Mini Project 2 – Students' Performance Prediction

Md Zahid (2302302)

Introduction:

In this project, the goal was to predict students' final grades in a nine-week online machine learning course using data collected from their performance in mini-projects, quizzes, and peer reviews. The dataset contains **107 students'** performance data, with features representing their scores in various tasks (such as quizzes and projects) and their interaction with the online platform. The project required the application of supervised learning techniques to predict the final grade, identify the most important features, and evaluate model performance.

To achieve this, I implemented and compared two machine learning models: **Random Forest** and **Gradient Boosting**. These models were fine-tuned through cross-validation and hyperparameter tuning to ensure optimal performance.

Data Processing:

The dataset consists of students' quiz, project, and peer review scores as well as logs of their interaction with the online learning platform. I began by loading the dataset and checking for missing values. No missing values were found, meaning all features could be utilized for training the models.

Key features include students' performance on quizzes (e.g., Week2_Quiz1), mini projects (e.g., Week3_MP1), and peer reviews (e.g., Week5_PR2). These features, along with interaction logs from weeks 1 to 9 (Week1_Stat0 to Week9_Stat3), were used to predict students' final grades (Week8_Total). After reviewing the data, all features were retained to help the models capture complex patterns in the students' performance.

And there are no missing values in my dataset and all features were retained for model training.

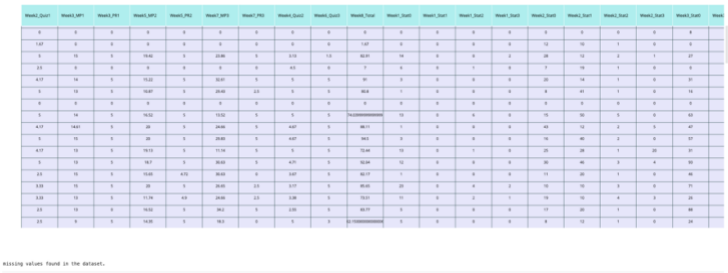


Fig: Data Collecting from CSV File

Data Split:

The dataset was split into training and test sets to allow for proper evaluation of model performance. The data was split with **79.4%** used for training and **20.6%** reserved for testing. This proportion allows the models to learn from a sufficient amount of data while still providing a separate set to evaluate how well the models generalize to unseen data.

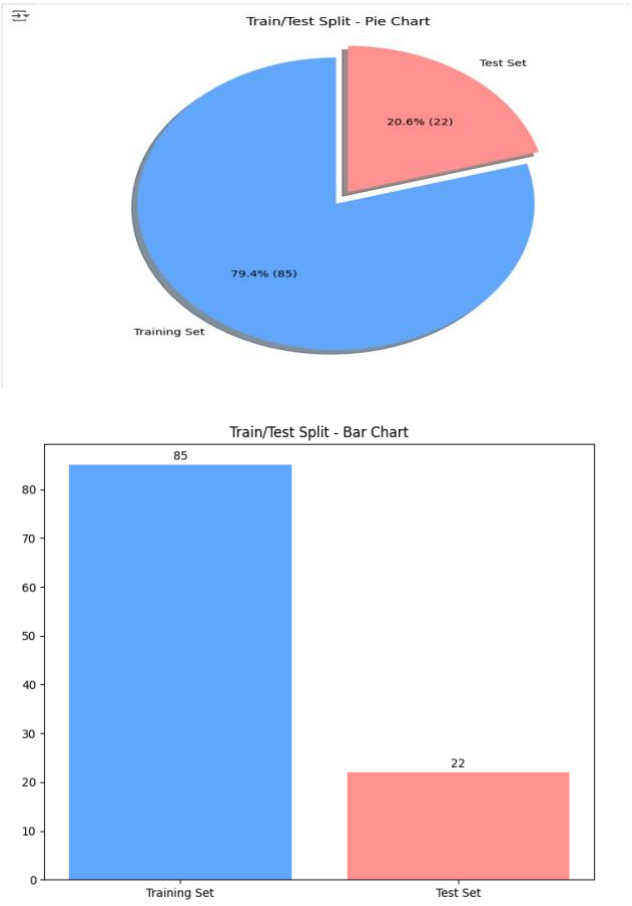


Fig: Training & Test Split Visualization

Model Training:

For testing purpose first, I used 4 models to check what is compatible with my data set and I found this result. And based on this result I selected 2 model.

```
Linear Regression - MSE: 0.9177101994409885, R²: 0.7819480920326762
Random Forest - MSE: 0.047654545454545456, R²: 0.9886770741286205
Gradient Boosting - MSE: 0.027581670994215348, R²: 0.9934464758167892
KNN Regressor - MSE: 0.5709090909090908, R²: 0.8643495336278841
```

Fig: Four model Accuracy result

Two supervised learning models were trained and compared: Random Forest and Gradient Boosting.

1. **Random Forest:**
    - Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of those trees.
    - I tuned the model by adjusting key hyperparameters such as the number of estimators (trees) and the maximum depth of each tree. The best configuration was found to be:
        - n_estimators = 200
        - max_depth = 10
2. **Gradient Boosting:**
    - Gradient Boosting is another ensemble method that builds models sequentially, with each model improving the prediction of the previous one.
    - I used hyperparameter tuning to optimize the number of trees (n_estimators), learning rate, and maximum depth. The best combination of hyperparameters was:
        - n_estimators = 200
        - learning_rate = 0.1
        - max_depth = 5

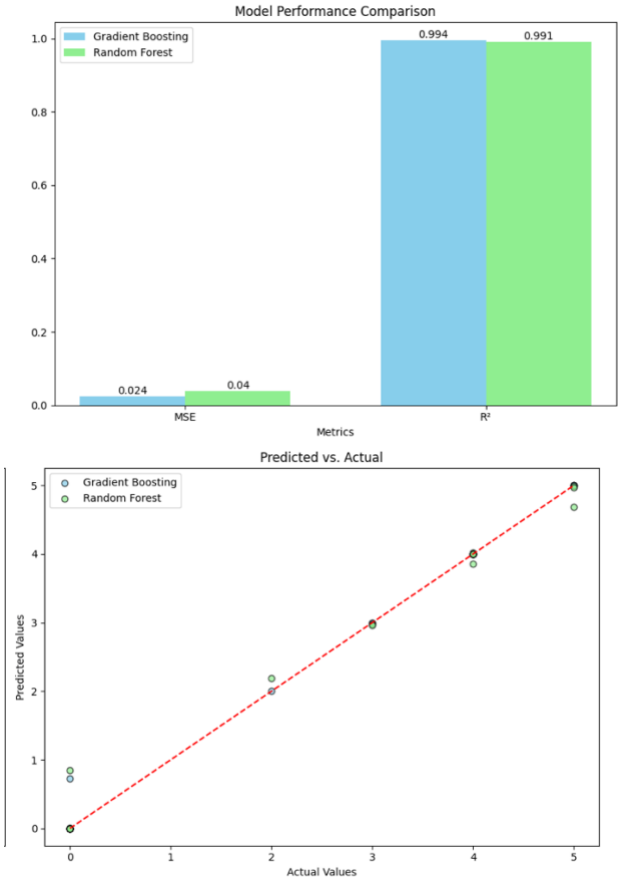


Fig: Model Performance Comparison

Performance Evaluation:

The performance of both models was evaluated using the Mean Squared Error (MSE) and R^2 (coefficient of determination), which measure the accuracy of the predictions.

- Random Forest: After tuning, the Random Forest model achieved: (Ref: Check the Code)
    - MSE: 0.0395
    - R^2: 0.9906
- Gradient Boosting: The Gradient Boosting model outperformed Random Forest, achieving:
    - MSE: 0.0244
    - R^2: 0.9942 (Ref: Check the Code)

These results indicate that both models performed well, but Gradient Boosting had a slight edge in both accuracy (higher R^2) and prediction error (lower MSE). Gradient Boosting's iterative learning method allowed it to better capture the complexities of the dataset.

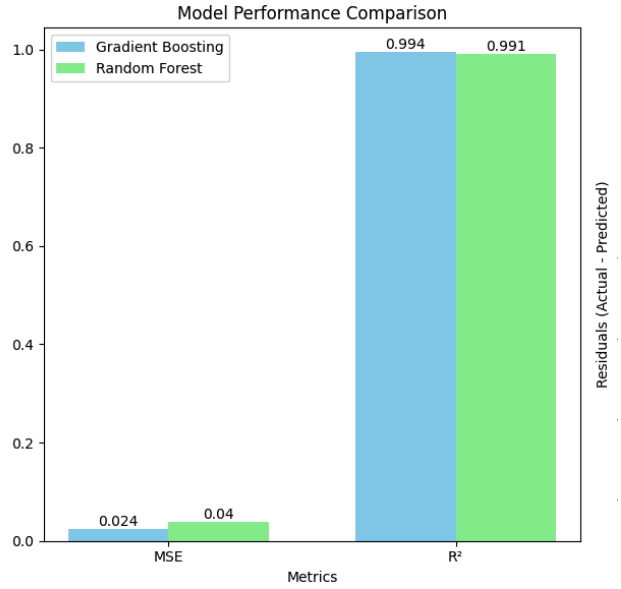


Fig: Model Performance Comparison

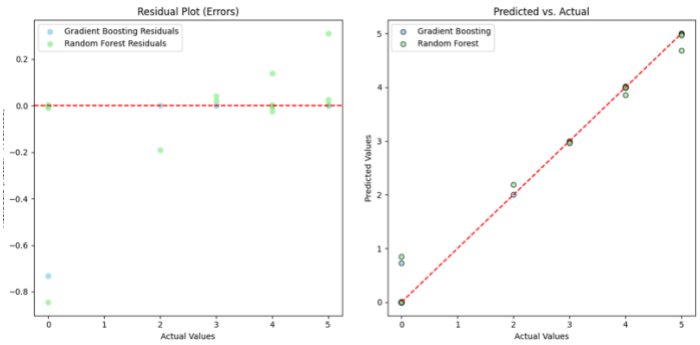


Fig: Model Performance Comparison

Analysis of Visualizing Model Performance:

Three types of visualizations were used to compare the models:

1. Bar Plot: This compared the MSE and R^2 of both models, clearly showing Gradient Boosting's superior performance.
2. Residual Plot: This visualized the prediction errors (residuals) for both models, indicating that the errors for Gradient Boosting were generally smaller.
3. Predicted vs Actual Plot: This showed how closely the predicted values aligned with the actual values for both models. Again, Gradient Boosting demonstrated more accurate predictions, with points more closely aligned along the perfect prediction line.

Important Features:

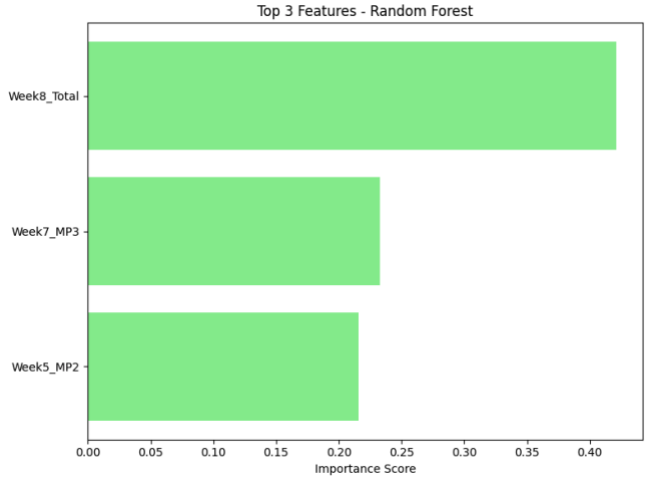
One of the project's goals was to identify the top 3 most important features that contributed to the prediction of the final grade.

- For Random Forest, the top 3 important features were:
    1. Week8_Total
    2. Week5_MP2
    3. Week7_MP3
- For Gradient Boosting, the top 3 important features were:
    1. Week8_Total
    2. Week7_MP3
    3. Week5_MP2

```
Top 3 Important Features from Random Forest:
       Feature  Random Forest Importance
9  Week8_Total                  0.421224
5    Week7_MP3                  0.232550
3    Week5_MP2                  0.215714
Top 3 Important Features from Gradient Boosting:
       Feature  Gradient Boosting Importance
9  Week8_Total                      0.381512
5    Week7_MP3                      0.343299
3    Week5_MP2                      0.275189
```

Fig: Important Features

The consistency between the two models highlights that students' performance in Week 8 (near the end of the course) and their scores on Mini-Project 3 (Week 7) and Mini-Project 2 (Week 5) are the strongest indicators of their final grades. These features likely capture a combination of students' effort and understanding of core material.



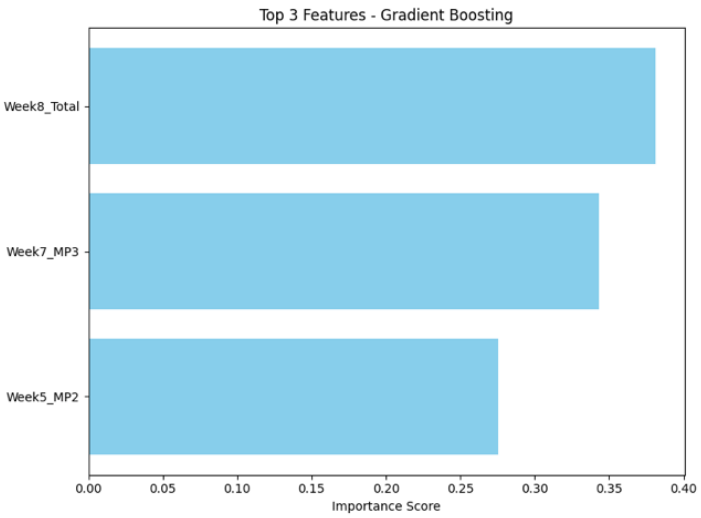


Fig: Top 3 features for Random Forest & Gradient Boosting

A bar plot was created to visualize the top 3 important features for both models. This visualization reinforces the conclusion that students' recent performance (especially in the final weeks of the course) plays a critical role in determining their final grade.

Conclusion:

Through this project, I successfully applied two supervised learning models to predict students' final grades based on a range of features, including their quiz scores, project scores, and interaction logs. After tuning both models, Gradient Boosting emerged as the better model, offering superior accuracy and lower prediction error compared to Random Forest.

During this project, I faced a few challenges, but I was able to work through them and improve the results.

1. **Overfitting**: At first, the models, especially Random Forest, were performing too well on the training data but not as well on the test data. This meant the models were overfitting—learning the training data too perfectly but struggling to generalize to new data. To fix this, I used **hyperparameter tuning**, which helped me find the best settings for the models, like how many trees to use and how deep they should go. This made the models perform better on the test data.
2. **Complexity of Models**: Gradient Boosting turned out to be the most accurate model, but it's a more complex model compared to something like Linear Regression, which is easier to understand. To make sense of how Gradient Boosting was making its predictions, I looked at **feature importance** to see which features (like quiz and project scores) had the biggest impact on the final grade predictions. This helped me interpret the results better.
3. **Choosing the Right Metric**: Initially, I relied on **R^2** to evaluate how well the models were doing. But I realized that while R^2 gives an idea of the overall fit, it wasn't telling me how far off the predictions were in real terms. So, I started focusing more on **Mean Squared Error (MSE)**, which shows how much the model's predictions differ from the actual values. This gave me a clearer picture of how well the models were predicting grades, especially when there were outliers.

In the end, **Gradient Boosting** was the best model. It captured complex patterns in the data and gave the most accurate predictions. The project showed that machine learning models like Random Forest and Gradient Boosting can be very useful in education, helping predict student performance and allowing teachers to step in and help when needed.