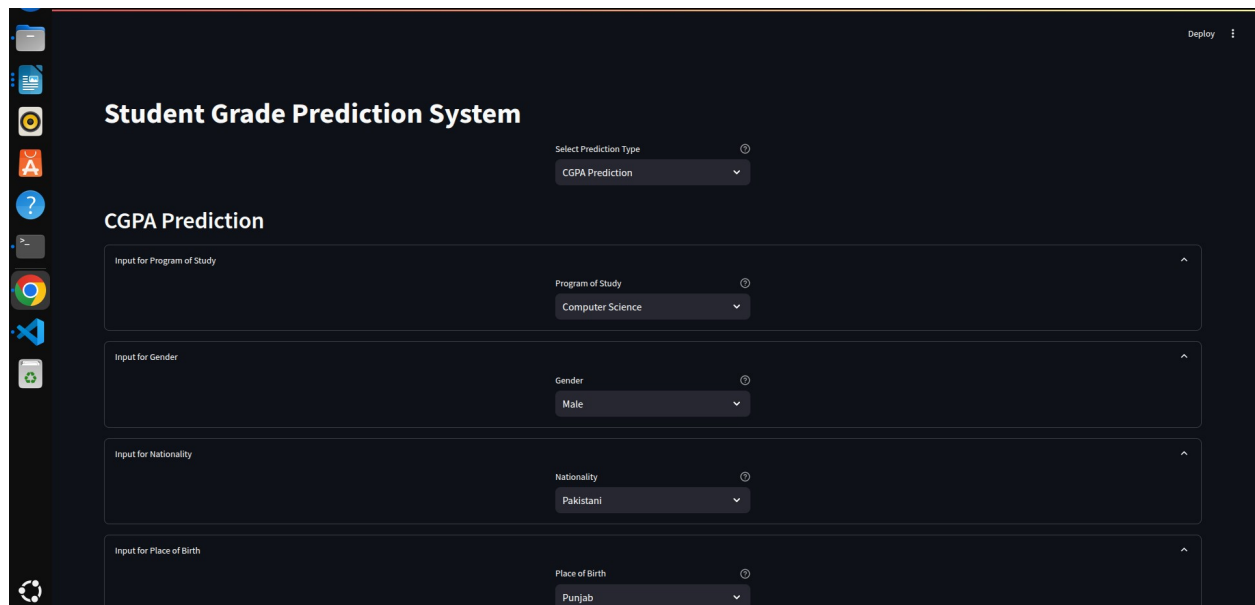
### 8.1 Prediction Interpretation Guidelines

- **Probabilistic Nature:** Emphasize the probabilistic nature of predictions, ensuring students and educators understand that predictions are not deterministic.

- **Non-discriminatory Use:** The system should not be used to make decisions that could unfairly disadvantage students based on predicted outcomes.
- **Supportive Role:** Predictions should be viewed as a tool for academic improvement, not as definitive judgments of a student's future success.

## 9. Output Screenshots

### 9.1 CGPA Prediction



The screenshot displays a web application titled "Student Grade Prediction System" with a "Deploy" button in the top right corner. The main heading is "CGPA Prediction". Below this, there are four input sections, each with a label, a dropdown menu, and an expand/collapse arrow:

- Select Prediction Type:** A dropdown menu showing "CGPA Prediction".
- Input for Program of Study:** A dropdown menu showing "Computer Science".
- Input for Gender:** A dropdown menu showing "Male".
- Input for Nationality:** A dropdown menu showing "Pakistani".
- Input for Place of Birth:** A dropdown menu showing "Punjab".



Deploy

Input for I often come across health issues that effect my academic performance:

I often come across health issues that effect my academic performance.

3.00

Input for My preferable mode of study is:

My preferable mode of study is:

Group study

Input for My preferable time of study is:

My preferable time of study is:

Late night

Input for My overall mobile usage (daily) for non-academic purpose is limited to:

My overall mobile usage (daily) for non-academic purpose is limited to:

3-4 hours

Predict CGPA

Predicted CGPA: 3.26

## 9.2 Grade Prediction

Deploy

### Student Grade Prediction System

Select Prediction Type

Grade Prediction

### Grade Prediction

Input for Program of Study

Program of Study

Computer Science

Input for Gender

Gender

Male

Input for Nationality

Nationality

Pakistani

Input for Place of Birth

Place of Birth

Punjab

Input for My overall mobile usage (daily) for non-academic purpose is limited to:

My overall mobile usage (daily) for non-academic purpose is limited to:

3-4 hours

Input for My preferable time of study is:

My preferable time of study is:

Late night

Input for My preferable mode of study is:

My preferable mode of study is:

Group study

Predict Grade

Predicted Grade: B+

## 10. Conclusion

The Student Grade Prediction System leverages machine learning techniques to provide actionable insights into student performance. By analyzing multiple features, the system offers valuable predictions for both grade classification and CGPA forecasting. The integration of interactive visuals and feature importance analysis enhances the overall utility of the system, making it a powerful tool for educators and students alike.

### Final Recommendation

For continued improvement:

- **Regular Updates:** Ensure that the model remains relevant by frequently updating it with fresh data and feedback.
- **Monitoring and Feedback:** Continuously track the model's performance and incorporate feedback from stakeholders for further refinements.