Midterm Assignment

Student ID: 246370037
Name: Eyimofe Orimolade Okikiola
Program: Software Engineering
Course Code: SEN 814
Date: April 16, 2025

1. Approach and Model Selection

Data Preprocessing & Feature Engineering

- **Data Loading & Exploration:**
  Two datasets were provided: one clean dataset (for reference) and one with missing values (used for modeling). The datasets were cleaned by trimming extra spaces from column names and exploring the distributions of variables using histograms and summary statistics.
- **Handling Missing Data:**
  To avoid data leakage, the data was split into training and test sets before applying mean imputation on the training data. The same imputation strategy was then applied to the test set.
- **Feature Engineering:**
  In addition to scaling the original features, polynomial feature expansion (degree 2) was applied to capture non-linear relationships. Subsequently, the top 10 polynomial features were selected using the `SelectKBest` method with the `f_regression` scoring function.

Model Selection

Several regression models were explored to predict the final exam score:

- **Linear Regression:** A baseline model.
- **Decision Tree Regressor:** To capture non-linear splits in the data.
- **Support Vector Regression (SVR):** With a radial basis function kernel and tuned hyperparameters.
- **Random Forest Regressor:** An ensemble method that aggregates multiple decision trees.
- **Gradient Boosting Regressor:** Another ensemble approach that builds trees sequentially to reduce errors.
- **Linear Regression (with Polynomial Features):** To take advantage of engineered non-linear relationships.
- **Stacking Regressor:** Combined predictions from multiple models (Linear Regression, Decision Tree, SVR, Random Forest) with a Gradient Boosting final estimator.
- **Ridge Regression:** Used regularization to mitigate overfitting; hyperparameters were tuned using GridSearchCV.

2. Performance Metrics

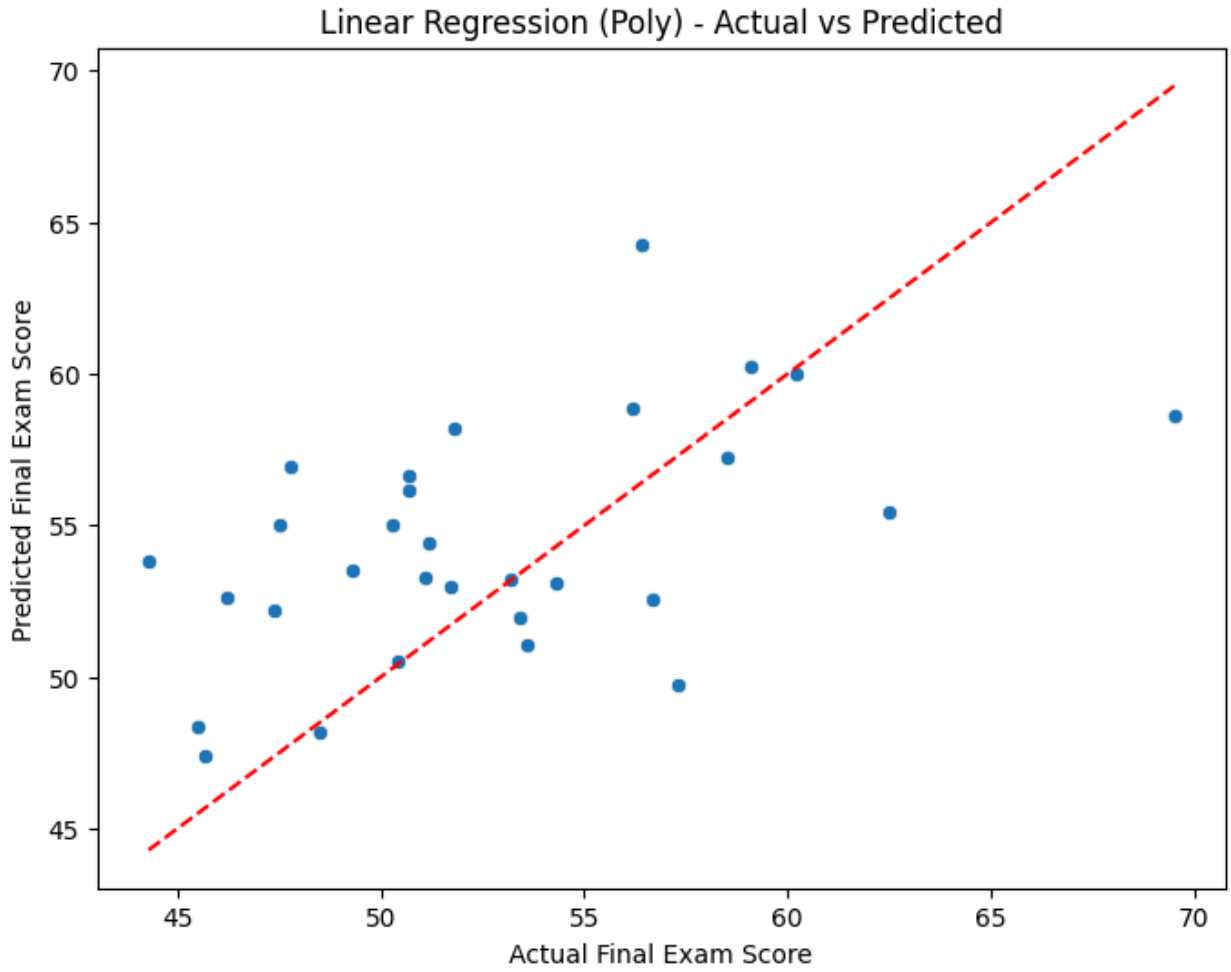
Each model was evaluated using the following metrics:

- **Root Mean Squared Error (RMSE):** Indicates the model's prediction error in the same units as the target.
- **Mean Absolute Error (MAE):** Provides an average error magnitude.
- **R^2 Score:** Reflects the proportion of variance in the target explained by the model.
- **Accuracy% % (derived from R^2):** R^2 was converted to a percentage for ease of comparison.

A summary table of the results is shown below (example values):

Model	RMSE	MAE	R^2	Accuracy %
Linear Regression	5.31	4.09	0.09	8.61
Decision Tree	7.62	6.26	-0.88	-88.34
SVR	5.18	4.43	0.13	12.92
Random Forest	5.81	5.13	-0.09	-9.48
Gradient Boosting	6.11	5.10	-0.21	-21.24
Stacking Regressor	5.60	4.64	-0.02	-1.94
Ridge Regression	5.25	4.12	0.10	10.44
Linear Regression (Poly)	5.14	4.12	0.14	14.24

3. Interpretation of Results



- **Best Performing Model:**
  Based on the evaluation metrics (particularly R^2 and RMSE), the best-performing model was **Linear Regression (Poly)**. This model demonstrated the strongest ability to explain the variance in final exam scores.
- **Impactful Features:**
  Polynomial features engineering highlighted that certain interactions (e.g., between study hours and attendance) and squared terms contributed significantly to the model's predictive power. This suggests that the relationship between the predictors and the final exam score is non-linear.

- **Areas for Improvement:**
    - **Larger Dataset:** Training with more data could help improve model generalizability.
    - **Advanced Feature Engineering:** Exploring higher-degree polynomial features or domain-specific feature interactions may capture additional nuances.
    - **Model Enhancements:**
        - **Stacking Regressor:** Combining models in an ensemble approach has the potential to harness strengths from multiple algorithms.
        - **Regularized Models:** Ridge Regression helped in mitigating overfitting, and further exploration with Lasso or ElasticNet could be beneficial.
    - **Hyperparameter Tuning:** More exhaustive tuning (e.g., through RandomizedSearchCV) could lead to further improvements in model performance.

4. Interactive Web Application

To demonstrate the functionality and real-world applicability of the final model, an interactive web application was developed and deployed using **Streamlit**.

Users can input student performance metrics such as study hours, attendance, assignment completion, and midterm scores to receive a predicted final exam score in real time. The app uses the best-performing model (Linear Regression with Polynomial Features) and follows the same preprocessing pipeline described earlier.

🔗 **Access the Web App:**
https://lyon-zas-student-performance-predictor-app-z6mp7j.streamlit.app/