Project Report

Course: CSE422

Artificial Intelligence

Submitted By

Nafiz Siddiqui Adnan [20301016]

MD Reaz Uddin [20101228]

Submitted To
Syed Zamil Hasan Shoumo
Lecturer
Department of Computer Science and Engineering
BRAC University
&
Zahin Wahab
Lecturer
Department of Computer Science and Engineering
BRAC University

Submission Date: 29-04-2024

Table of Contents

Introduction

The importance of predicting the student scores is significant for both educator and student for further improvements. This project aims to predict student scores through regression models so that in the near future students and teachers can identify factors that help students to achieve noteworthy results that can be implemented for the weaker students through corrective measures. Overall, our motivation to do the student score prediction is to update the students with proper measurements so that they can compete with each other neck to neck. Also, we aim to understand the influence of social, family, and academic backgrounds on student success. By identifying these factors, educational institutions can create targeted interventions to help students improve their academic outcomes.

Dataset Description

The dataset used in this study comes from the UCI Machine Learning Repository, specifically the "Student Performance" dataset. This dataset comprises data from two Portuguese schools for a Portuguese language course. It includes 33 attributes for 649 students, ranging from demographic information to academic performance.

Link: https://archive.ics.uci.edu/dataset/320/student+performance
Reference: Cortez,Paulo. (2014). Student Performance. UCI Machine Learning Repository. https://doi.org/10.24432/C5TG7T.

Dataset Characteristics:

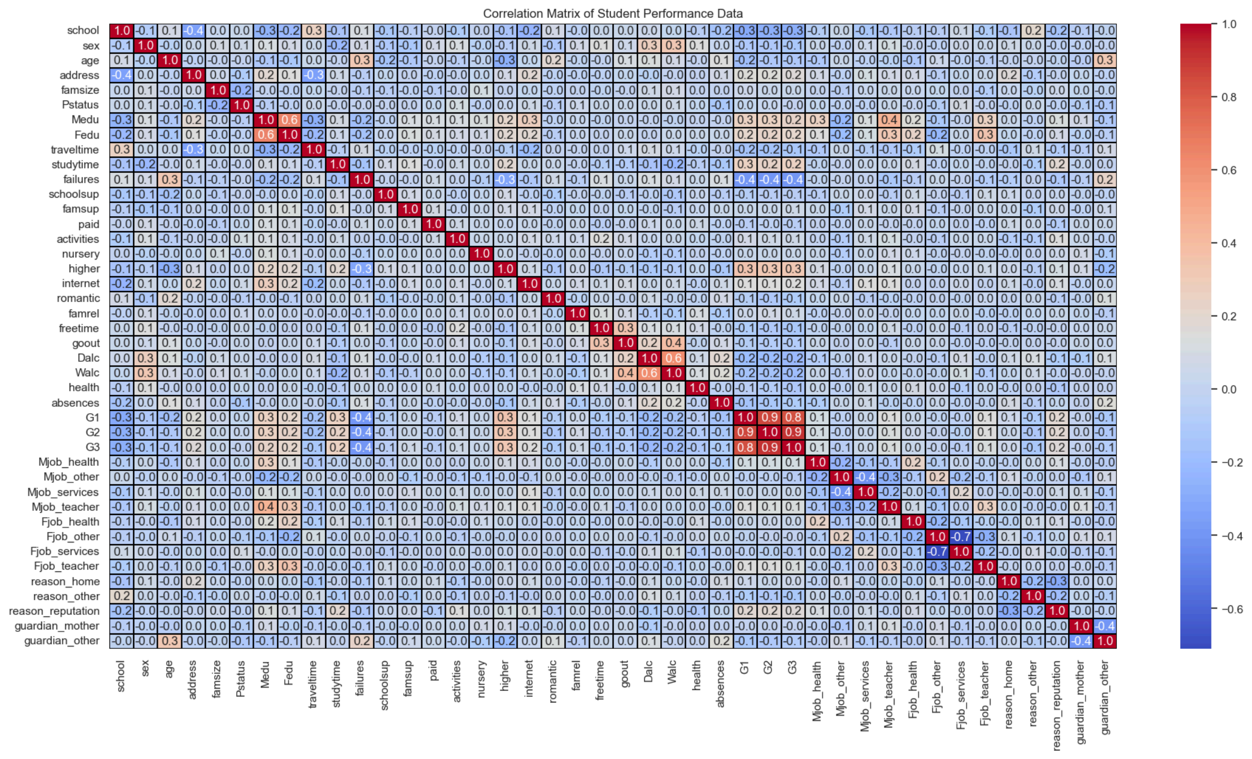
- Number of Features: The dataset contains a total of 33 feature vectors.
- Problem Type: This is a regression problem, as the main variable to predict, the final grade (G3), is continuous in nature, ranging from 0 to 20.
- Number of Data Points: There are 649 instances, each representing a student.
- Types of Features: The dataset consists of both quantitative and categorical features.
    - **Quantitative:** Age, travel time, study time, number of past class failures, family relationships, free time, going out, workday and weekend alcohol consumption,

current health status, and number of school absences, as well as the three grade attributes (G1, G2, and G3).

- Categorical: School, sex, address, family size, parent's cohabitation status, mother's and father's education levels, mother's and father's jobs, reason for choosing the school, guardian, whether the student has extra educational support, wants to take higher education, has internet access at home, is in a romantic relationship, participated in extra-curricular activities, and attended nursery school.

Correlation Analysis:

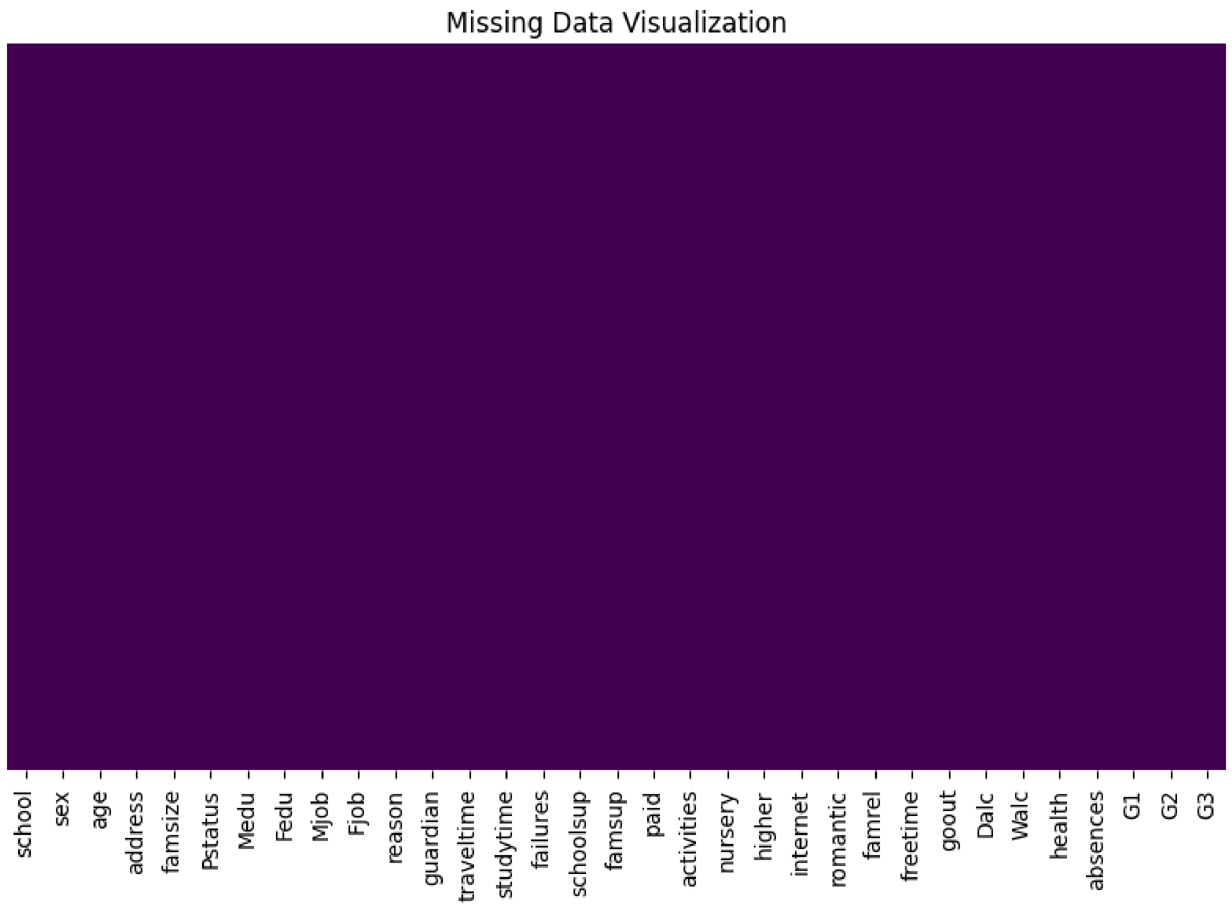
A heatmap was generated using the Seaborn library to analyze the correlation between all the features, including input and output attributes. This visualization helps in understanding the relationships and potential collinearity between different variables, which is critical for the model selection and feature engineering phases.

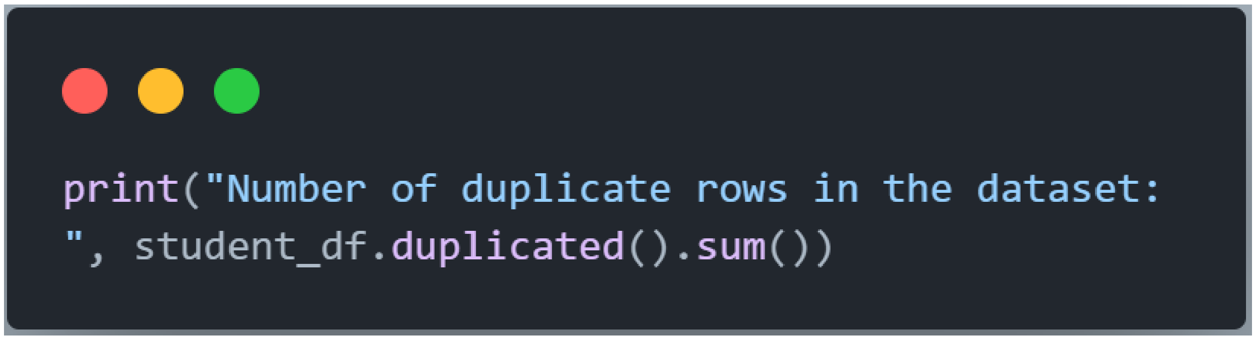


Dataset Preprocessing

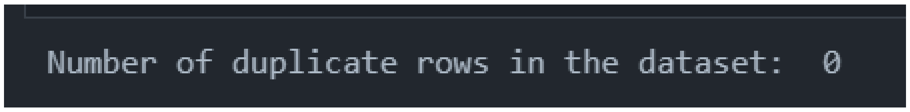
Preprocessing of the dataset is a crucial step in data analysis and predictive modeling. The integrity and quality of the data must be ensured to achieve reliable results. In this stage, we dealt with the missing values, duplicate data points, and categorical data.

Missing Values and Duplicate Data: Upon analyzing our dataset, we did not discover any duplicate entries or missing values. So, in this part, we did not have to do any cleaning or imputation.





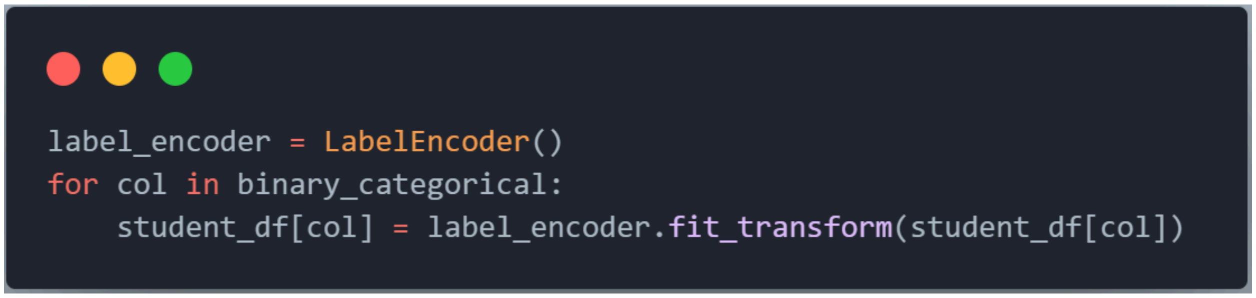
```python
print("Number of duplicate rows in the dataset: ", student_df.duplicated().sum())
```



```
Number of duplicate rows in the dataset:  0
```

Categorical Values: In the pre-processing of the dataset, two types of categorical variables were encountered: binary and non-binary. These categorical variables pose a challenge for machine learning models, as they require numerical input for computations.
There were a total of 13 binary categorical columns and 4 non-binary categorical columns identified in the dataset. Binary categorical variables have only two categories and include features such as 'school', 'sex', 'internet', and 'romantic'. Non-binary categorical variables have more than two categories and include features such as 'Mjob', 'Fjob', 'reason', and 'guardian'. These variables needed to be encoded to numerical values before they could be used for machine learning algorithms.

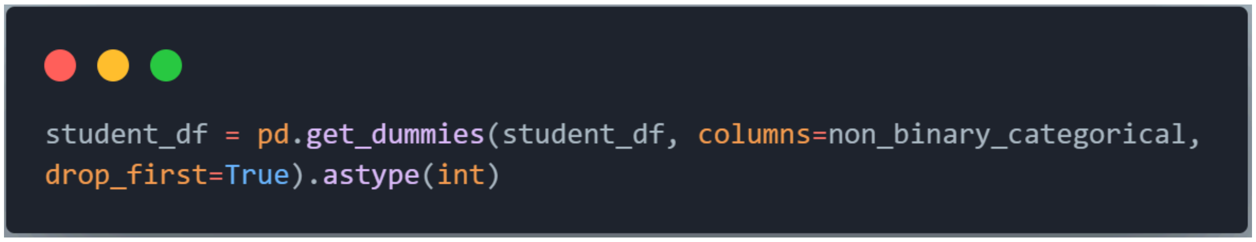
To address the problem, label encoding was applied to binary categorical variables. Label encoding is suitable for binary categorical variables as it converts them into a simple numerical binary format (0 or 1), which aligns with their nature of having only two categories.



```python
label_encoder = LabelEncoder()
for col in binary_categorical:
    student_df[col] = label_encoder.fit_transform(student_df[col])
```

For non-binary categorical variables, one-hot encoding was chosen. One-hot encoding is a preferred method as it creates new binary columns for each category of the non-binary variables.

This prevents the introduction of artificial ordinality, which could mislead the model into assuming a natural order in the categories where there isn't one. The first category was dropped to avoid the dummy variable trap and ensure model interpretability.



```
student_df = pd.get_dummies(student_df, columns=non_binary_categorical,
drop_first=True).astype(int)
```

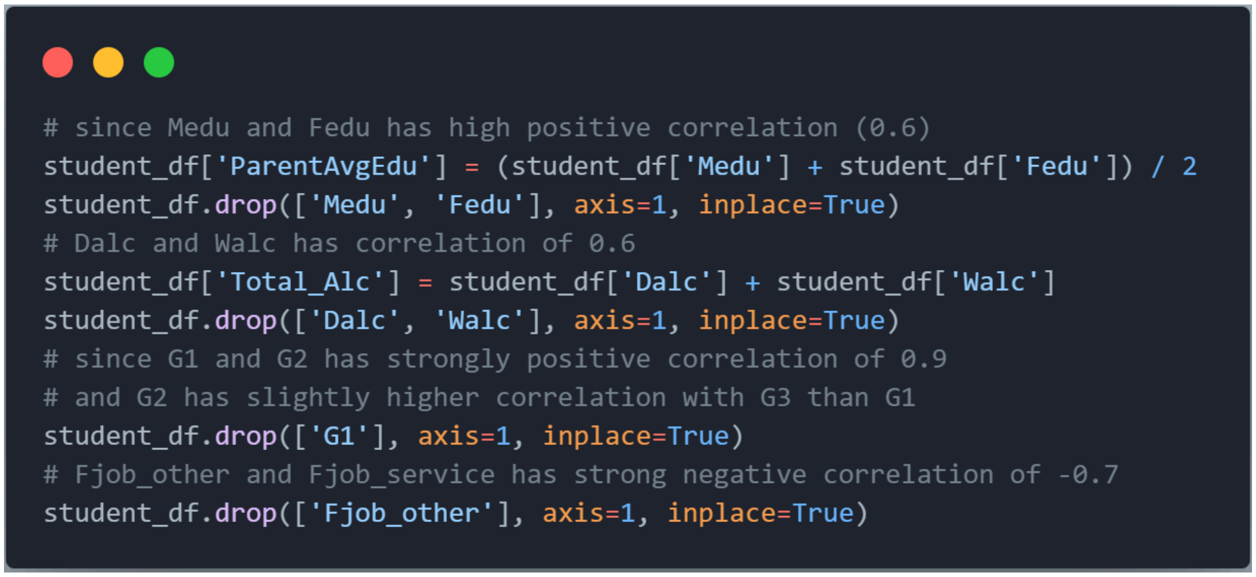
After encoding, the shape of the dataframe changed to (649, 42), indicating that additional columns were created for each category of the non-binary variables, except for the first category in each original column, which was excluded to avoid multicollinearity.

Feature Engineering

In the feature engineering phase, we tackled the issue of multicollinearity, which can affect model performance by making the model parameters unstable and difficult to interpret. Multicollinearity was addressed by creating new features and eliminating redundant ones, based on the correlation observed between existing features.

New Features: We optimized our dataset by creating composite features and removing redundant ones:

- ParentAvgEdu: Averaged 'Medu' and 'Fedu' due to their high correlation.
- Total_Alc: Summed 'Dalc' and 'Walc' to consolidate alcohol consumption data.
- Dropped the 'G1' column, as 'G2' showed a stronger relationship with the target variable and ‘G1’ and ‘G2’ had a strongly positive correlation with each other.
- Removed 'Fjob_other' because of its significant negative correlation with 'Fjob_service'.

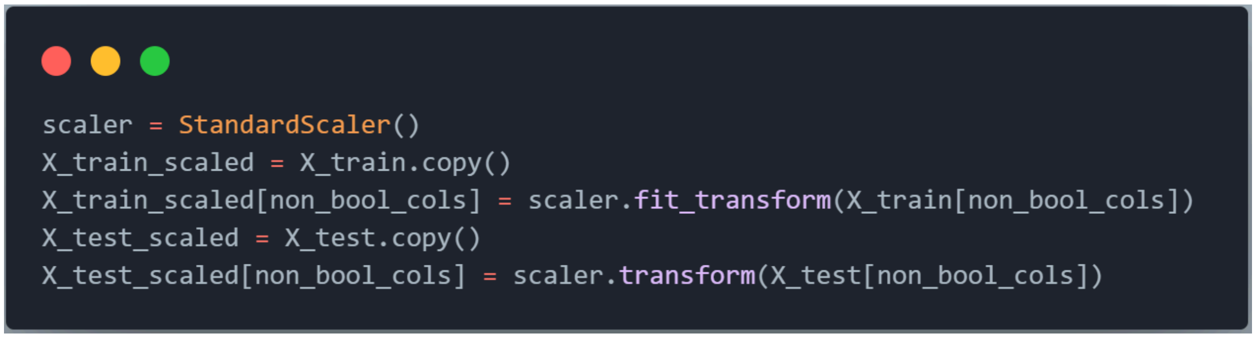


```
# since Medu and Fedu has high positive correlation (0.6)
student_df['ParentAvgEdu'] = (student_df['Medu'] + student_df['Fedu']) / 2
student_df.drop(['Medu', 'Fedu'], axis=1, inplace=True)
# Dalc and Walc has correlation of 0.6
student_df['Total_Alc'] = student_df['Dalc'] + student_df['Walc']
student_df.drop(['Dalc', 'Walc'], axis=1, inplace=True)
# since G1 and G2 has strongly positive correlation of 0.9
# and G2 has slightly higher correlation with G3 than G1
student_df.drop(['G1'], axis=1, inplace=True)
# Fjob_other and Fjob_service has strong negative correlation of -0.7
student_df.drop(['Fjob_other'], axis=1, inplace=True)
```

After the modification, the dataset dimensionality was streamlined to 38 features from 42.

Feature Scaling: The feature scaling process is essential, especially for models that are sensitive to the magnitude of input values, like SVM. The goal is to ensure that numerical features have equal weight and the model is not biased toward features with larger magnitudes.

For this dataset, feature scaling was conducted on the 12 non-boolean numerical features, including the newly engineered features 'ParentAvgEdu' and 'Total_Alc'. The StandardScaler from scikit-learn was employed to standardize features by removing the mean and scaling to unit variance:



```
scaler = StandardScaler()
X_train_scaled = X_train.copy()
X_train_scaled[non_bool_cols] = scaler.fit_transform(X_train[non_bool_cols])
X_test_scaled = X_test.copy()
X_test_scaled[non_bool_cols] = scaler.transform(X_test[non_bool_cols])
```

The scaling was applied separately to the training and test sets to prevent data leakage. The scaled features then replaced the original values in X_train_scaled and X_test_scaled, ensuring that the scaling transformation would be consistent across both datasets.

Dataset Splitting

The dataset was divided into training and testing subsets using a random split to prevent any sampling bias, ensuring that the models would be evaluated on a varied sample of data. This randomness is key in regression tasks to ensure the model can generalize well to new, unseen data.

We used a 70:30 split ratio, allocating 70% of the data for model training and 30% for validation testing. This balance allows for sufficient learning while retaining enough data to test the models effectively. Consequently, the training subset contained 454 data points, and the testing subset had 195, with both subsets featuring the same number of predictors.

The final shapes of the training and testing subsets were (454, 37) and (195, 37) respectively, ready for the subsequent model training and evaluation steps in the regression analysis.

Model Training and Testing

After data preparation was done, we employed three different regression models to predict student final grades.

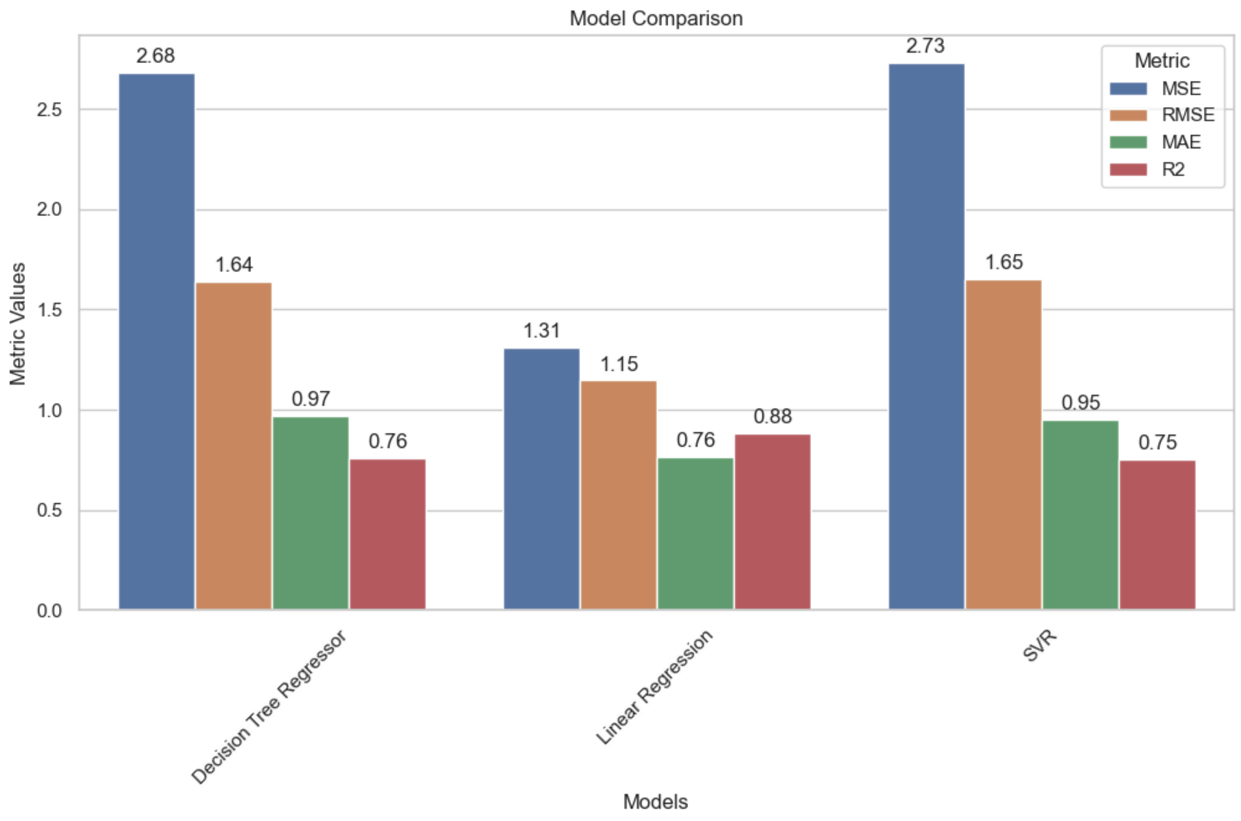
- A Decision Tree Regressor was trained, achieving an RMSE of 1.6377 and an R^2 of 0.7575. It demonstrated an accuracy within a 5% deviation of the true value for 83.08% of the cases.
- The Linear Regression model yielded an RMSE of 1.1458 and an R^2 of 0.8813, with an accuracy within 5% deviation of the true value for 70.77% of the cases.
- Lastly, a Support Vector Machine (SVM) with the SVR estimator provided an RMSE of 1.6520 and an R^2 of 0.7532. It had an accuracy within a 5% deviation of the true value in 73.85% of the instances.

Each model was fitted using the scaled training data and then evaluated on the scaled test data. The performance evaluation suggests that the models have captured the underlying patterns in the data related to student final grades to different extents. Notably, the Linear Regression model demonstrates the most promising performance, as indicated by its lower RMSE (Root Mean Square Error) and higher R-squared scores.

Model Selection/Comparison Analysis

The model selection process involved a thorough comparison of three regression algorithms: Decision Tree Regressor, Linear Regression, and Support Vector Regression (SVR). This comparison was based on key metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2).

The accompanying bar chart provides a visual representation of each model's performance across these metrics. Notably:



- **The Linear Regression** model showcased the lowest RMSE, indicating higher accuracy in predicting the final grades.
- It also scored the highest R^2 value, suggesting that it could explain a larger proportion of the variance in the student final grades than the other models.
- **The Decision Tree Regressor and SVR** models showed competitive performance, yet they were slightly outperformed by the Linear Regression model in terms of both RMSE and R^2.

Based on these findings, the Linear Regression model emerged as the most promising choice for this task, balancing simplicity with predictive power. It's clear from the analysis that while all models have their merits, the simplicity and effectiveness of Linear Regression make it a suitable choice for predicting student performance in this context.

Conclusion

To conclude, this study effectively demonstrated the utility of machine learning in predicting student performance. Linear Regression emerged as the most effective model, illuminating the potential of data-driven insights to inform educational strategies. The skills and techniques applied here are transferable across domains, showcasing the broad applicability of predictive modeling.