Data Analysis and Machine Learning Implementation
Project Documentation

I. Project Overview

This analysis assesses student exam performance using a variety of demographic and socioeconomic parameters. By examining the dataset, we hope to learn more about how gender, race/ethnicity, parental education level, lunch type, and test preparation affect students' math, reading, and writing scores.

The findings will assist in identifying crucial elements influencing academic achievement, which can then be used to inform educational practices and policies to improve student outcomes.

The dataset provides information on student exam performance. The critical user attributes or data elements that will be evaluated are:

1. **Gender**: Male or female. Gender analysis can provide insights into performance variations and preferences between male and female students, allowing for more individualized educational tactics and support services.

2. **Race/Ethnicity**: This categorical variable represents the student's race/ethnicity. Understanding racial and cultural backgrounds can assist in detecting educational discrepancies and inspire policies that promote equity and inclusion.

3. **Parental Level of Education**: This pertains to the highest level of education achieved by the student's parents. This statistic can shed light on the impact of parental education on student achievement, emphasizing the need for focused interventions.

4. **Lunch**: The meal the student received (regular, free, or reduced). Lunch kinds can disclose socioeconomic aspects that influence student achievement and guide resource allocation to help students from low-income homes.

5. **Test Preparation Course**: Indicates whether or not the student completed a test preparation course. Tracking the completion of test preparation courses can determine their effectiveness in improving student performance and informing educational program selections.

6. **Math Score**: A student's math exam score. This statistic is critical for evaluating students' math proficiency, finding areas for development, and customizing math training approaches.

7. **Reading Score**: A student's reading exam score. Analyzing reading scores assists in understanding students' literacy levels, guiding reading programs, and developing targeted reading interventions.

8. **Writing Score**: The student's score on the writing exam. This score provides information about students' writing ability, which informs writing training strategies and identifies the need for more writing support.

The primary goals of this project are to better understand demographic and socioeconomic determinants by examining how gender, race/ethnicity, parental education, lunch style, and test preparation courses affect exam results. The project also aims to identify performance trends by comparing trends across demographic groups and creating a model to predict student performance using the provided attributes.

This analysis is expected to provide insights into performance disparities among different gender and race/ethnicity groups, understanding how parents' educational level correlates with student performance, analyzing how receiving standard or free/reduced lunch and completing a course for test preparation affect exam scores, and predictions about future student performance to help identify at-risk students and provide targeted support.

II. Libraries and Data Handling

Libraries Used

Data Manipulation and Visualization Libraries:

1. **Pandas**: This library is Python's advanced data manipulation and analysis library. It includes data structures like DataFrames and Series, which are necessary for processing and analyzing structured data. This project uses this library for dataset loading, cleaning, manipulation, and transformation.
2. **Seaborn**: A Matplotlib-based library for visualizing statistics. It provides a high-level interface for creating visually appealing and valuable statistical graphs. This is used in this project to generate a variety of plots such as bar charts, pie charts, and heatmaps to view data and get insights.
3. **Matplotlib**: A Python charting toolkit that lets us create static, interactive, and animated displays. It is commonly used to generate plots and charts. It works in collaboration with Seaborn to fine-tune and customize visuals.

Inferential Statistics Libraries:

4. **SciPy**: A package that supports scientific and technical computing. It includes modules for optimization, integration, interpolation, eigenvalue problems, algebraic equations, and other complex computations. In this project, t-tests are used to

compare means between groups, which aids in determining the statistical significance of the data.

5. **Statsmodels**: A library that estimates and tests statistical models. It provides tools for running a variety of statistical tests and models. It is used for more advanced statistical studies and models, such as regression analysis.

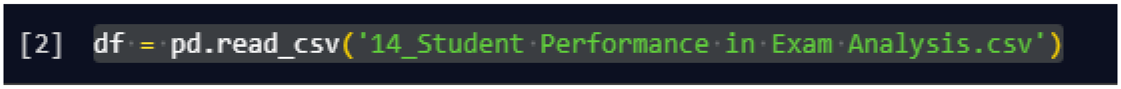
Machine Learning Libraries:

6. **Scikit-Learn**: Scikit-Learn is a machine learning toolkit that offers easy-to-use tools for data mining and analysis and support for various supervised and unsupervised learning methods. It is used in this project to split the dataset into training and testing sets, implement multiple linear regression models, and use ensemble methods for more robust predictions. It performs preprocessing tasks such as categorical data encoding with OneHotEncoder and feature scaling with StandardScaler, applies different preprocessing steps to different columns using ColumnTransformer, and evaluates machine learning model performance using metrics like mean squared error and R².

Data Loading and Preprocessing

- ***Data Loading***: The dataset containing student exam performance is loaded into a Pandas DataFrame from a CSV file, a common practice for data analysis. Using the pd.read_csv() function, the structured data is converted into a DataFrame, enabling powerful data manipulation capabilities within Python.

  The code snippet for loading the data is as follows:



```
[2]  df = pd.read_csv('14_Student Performance in Exam Analysis.csv')
```

This method reads the CSV file 'student_performance.csv' and loads the data into the DataFrame df, facilitating further analysis and processing.

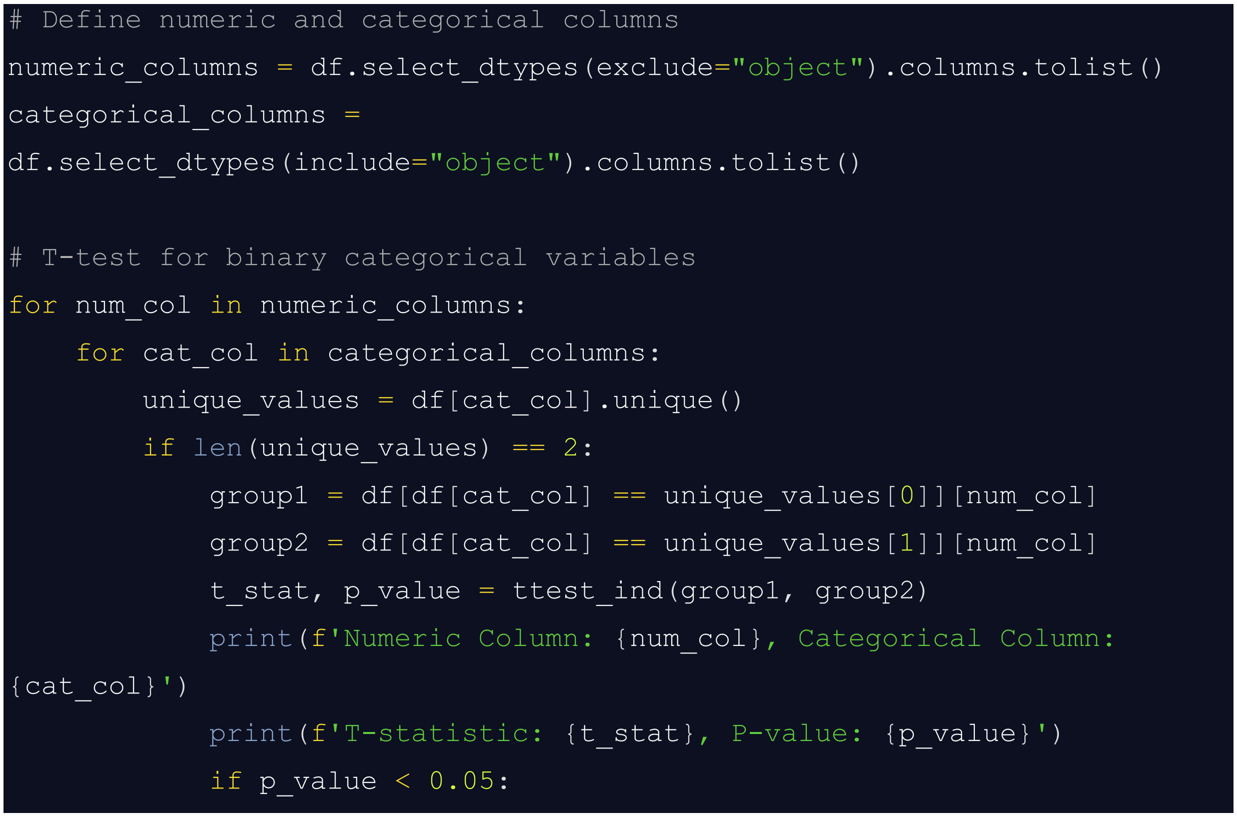
III. Data Analysis Techniques

Descriptive Statistics

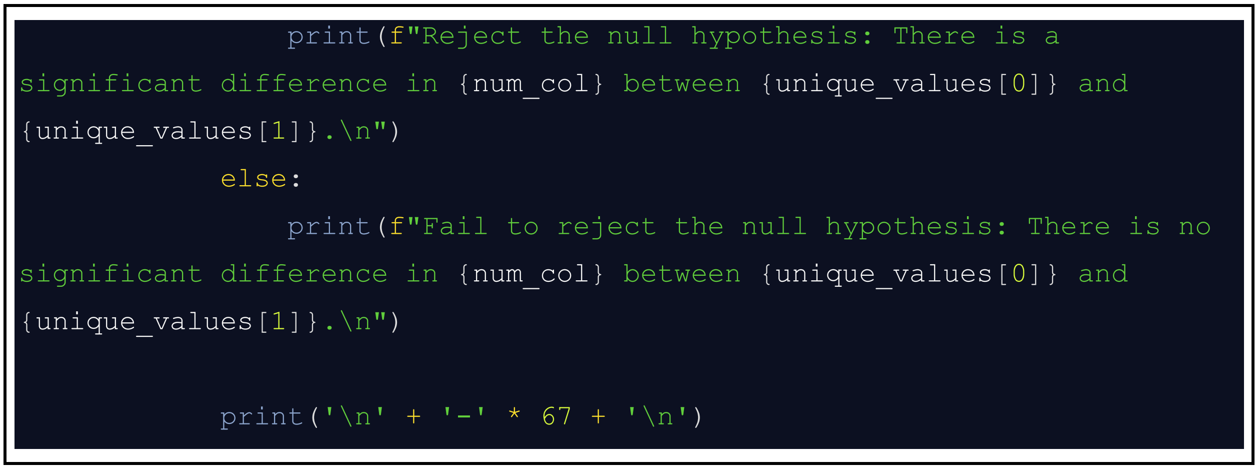
- **df.head()**: This function returns the first five rows of the DataFrame, providing a quick overview of the dataset's initial entries, verifying data loading, understanding its structure, and ensuring correct data import, aiding in data manipulation, analysis, and visualization tasks.
- **df.tail()**: This function returns the last five rows of the DataFrame, confirming the dataset's completeness and identifying any patterns or anomalies, thus ensuring its scope and extent and complementing df.head().
- **df.shape()**: This function returns a tuple indicating the dimensionality of the DataFrame, indicating the number of rows and columns, providing a concise overview of the dataset's size for analysis and visualization.
- **df.describe()**: This function generates descriptive statistics, including central tendency, dispersion, and distribution shape of the dataset, excluding NaN values. Key metrics like mean, median, standard deviation, min, and max are provided for numerical columns, enabling analysts to identify potential patterns or outliers.
- **df.columns()**: This function provides column labels for the DataFrame, enabling analysts to identify attributes, structure and variables and guide data exploration and interpretation, which is crucial for dataset comprehension.
- **df.dtypes()**: This function identifies the data types of each column in a DataFrame, indicating whether it's an integer, float, or object format, aiding analysts in effective data handling and interpretation.

- **df.isnull().sum()**: This function identifies missing values in each column, affecting analysis accuracy and reliability. Analysts can assess dataset completeness and use strategies like imputation or removal to address missing data.
- **df.info()**: This function offers a detailed overview of the DataFrame, including index dtype, column dtypes, non-null values, and memory usage, enabling analysts to assess the dataset's integrity, identify potential issues, and plan subsequent data processing and analysis steps.
- **df.nunique()**: This function calculates unique values in each column, aiding in identifying data diversity, categorical analysis, and anomalies, guiding further analysis and interpretation of data variability and distribution.

Inferential Statistics



```
# Define numeric and categorical columns
numeric_columns = df.select_dtypes(exclude="object").columns.tolist()
categorical_columns =
df.select_dtypes(include="object").columns.tolist()

# T-test for binary categorical variables
for num_col in numeric_columns:
    for cat_col in categorical_columns:
        unique_values = df[cat_col].unique()
        if len(unique_values) == 2:
            group1 = df[df[cat_col] == unique_values[0]][num_col]
            group2 = df[df[cat_col] == unique_values[1]][num_col]
            t_stat, p_value = ttest_ind(group1, group2)
            print(f'Numeric Column: {num_col}, Categorical Column:
{cat_col}')
            print(f'T-statistic: {t_stat}, P-value: {p_value}')
            if p_value < 0.05:
```



```
                print(f"Reject the null hypothesis: There is a
significant difference in {num_col} between {unique_values[0]} and
{unique_values[1]}.\n")
            else:
                print(f"Fail to reject the null hypothesis: There is no
significant difference in {num_col} between {unique_values[0]} and
{unique_values[1]}.\n")

            print('\n' + '-' * 67 + '\n')
```

This code snippet performs T-tests for binary categorical variables to assess whether there is a significant difference in numeric columns between the two categories of the categorical variable. Here's an explanation of how it works:

Define Numeric and Categorical Columns:

- numeric_columns: Selects numeric columns (excluding object dtype) from the DataFrame and converts them into a list.
- categorical_columns: Selects categorical columns (including object dtype) from the DataFrame and converts them into a list.

T-test for Binary Categorical Variables:

- Nested loops iterate over each numeric column (num_col) and each categorical column (cat_col) in the dataset.
- For each numeric column and categorical column pair, unique values of the categorical column are identified.
- Suppose the categorical column has only two unique values (binary categorical variable). In that case, a T-test (independent two-sample T-test) is conducted to compare each category's means of the numeric column.
- The T-statistic and p-value are calculated using the ttest_ind function from SciPy's stats module.

Output Interpretation:

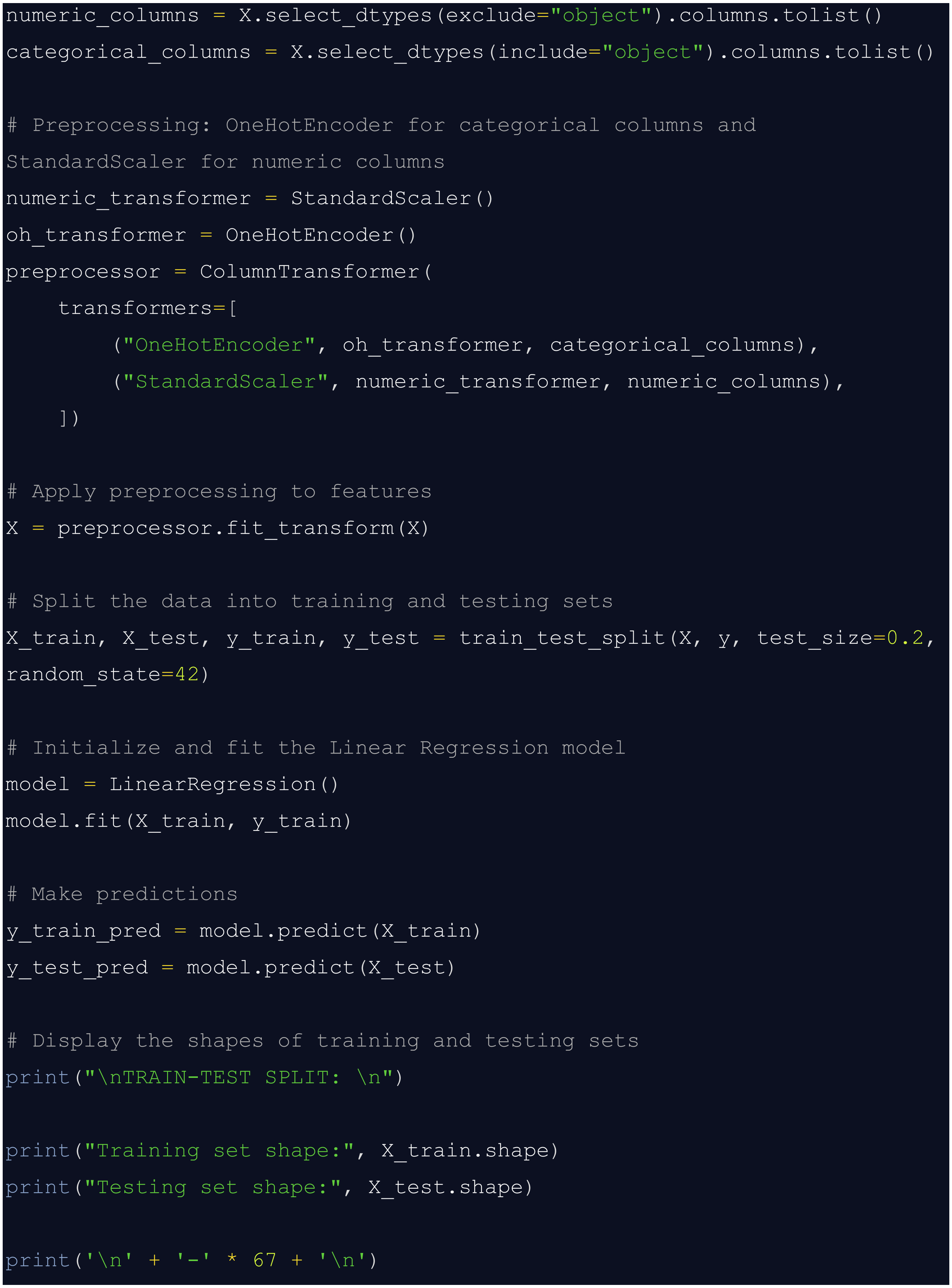
```
Numeric Column: math score, Categorical Column: gender
T-statistic: -5.383245869828983, P-value: 9.120185549328822e-08
Reject the null hypothesis: There is a significant difference in math score between female and male.
--------------------------------------------------------------
Numeric Column: math score, Categorical Column: lunch
T-statistic: 11.837180472914612, P-value: 2.4131955993137074e-30
Reject the null hypothesis: There is a significant difference in math score between standard and free/reduced.
--------------------------------------------------------------
Numeric Column: math score, Categorical Column: test preparation course
T-statistic: -5.704616417349102, P-value: 1.5359134607147415e-08
Reject the null hypothesis: There is a significant difference in math score between none and completed.
--------------------------------------------------------------
Numeric Column: reading score, Categorical Column: gender
T-statistic: 7.959308005187657, P-value: 4.680538743933289e-15
Reject the null hypothesis: There is a significant difference in reading score between female and male.
--------------------------------------------------------------

Numeric Column: reading score, Categorical Column: lunch
T-statistic: 7.451056467473455, P-value: 2.0027966545279011e-13
Reject the null hypothesis: There is a significant difference in reading score between standard and free/reduced.
--------------------------------------------------------------

Numeric Column: reading score, Categorical Column: test preparation course
T-statistic: -7.871663538941468, P-value: 9.081783336892205e-15
Reject the null hypothesis: There is a significant difference in reading score between none and completed.
--------------------------------------------------------------

Numeric Column: writing score, Categorical Column: gender
T-statistic: 9.979557910004507, P-value: 2.019877706867934e-22
```

```
Reject the null hypothesis: There is a significant difference in writing score between female and male.
--------------------------------------------------------------
Numeric Column: writing score, Categorical Column: lunch
T-statistic: 8.009784197834758, P-value: 3.1861895831664765e-15
Reject the null hypothesis: There is a significant difference in writing score between standard and free/reduced.
--------------------------------------------------------------
Numeric Column: writing score, Categorical Column: test preparation course
T-statistic: -10.409173436808748, P-value: 3.68529173524572e-24
Reject the null hypothesis: There is a significant difference in writing score between none and completed.
--------------------------------------------------------------
```

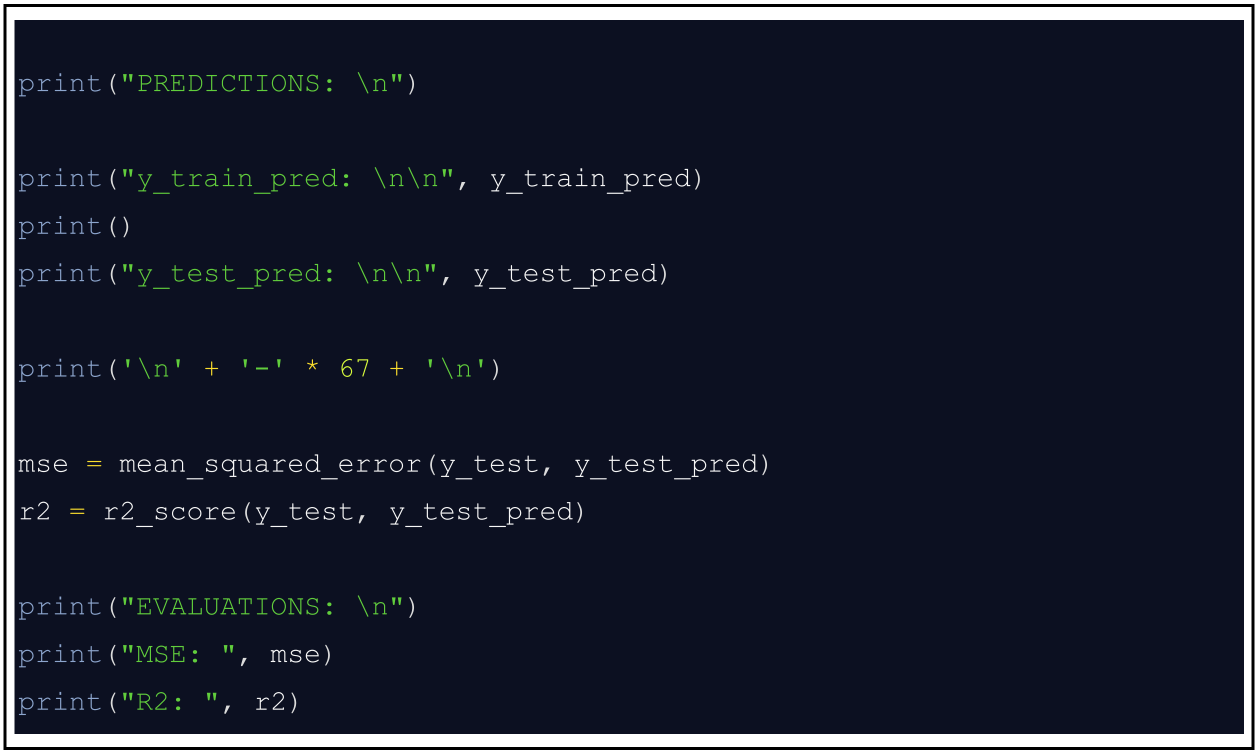
- For each numeric column and categorical column pair, the T-statistic and p-value are printed.
- If the p-value is less than 0.05 (joint significance threshold), the null hypothesis (no difference between the groups) is rejected, indicating a significant difference in the numeric column between the two categories of the categorical variable.
- Suppose the p-value is greater than or equal to 0.05. In that case, the null hypothesis is not rejected, suggesting no significant difference in the numeric column between the two categories of the categorical variable.

Predictive Modeling

```
# Define features (X) and target (y)
# These variables will be used in all machine learning models below
# Edit the column_name for other analysis
X = df.drop(['writing score'], axis=1)
y = df['writing score']

# Define numeric and categorical columns
```



```
numeric_columns = X.select_dtypes(exclude="object").columns.tolist()
categorical_columns = X.select_dtypes(include="object").columns.tolist()

# Preprocessing: OneHotEncoder for categorical columns and
StandardScaler for numeric columns
numeric_transformer = StandardScaler()
oh_transformer = OneHotEncoder()
preprocessor = ColumnTransformer(
    transformers=[
        ("OneHotEncoder", oh_transformer, categorical_columns),
        ("StandardScaler", numeric_transformer, numeric_columns),
    ])

# Apply preprocessing to features
X = preprocessor.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and fit the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Display the shapes of training and testing sets
print("\nTRAIN-TEST SPLIT: \n")

print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)

print('\n' + '-' * 67 + '\n')
```



```
print("PREDICTIONS: \n")

print("y_train_pred: \n\n", y_train_pred)
print()
print("y_test_pred: \n\n", y_test_pred)

print('\n' + '-' * 67 + '\n')

mse = mean_squared_error(y_test, y_test_pred)
r2 = r2_score(y_test, y_test_pred)

print("EVALUATIONS: \n")
print("MSE: ", mse)
print("R2: ", r2)
```

This code snippet performs several steps for machine learning modeling and evaluation:

1. **Define Features and Target**:
   - X: Features are all columns except the 'writing score' column.
   - y: The target variable is the 'writing score' column.
2. **Define Numeric and Categorical Columns**:
   - Separate numeric and categorical columns from the features (X) using the select_dtypes method.
3. **Preprocessing**:
   - Numeric features are standardized using StandardScaler.
   - Categorical features are one-hot encoded using OneHotEncoder.
   - Both preprocessing steps are combined using ColumnTransformer.
4. **Apply Preprocessing**:

- Preprocessing is applied to the features (X) using the fit_transform method of the ColumnTransformer.

5. **Split Data into Training and Testing Sets**:
    - The dataset is split into training and testing sets using train_test_split, with a test size of 20% and a random state 42.

6. **Initialize and Fit the Linear Regression Model**:
    - A Linear Regression model is initialized and fitted to the training data.

7. **Make Predictions**:
    - The trained model makes Predictions on both the training and testing sets.

8. **Display Shapes of Training and Testing Sets**:
    - The shapes of the training and testing sets are displayed to confirm the splitting process.

9. **Display Predictions**:
    - Predicted values for both the training and testing sets are displayed.

10. **Evaluation Metrics**:
    - Mean Squared Error (MSE) and R-squared (R^2) scores are calculated to evaluate the model's performance on the testing set.

These steps collectively enable the training, evaluation, and interpretation of a Linear Regression model for predicting writing scores based on other features in the dataset. The displayed evaluation metrics provide insights into the model's performance and its ability to generalize to unseen data.

IV. Key Findings

In predictive modeling, MSE (Mean Squared Error) and R-squared (R2) are commonly used metrics to evaluate a model's performance.

- **Mean Squared Error (MSE):** MSE measures the average squared difference between the actual values (observed) and the predicted values (estimated) by the model. It quantifies the overall quality of model's predictions. Lower values of MSE indicate better predictive performance, as they reflect smaller errors between predicted and actual values.

- **R-squared (R2):** R2 represents the proportion of variance in the dependent variable (target) that is explained by the independent variables (features) in the model. It ranges from 0 to 1, where R2 = 0 indicates that the model does not describe any variability in the target variable. R2 = 1 suggests that the model perfectly explains the variability in the target variable. Higher values of R2 indicate a better fit of the model to the data, suggesting that the independent variables effectively explain variation in the dependent variable.

Math Score Evaluation

	MSE	R2
Linear Regression	29.095	0.880
Ridge Regression	29.056	0.881
Lasso Regression	28.821	0.882
Random Forest Regressor	36.212	0.851
AdaBoost Regressor	44.868	0.816

Reading Score Evaluation

	MSE	R2

Linear Regression	18.378	0.919
Ridge Regression	18.514	0.918
Lasso Regression	18.723	0.917
Random Forest Regressor	20.512	0.909
AdaBoost Regressor	24.236	0.893

Writing Score Evaluation

	MSE	R2
Linear Regression	14.945	0.938
Ridge Regression	14.909	0.938
Lasso Regression	15.232	0.937
Random Forest Regressor	20.383	0.915
AdaBoost Regressor	27.422	0.886

Findings:

- Linear Models (Linear Regression, Ridge Regression, Lasso Regression) consistently performed better than Non-Linear Models (Random Forest Regressor, AdaBoost Regressor) across all three score evaluations (math, reading, and writing).

- Among the linear models, Lasso Regression slightly outperformed the others for math scores, Linear Regression was best for reading scores, and Ridge Regression was best for writing scores.

- Random Forest Regressor and AdaBoost Regressor consistently showed higher prediction errors (higher MSE) and less explained variance (lower R2), making them less effective for these datasets.

V. Advanced Analysis

Feature Importance Analysis

Feature importance analysis using a Random Forest model provides insights into which features most influence predicting the target variables. The Random Forest algorithm evaluates the importance of each feature by measuring the decrease in prediction error when the feature is included in the model. The more a feature decreases the error, the more important it is.

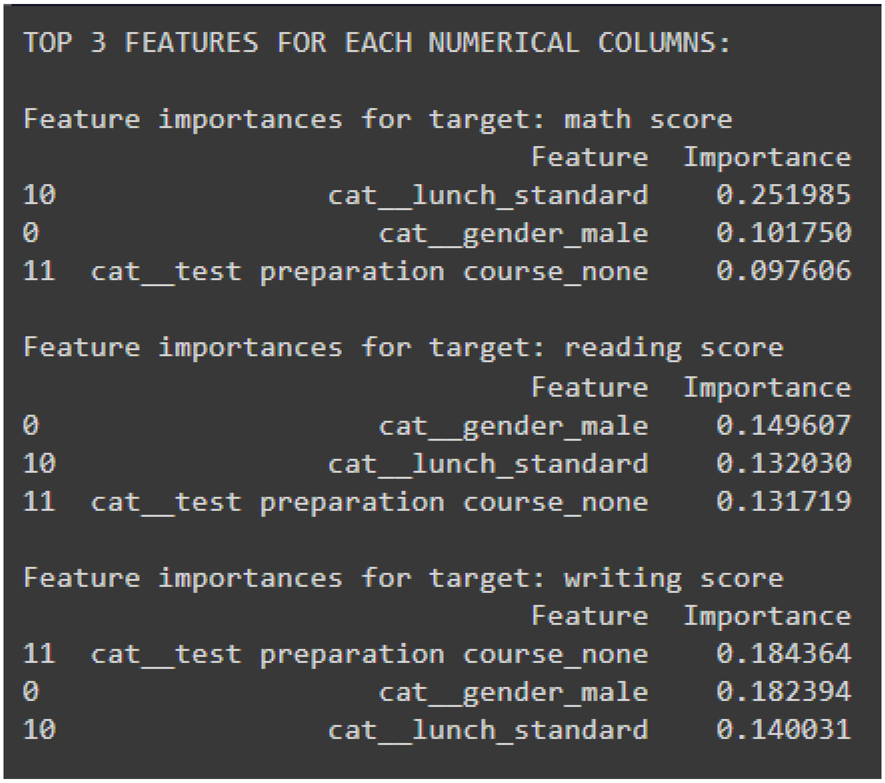
The feature importance analysis highlights the most influential factors affecting student performance in math, reading, and writing scores. This analysis may contribute to understanding demographic and socioeconomic factors and preparation and support influences such as:

- **Gender**: If gender is identified as an essential feature, it could highlight gender disparities in educational outcomes.

- **Race/Ethnicity**: Understanding how different racial or ethnic backgrounds influence performance is essential, possibly indicating systemic issues or the need for tailored educational strategies.

- **Lunch**: If lunch status is significant, it may underline the effects of socioeconomic status on academic performance, advocating for policies that address nutritional support in schools.

- **Test Preparation Course**: Highlights the importance of preparation courses, suggesting that access to additional resources and structured preparation can improve student outcomes.



```
TOP 3 FEATURES FOR EACH NUMERICAL COLUMNS:

Feature importances for target: math score
                              Feature  Importance
10                cat__lunch_standard    0.251985
0                    cat__gender_male    0.101750
11  cat__test preparation course_none    0.097606

Feature importances for target: reading score
                              Feature  Importance
0                    cat__gender_male    0.149607
10                cat__lunch_standard    0.132030
11  cat__test preparation course_none    0.131719

Feature importances for target: writing score
                              Feature  Importance
11  cat__test preparation course_none    0.184364
0                    cat__gender_male    0.182394
10                cat__lunch_standard    0.140031
```

The image above shows the top three features for each numerical column (math, reading, and writing scores).

VI. Machine Learning Implementation

```
# Initialize and fit the Linear Regression model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)

# Initialize and fit the Ridge model
ridge_model = Ridge(alpha=1.0)  # You can adjust the alpha value
ridge_model.fit(X_train, y_train)

# Initialize and fit the Lasso model
lasso_model = Lasso(alpha=0.1)  # You can adjust the alpha value
```

```
lasso_model.fit(X_train, y_train)

# Initialize and fit the Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Initialize and fit the AdaBoost Regressor
adaboost_model = AdaBoostRegressor(n_estimators=50, learning_rate=0.1,
random_state=42)
adaboost_model.fit(X_train, y_train)
```

Data Preparation

- ***Training Data***: The dataset is split into X_train (features) and y_train (target) for model training.
- ***Test Data***: X_test (features) is prepared to evaluate the models' performance.

Model Initialization and Fitting

Different regression models are initialized and trained on the training data. Each model has its strengths and handles data differently:

- ***Linear Regression***: A straightforward linear model is initialized and fitted to the training data.

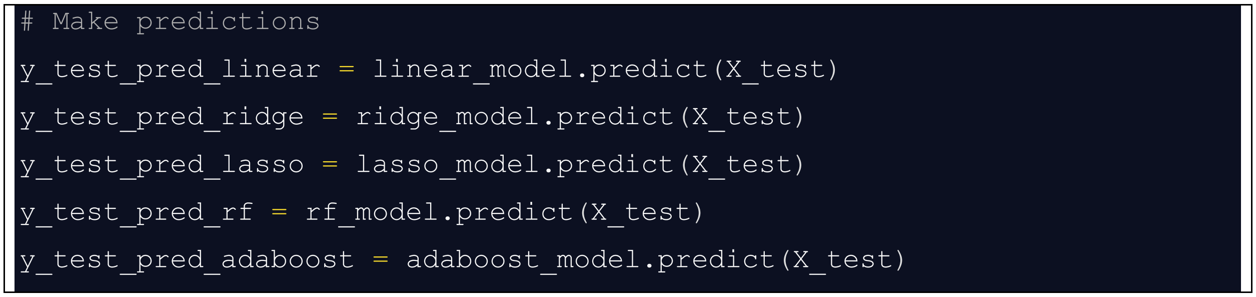
- ***Ridge Regression***: A linear model with L2 regularization to prevent overfitting is initialized and fitted. The alpha parameter controls the regularization strength.

- ***Lasso Regression***: A linear model with L1 regularization to encourage sparsity in the coefficients is initialized and fitted. The alpha parameter controls the regularization strength.

-

- ***Random Forest Regressor***. An ensemble model that builds multiple decision trees and averages their predictions. It is initialized with a specified number of trees and fitted.

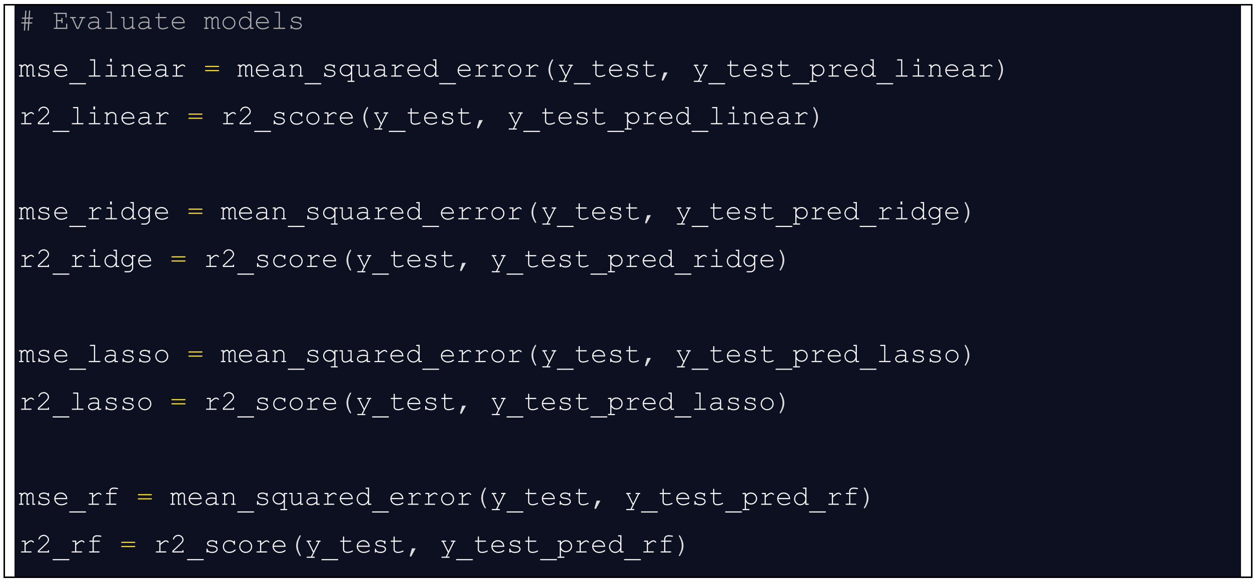
- ***AdaBoost Regressor***. An ensemble model that combines weak learners to create a strong learner through iterative training. It is initialized with a specified number of estimators and learning rate and then fitted.



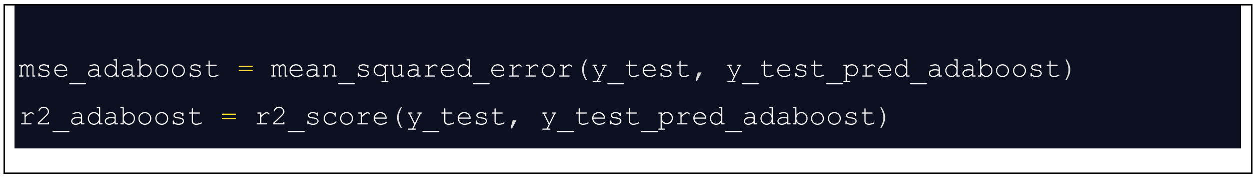
```
# Make predictions
y_test_pred_linear = linear_model.predict(X_test)
y_test_pred_ridge = ridge_model.predict(X_test)
y_test_pred_lasso = lasso_model.predict(X_test)
y_test_pred_rf = rf_model.predict(X_test)
y_test_pred_adaboost = adaboost_model.predict(X_test)
```

Making Predictions

Each trained model predicts the target variable(s) on the test dataset (X_test). The predicted values are stored for later evaluation.



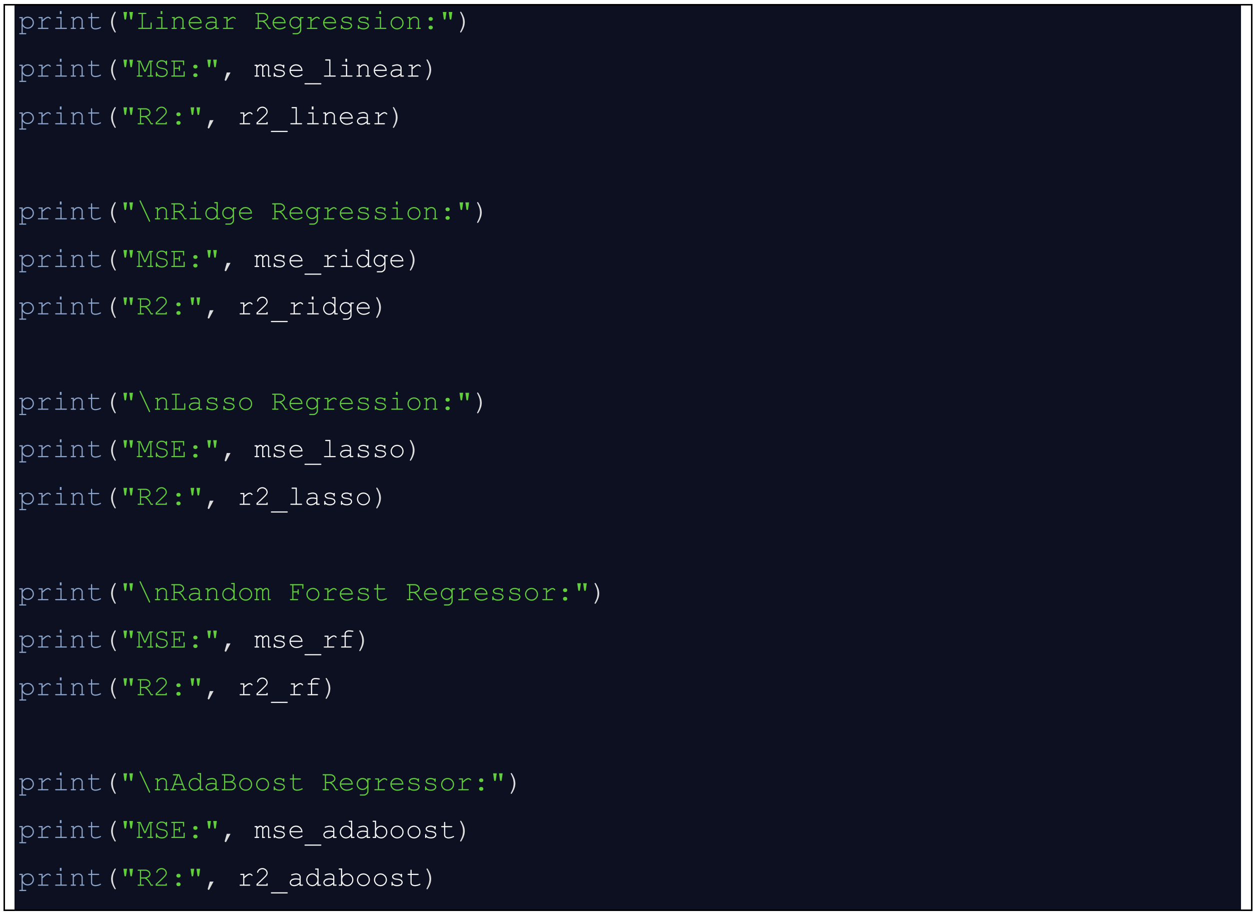
```
# Evaluate models
mse_linear = mean_squared_error(y_test, y_test_pred_linear)
r2_linear = r2_score(y_test, y_test_pred_linear)

mse_ridge = mean_squared_error(y_test, y_test_pred_ridge)
r2_ridge = r2_score(y_test, y_test_pred_ridge)

mse_lasso = mean_squared_error(y_test, y_test_pred_lasso)
r2_lasso = r2_score(y_test, y_test_pred_lasso)

mse_rf = mean_squared_error(y_test, y_test_pred_rf)
r2_rf = r2_score(y_test, y_test_pred_rf)
```



```
mse_adaboost = mean_squared_error(y_test, y_test_pred_adaboost)
r2_adaboost = r2_score(y_test, y_test_pred_adaboost)
```

Model Evaluation

MSE and R2 metrics provide a quantitative measure of model accuracy and goodness of fit. Lower MSE and higher R2 indicate better model performance.

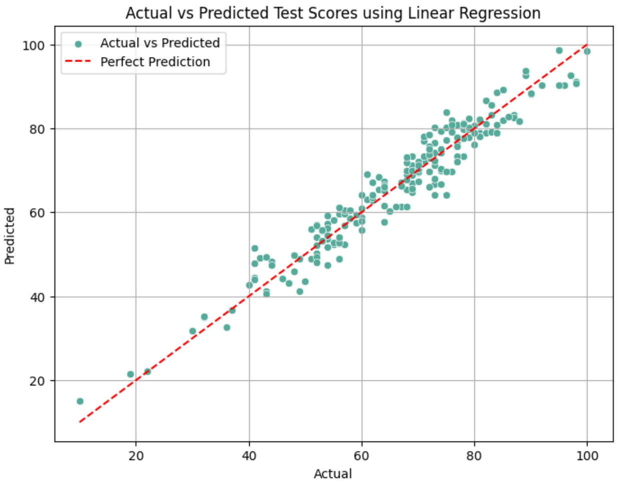


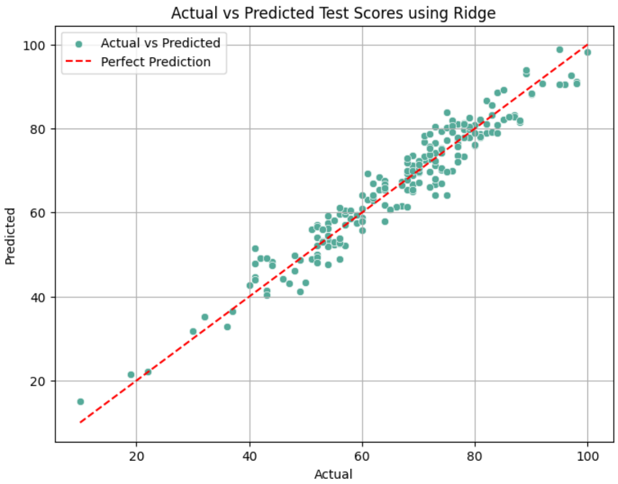
```
print("Linear Regression:")
print("MSE:", mse_linear)
print("R2:", r2_linear)

print("\nRidge Regression:")
print("MSE:", mse_ridge)
print("R2:", r2_ridge)

print("\nLasso Regression:")
print("MSE:", mse_lasso)
print("R2:", r2_lasso)

print("\nRandom Forest Regressor:")
print("MSE:", mse_rf)
print("R2:", r2_rf)

print("\nAdaBoost Regressor:")
print("MSE:", mse_adaboost)
print("R2:", r2_adaboost)
```

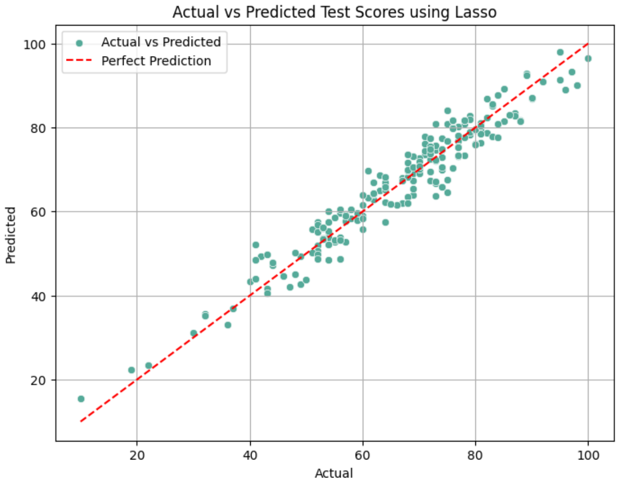
Printing Evaluation Results

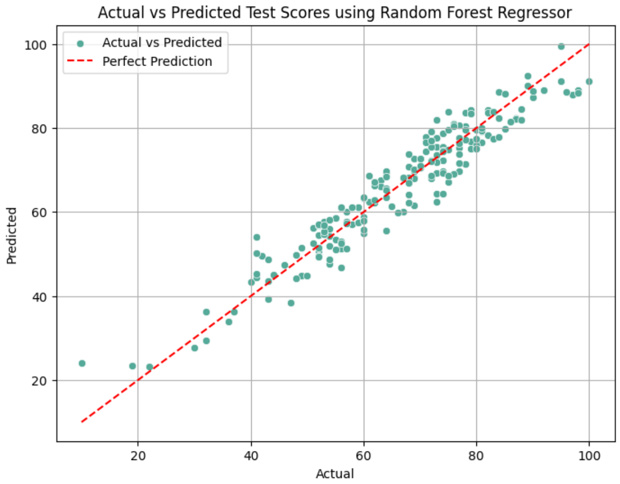
Each model's evaluation metrics (MSE and R2) are printed to compare their performance. This helps identify which model best predicts the target variable.

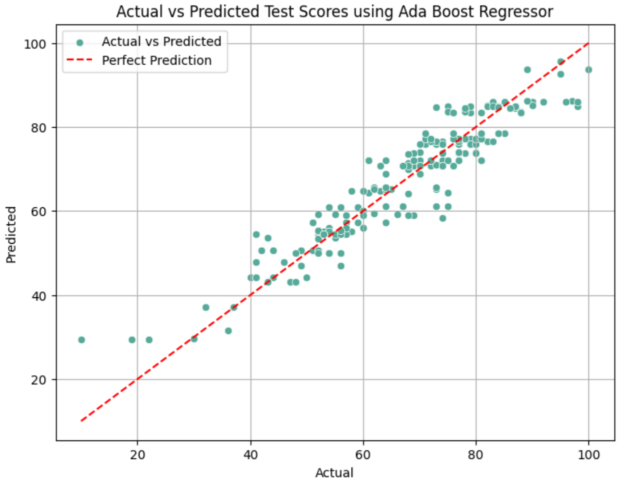
Actual Plot vs. Predicted Values for each Model


Actual vs Predicted Test Scores using Linear Regression


Actual vs Predicted Test Scores using Ridge


Actual vs Predicted Test Scores using Lasso


Actual vs Predicted Test Scores using Random Forest Regressor



VII. Visual Insights

Various plots such as histograms, bar charts, pie charts, and violin plots are used to visualize the student performance in examinations by correlating their gender, parental level of education, lunch they receive, race/ethnicity, and test preparation courses to their math, reading, and writing examination scores. Here's how various types of plots are employed:

- **Histogram:** A histogram is a graph used to represent the frequency distribution of a few data points of one variable by grouping them into logical ranges or bins. In this project, it illustrates the student's scores in math, reading, and writing.

- **Bar chart:** This type of chart usually presents categorical, discrete, or continuous variables grouped in class intervals. It is used in this project to illustrate the distribution of students of various genders, races/ethnicities, their parental levels of education, lunches, and test preparation courses.

- **Pie Chart:** Shows a parts-to-whole relationship for categorical data, including ordinal and nominal data. In this project, it is used to show the overall percentage of the student distribution across different independent variables.

- **Violin Plot:** Allows to visualize the distribution of a numeric variable for one or several groups. In this project, it is used to illustrate the correlation of the dependent variable (three subject scores) to the independent variables (genders, race/ethnicity, parental level of education, lunches, and test preparation courses).

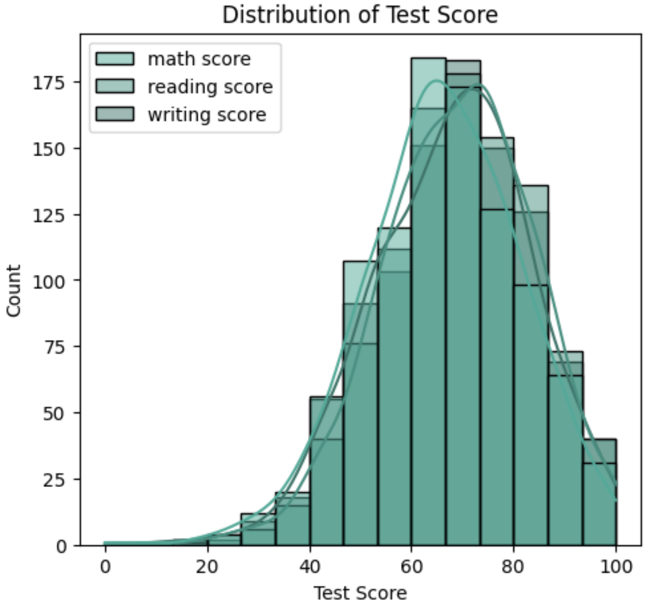


Figure 1.0. The histogram with density plots shows that student performance in math, reading, and writing is generally average, with scores clustering between 60 and 80. Scores are strongly correlated with minimal skewness and consistent variability, indicating similar overall performance.

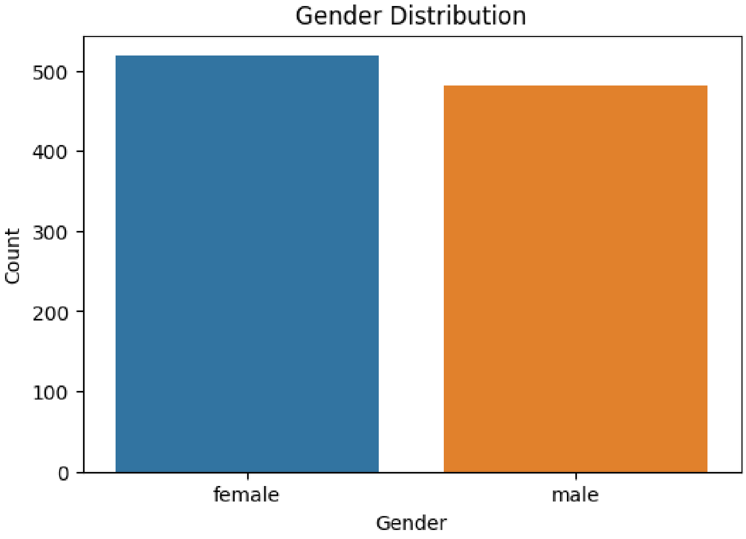


Figure 2.0. The bar chart shows a near-equal gender distribution of students, with a slight majority of female students, with a count of just over 500, and a slightly higher number of male students, indicating a balanced representation of genders in the dataset.

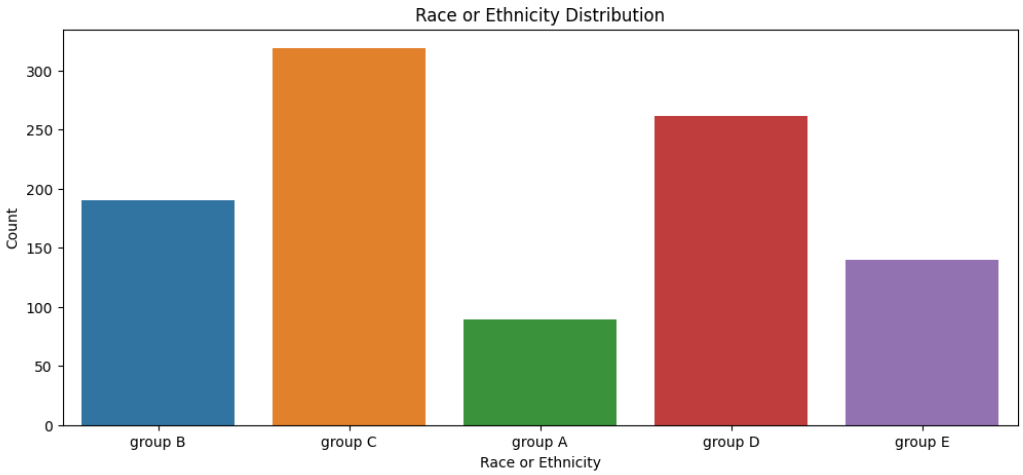


Figure 3.0. The bar chart displays the distribution of students across five race or ethnicity groups. Group C has the highest count, with over 300 students, followed by Group D, with slightly under 300 students. Group B has 200 students, Group E has over 100 students, and Group A has the lowest count, with under 100 students.

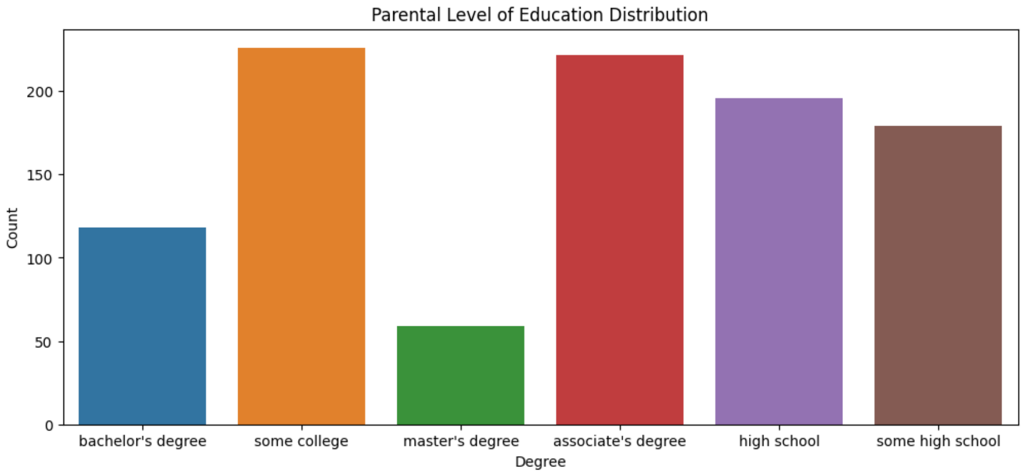


Figure 4.0. The bar graph shows the distribution of parental education levels among students. A bachelor's degree is the most common attainment, followed by an associate degree with a minimal gap. Master's degrees are the least common attainment.

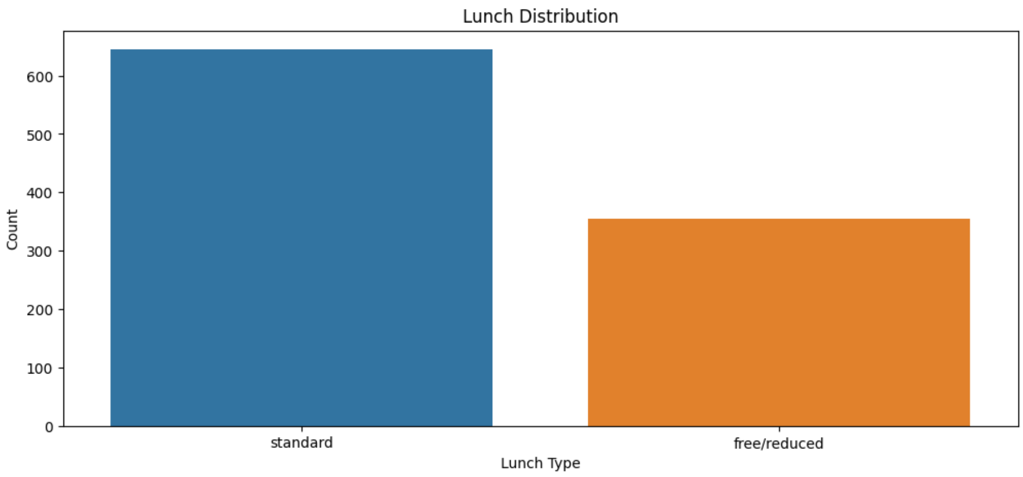


Figure 5.0. The bar graph shows the distribution of lunches by type, categorized as standard and free/reduced. Standard lunches are distributed more than free/reduced lunches, with approximately twice as many.

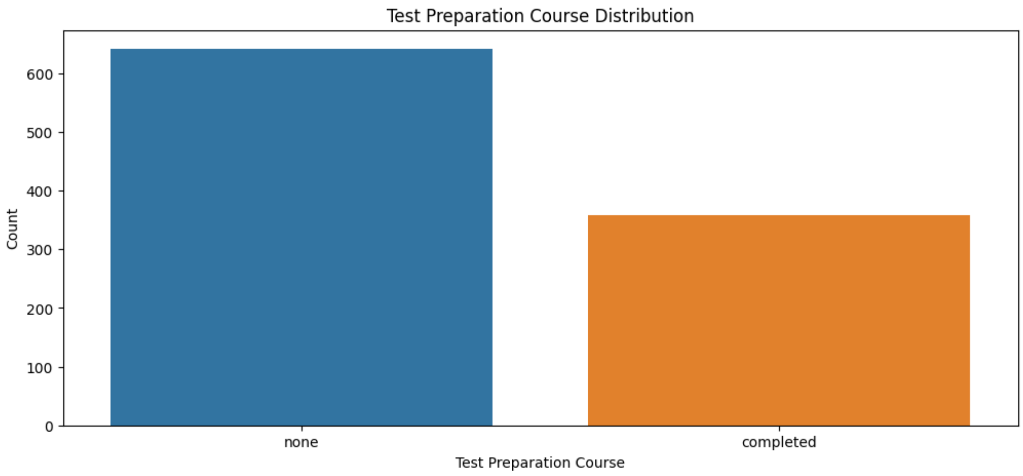


Figure 6.0. The bar graph shows the distribution of test preparation courses by completion status, with the orange bar representing students who have completed the course and the blue bar representing those who haven't. The taller blue bar indicates that more students still need to complete the course than those who have.

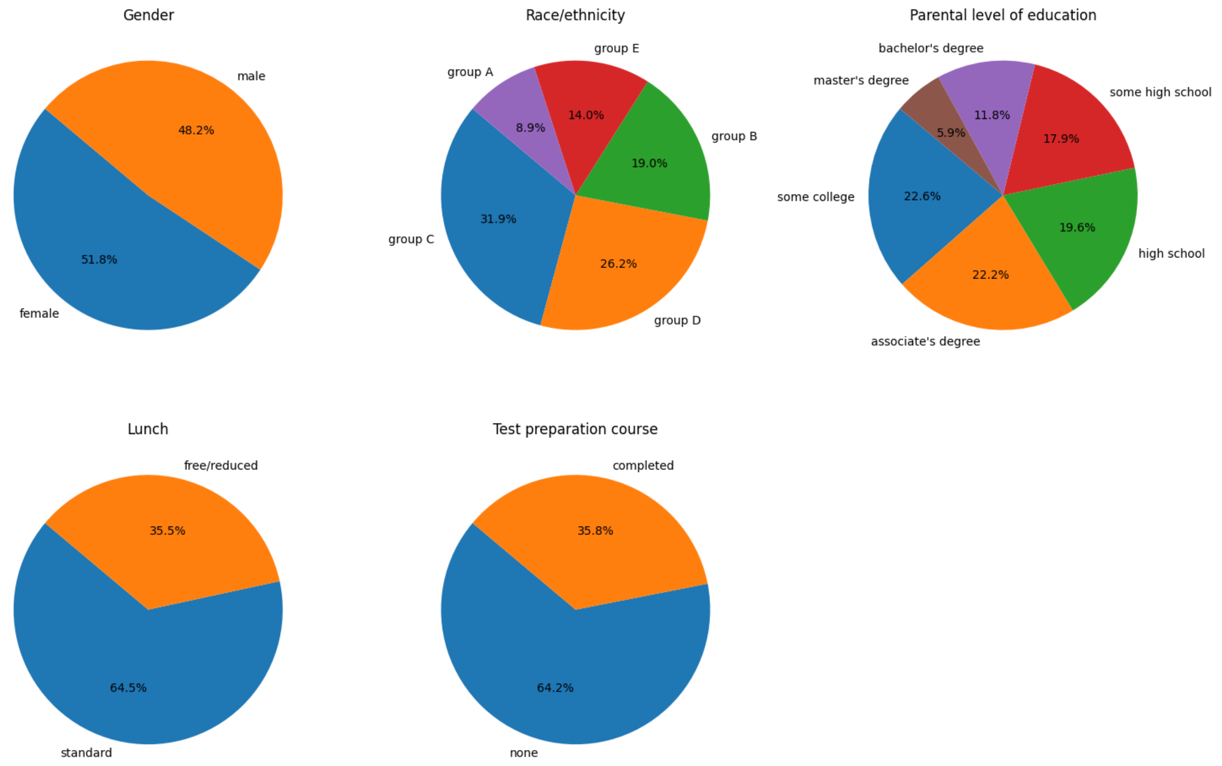


Figure 7.0. The figure shows four pie charts. The first chart depicts nearly equal proportions of males and females. The second chart categorizes students by race or ethnicity into five distinct groups. The third chart illustrates parents' education levels, including high school diplomas, some college, and bachelor's degrees. The fourth chart combines data on lunch types and test preparation course completion.

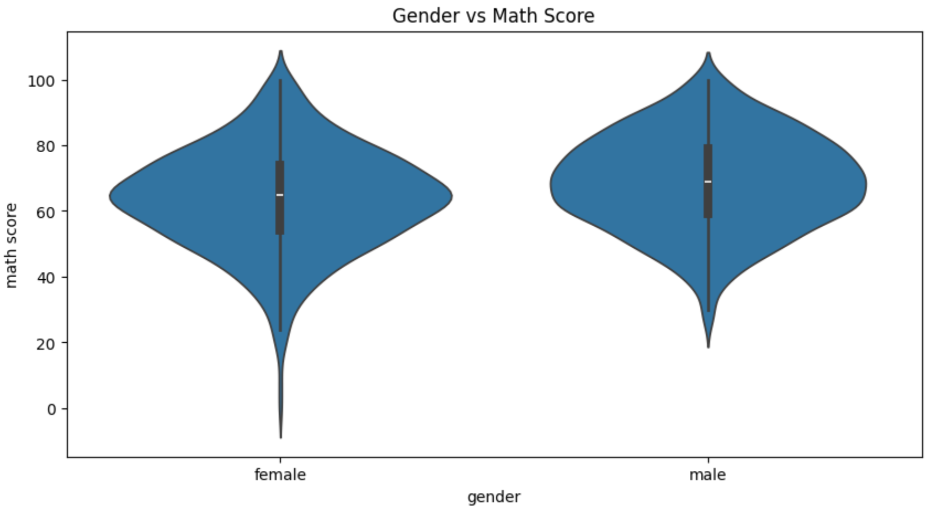


Figure 8.0. The violin plot shows the distribution of math scores across two genders. The width represents the concentration of data, while thin tails extend to extreme scores. Females appear to have higher math scores than males.

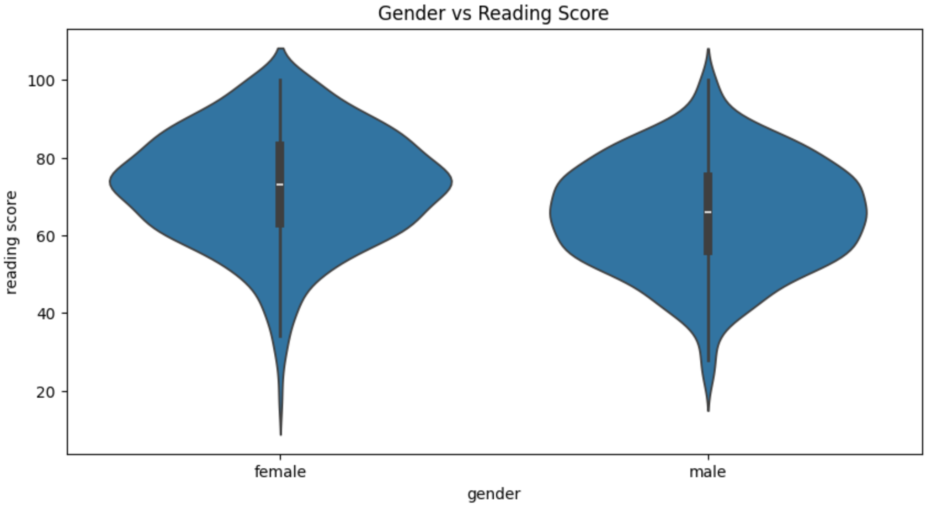


Figure 9.0. The violin plot indicates that females have slightly higher reading scores than males. The center line for females is higher than for males, indicating a potential difference in median scores.

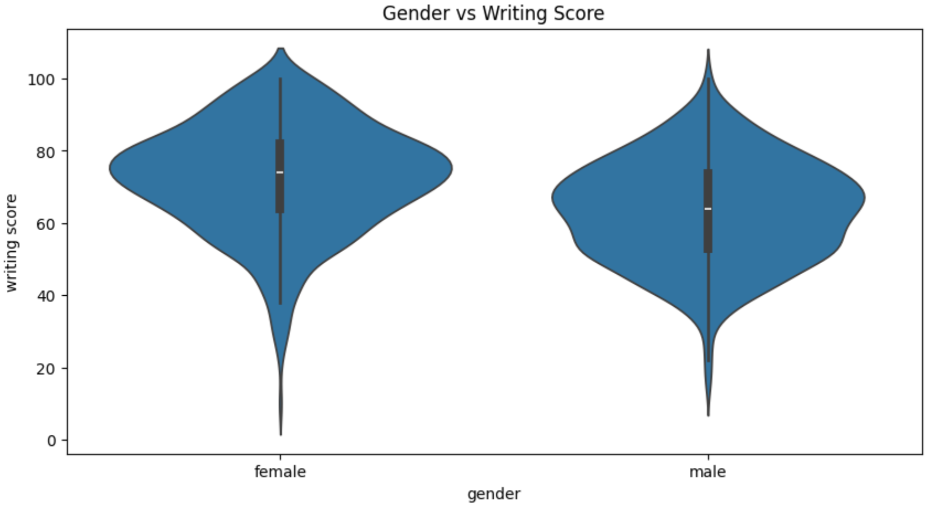


Figure 10.0. The violin plot indicates potential gender differences in writing scores. Females have higher median scores than males, and wider violins indicate a more extensive spread of scores among females. The center line for females is higher than for males, suggesting a potential difference in writing distribution.

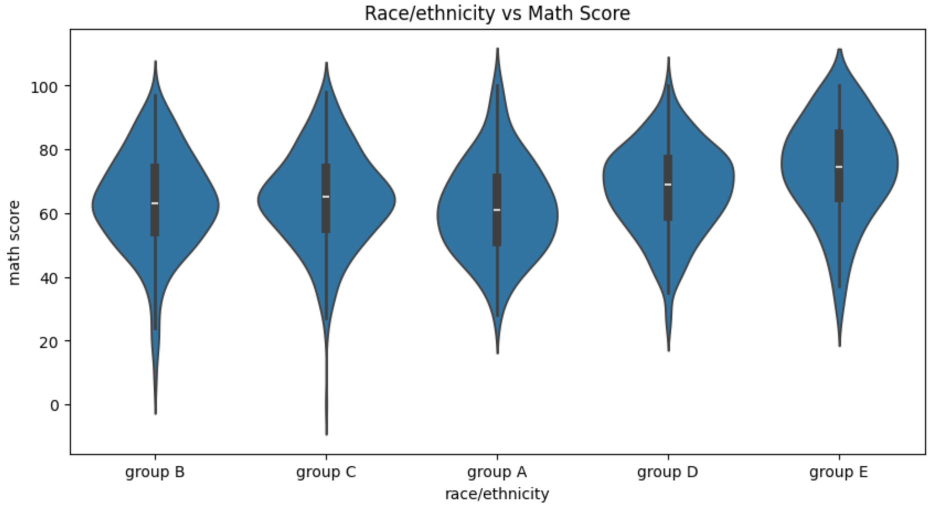


Figure 11.0. The violin plot depicts the distribution of math scores across five different racial/ethnic groups. Groups A and D show a higher density of higher scores, suggesting better performance in math compared to the other groups. Group E shows a higher density of lower scores, suggesting relatively poorer performance in math. Group A stands out with higher overall performance in math scores, while Group E shows the lowest performance and most minor variability in math.

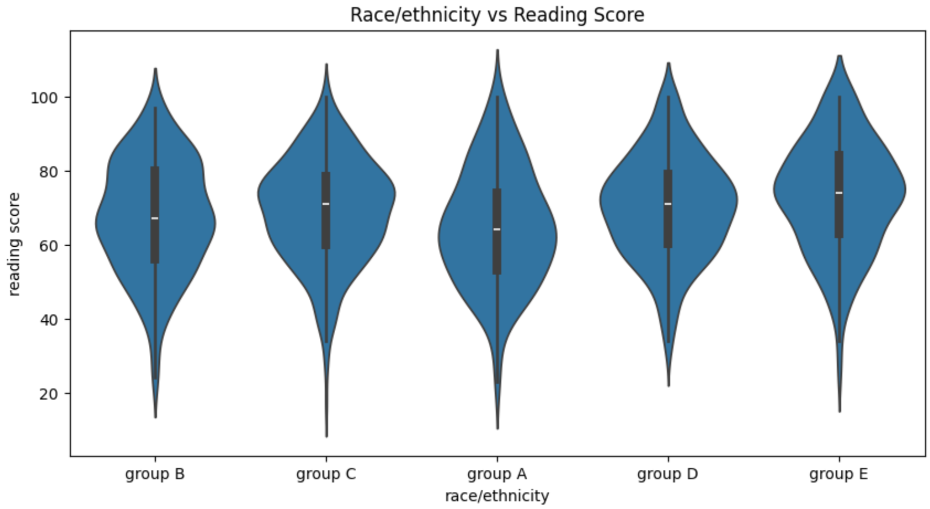


Figure 12.0. The violin plot shows the distribution of reading scores across five different racial/ethnic groups. All groups have a similar overall spread of reading scores, ranging from approximately 20 to 100. Groups C and E have the highest median reading scores, while Group A has the lowest. Groups B and D have an even distribution, while Groups C and E have a higher concentration of scores in the upper range.

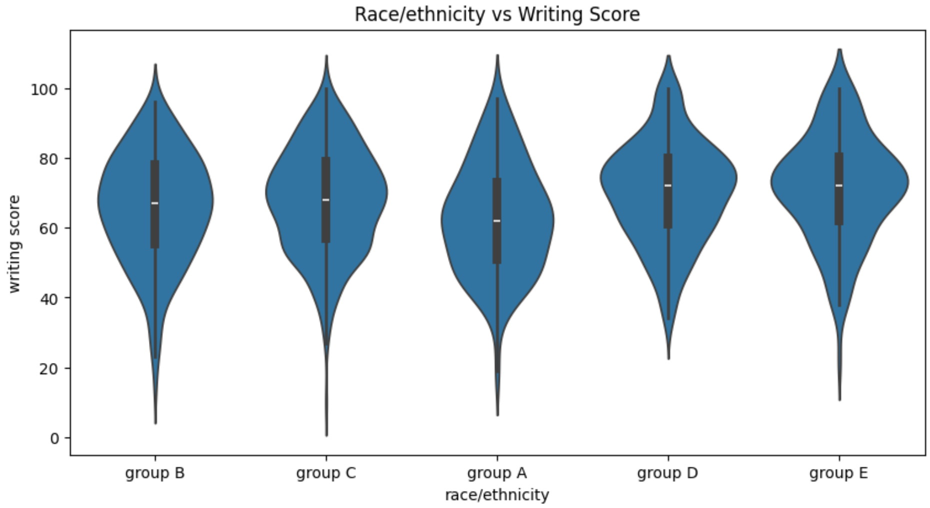


Figure 13.0. The violin plot shows the distribution of writing scores across five different racial/ethnic groups. The plot shows a similar range of scores, with Groups C and E having the highest median scores. Groups B and D have an even distribution, while Groups C and E have a

higher concentration in the upper range. Group A has a more uniform distribution with less concentration, indicating a more varied performance.

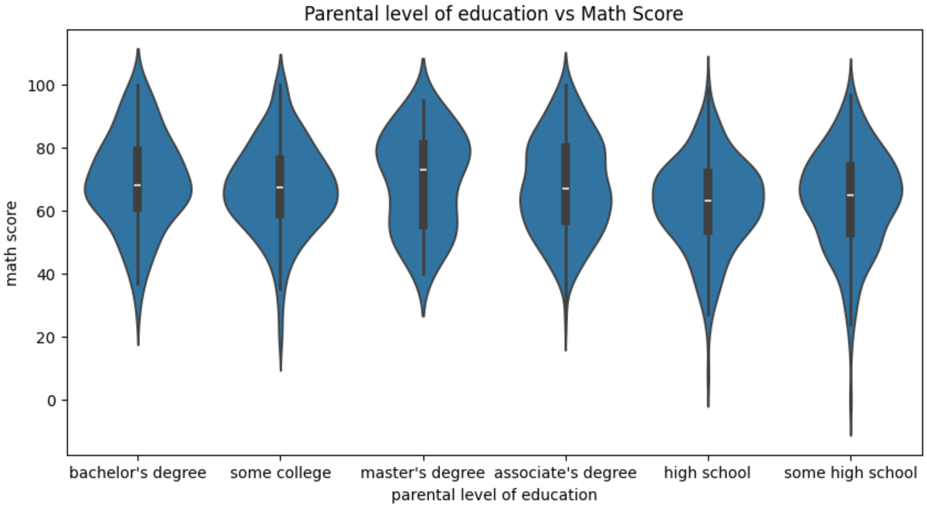


Figure 14.0. The plot shows a positive correlation between parental education and math scores. As parental education levels increase, the distribution of math scores widens, indicating a more extensive spread among students with more educated parents.

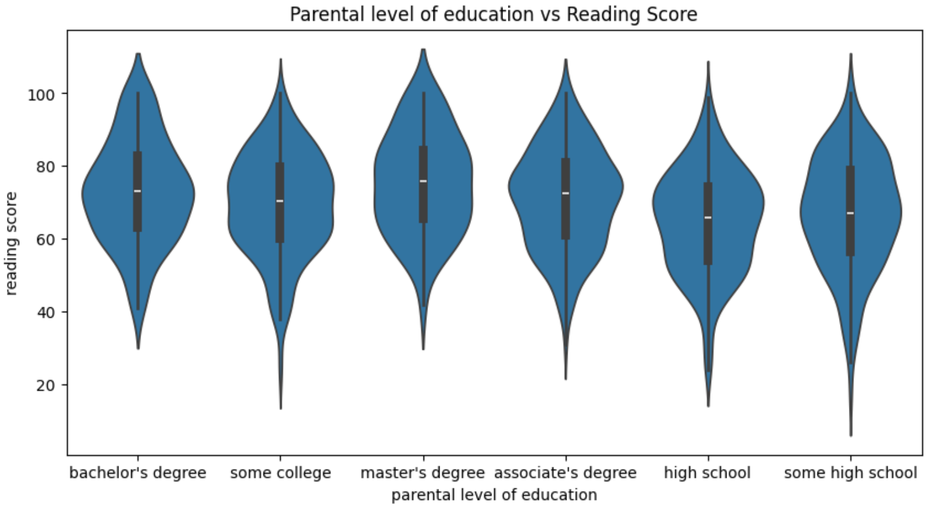


Figure 15.0. The violin plot suggests a correlation between parental education and math scores. As parental education increases, the center lines of the plots increase on the y-axis, indicating a

trend of increasing median scores. Wider violins at higher education levels also indicate a more extensive spread of math scores among students in those groups.

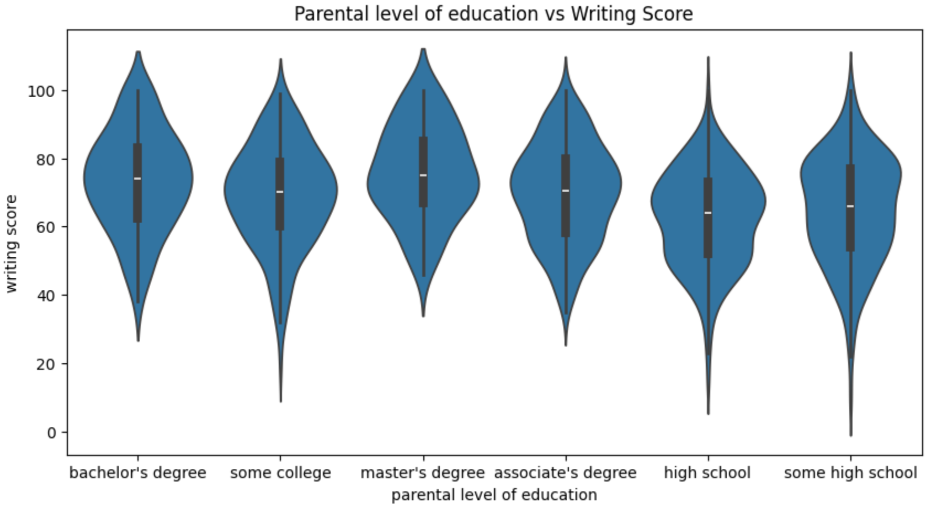


Figure 16.0. The plot shows a positive correlation between parental education and writing scores. As parental education levels increase, the median writing score may increase with the parents' education. The distribution of writing scores widens, indicating a more extensive spread among students with more educated parents.

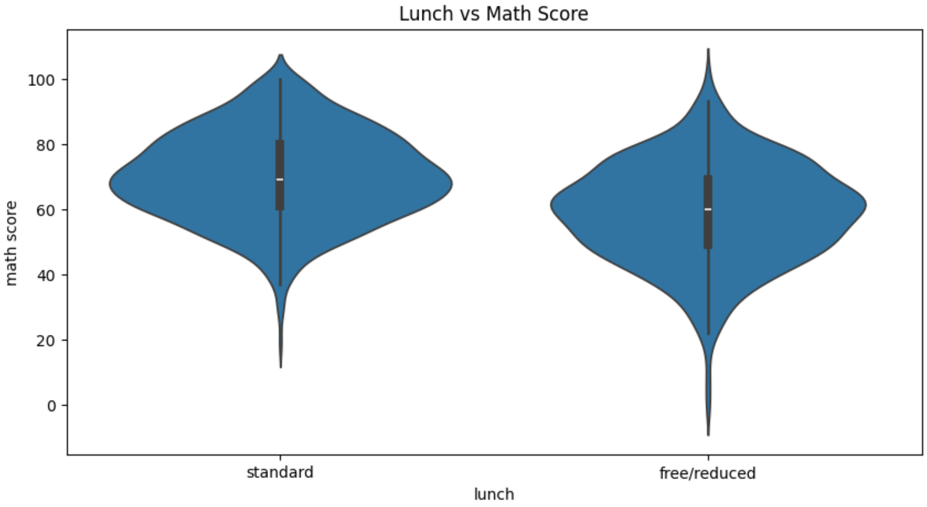


Figure 17.0. The violin plot reveals that students with standard lunch generally have higher math scores than those with free/reduced lunch. The distribution of scores is broader and more

concentrated around higher scores, particularly between 60 and 80. The distribution for free/reduced lunch is slightly narrower and more focused on lower scores, particularly between 40 and 60.

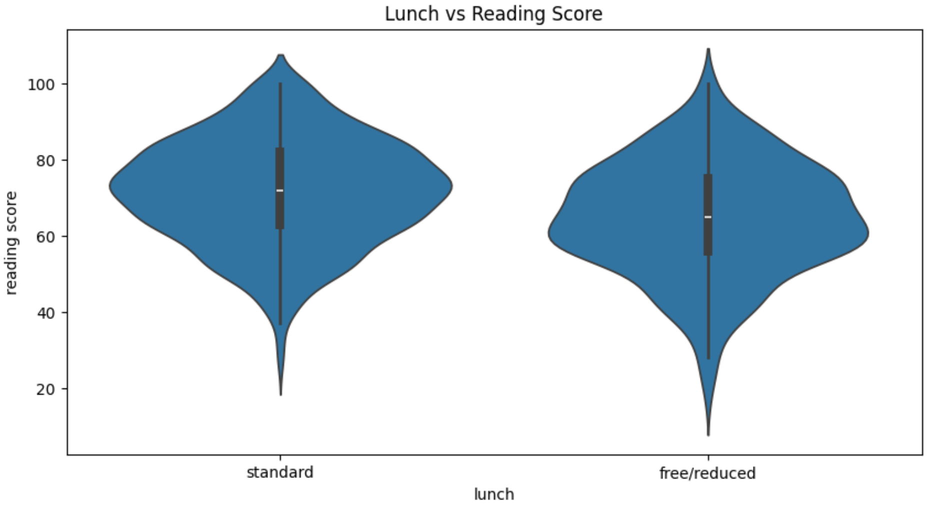


Figure 18.0. The violin plot shows the distribution of reading scores for two groups of students based on their lunch status. The "standard" lunch group appears broader and more symmetrical, while the "free/reduced" lunch group has a slightly narrower and less symmetrical distribution. The median reading score for the "standard" lunch group is higher than that for the "free/reduced" lunch group. This suggests that socioeconomic factors related to lunch status impact students' academic performance.

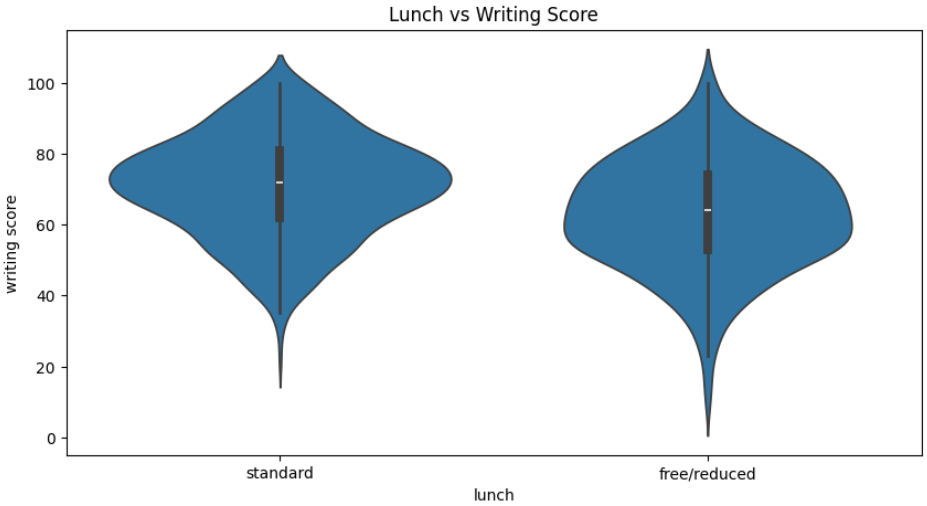


Figure 19.0. The violin plot reveals that students with standard lunch tend to have better writing scores on average. The standard lunch group's median writing score is higher, suggesting better performance. The overall spread of scores is more comprehensive for the standard lunch group. The density of scores around the higher range of writing scores suggests more students achieve higher scores. The lower median and more centralized distribution of scores for the free/reduced lunch group may negatively affect writing performance.

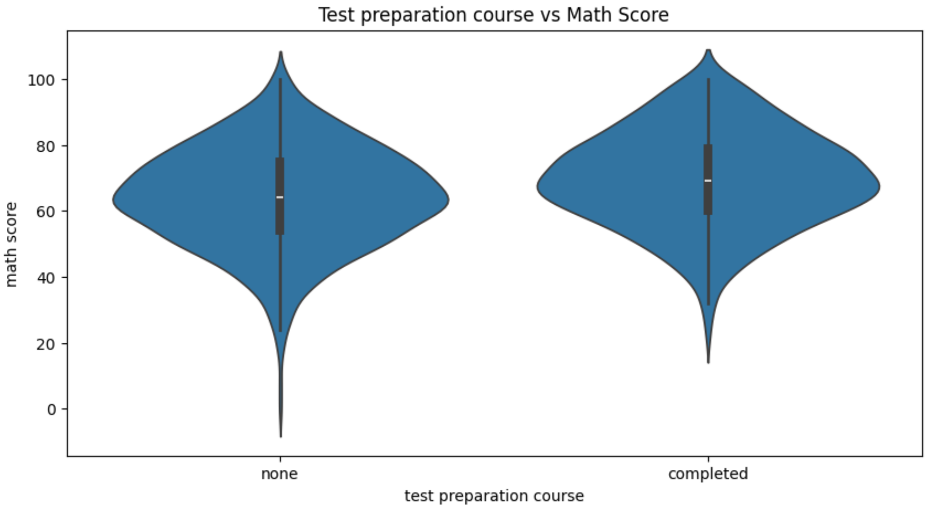


Figure 20.0. The plot compares test preparation course completion and math score. This shows that students who completed a test preparation course may have higher math scores. The center line and broader part of the plot are higher for those who completed the course.

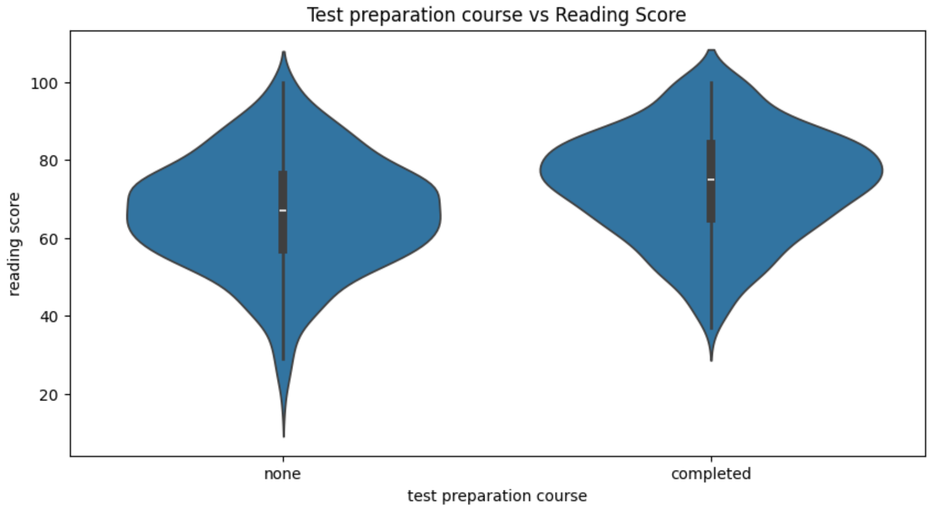


Figure 21.0. The plot shows that students who completed a test preparation course had higher reading scores and a more concentrated distribution. This suggests that the course effectively improves reading performance and reduces variability. However, students who did not take the course had a wider spread and higher density, suggesting less consistent performance.

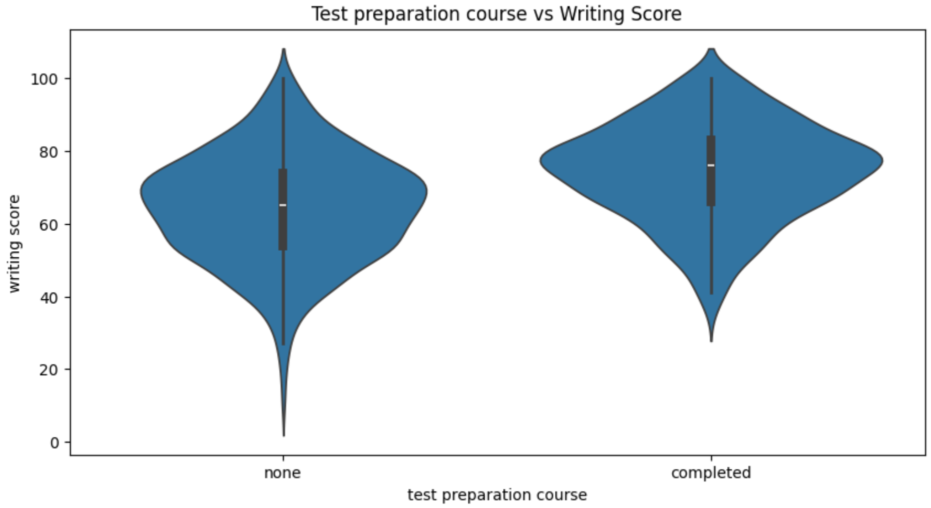


Figure 22.0. The plot compares test preparation course completion and writing score. This shows that students who completed a test preparation course may have higher writing scores. The center line and broader part of the plot are higher for those who completed the course.

VIII. Conclusion

The analysis of the student performance dataset produced several important discoveries. According to descriptive statistics, the dataset has 1000 records, with comprehensive data on various performance and demographic characteristics. Significant variations in test preparation, lunch type, parental education, gender, and race/ethnicity were found using inferential statistics. Particularly, performance varied across racial and cultural groups, although males and females showed variations in math and writing scores. Higher parental education levels have been linked to improved student performance; test-preparation course completion and standard lunch recipients typically yielded higher scores. With high R2 values and low MSE, predictive modeling employing ridge regression, lasso regression, random forest regressor, and AdaBoost regressor revealed that ridge regression and linear regression were the most successful models. The most significant predictors of the three subject examination scores were found to be gender, lunch, and test preparation course completion by feature importance analysis. Score distributions and demographic characteristics were made more evident by visual aids such as violin plots, bar charts, and histograms.

The analysis concludes by highlighting the critical influence that socioeconomic and demographic factors have on student achievement. Several focused recommendations are made to improve educational outcomes. First, creating educational programs tailored to the unique learning preferences and needs of male and female students helps promote a more welcoming and productive learning environment. Second, by making the material more interesting and relevant for all students, the use of culturally responsive teaching practices can aid in closing the achievement gap between various racial and ethnic groups. Fostering parental involvement via workshops and resources can enable parents to support their kids' education better, strengthening the bond between the home and the school.

Furthermore, it is critical to guarantee that all kids have access to sustaining meals because a healthy diet is associated with improved focus and general academic success. Encouraging and offering easily accessible test-taking programs can help level the playing field by equipping all students with the skills they need to achieve. By putting in place mechanisms for ongoing observation and support of student performance, problems can be quickly identified and resolved, guaranteeing that students have the support they require to succeed. Policymakers should consider these findings when developing inclusive and equitable educational programs. By addressing these areas, educational institutions can help all students reach their full academic potential and create a more equal and encouraging learning environment.

Appendix

Datasets

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score
female	group B	bachelor's degree	standard	none	72
female	group C	some college	standard	completed	69
female	group B	master's degree	standard	none	90
male	group A	associate's degree	free/reduced	none	47
male	group C	some college	standard	none	76
female	group B	associate's degree	standard	none	71
female	group B	some college	standard	completed	88
male	group B	some college	free/reduced	none	40
male	group D	high school	free/reduced	completed	64
female	group B	high school	free/reduced	none	38
male	group C	associate's degree	standard	none	58
male	group D	associate's degree	standard	none	40
female	group B	high school	standard	none	65
male	group A	some college	standard	completed	78
female	group A	master's degree	standard	none	50
female	group C	some high school	standard	none	69
male	group C	high school	standard	none	88
female	group B	some high school	free/reduced	none	18
male	group C	master's degree	free/reduced	completed	46
female	group C	associate's degree	free/reduced	none	54
male	group D	high school	standard	none	66
female	group B	some college	free/reduced	completed	65
male	group D	some college	standard	none	44
female	group C	some high school	standard	none	69
male	group D	bachelor's degree	free/reduced	completed	74
male	group A	master's degree	free/reduced	none	73
male	group B	some college	standard	none	69
female	group C	bachelor's degree	standard	none	67
male	group C	high school	standard	none	70
female	group D	master's degree	standard	none	62
female	group D	some college	standard	none	69
female	group B	some college	standard	none	63
female	group E	master's degree	free/reduced	none	56
male	group D	some college	standard	none	40
male	group E	some college	standard	none	97
male	group E	associate's degree	standard	completed	81
female	group D	associate's degree	standard	none	74
female	group D	some high school	free/reduced	none	50
female	group D	associate's degree	free/reduced	completed	75
male	group B	associate's degree	free/reduced	none	57
male	group C	associate's degree	free/reduced	none	55
female	group C	associate's degree	standard	none	58
female	group B	associate's degree	standard	none	53