A Conclusive report on Student Performance Analysis

Arshia Dezfooli, Dhruv Mann and Dipak Kushwaha

School of Software Design and Data Science, Seneca College

SEA500NAA: Introduction to Data Mining

Prof. Mufleh Al-Shatnawi

December 01,2024

Abstract

This project explores the key factors influencing student performance in Nepalese schools by analyzing a comprehensive dataset containing 6,607 records across 20 variables. The dataset encompasses a wide range of attributes, including socio-economic factors (such as family income and parental education), academic inputs (hours studied, tutoring sessions, and teacher quality), and behavioral factors (motivation level, peer influence, and physical activity). The target variable, Exam Score, serves as the benchmark for evaluating student outcomes.

The primary objective of this analysis is to provide actionable insights to the Ministry of Education in Nepal, enabling data-driven policy formulation aimed at improving educational outcomes. To achieve this, we employed rigorous data preparation techniques, including handling missing data, encoding categorical variables, and scaling numerical features. Following data preprocessing, machine learning models—specifically Decision Tree and Logistic Regression—were trained and evaluated to predict student performance and uncover critical determinants. The models were chosen based on their interpretability and ability to capture complex interactions between features.

Results from both models revealed that factors such as access to resources, parental involvement, and the presence of learning disabilities significantly impact Exam Scores. The analysis also highlighted disparities in performance based on socio-economic conditions and school proximity, offering insights into areas where targeted interventions could make a meaningful difference.

The findings of this project will be presented to the Ministry of Education to support the development of policies focused on enhancing educational equity and resource allocation. By identifying and addressing key barriers to academic success, this study aims to contribute to long-term improvements in the Nepalese education system, ensuring that students receive the support they need to thrive academically. Through data-driven decision-making, this project underscores the importance of understanding the multi-faceted nature of student performance and leveraging insights to foster a more inclusive and effective learning environment.

Introduction

Education is a key driver of socio-economic development, particularly in emerging economies like Nepal, where disparities in access to quality education remain a persistent challenge. To address these challenges, it is critical to understand the underlying factors that influence academic performance. This project aims to analyze a comprehensive dataset of student performance, collected from schools across Nepal, to uncover the most significant factors contributing to student outcomes. The insights gained will be shared with the Ministry of Education (MOE) to assist in drafting and implementing data-driven educational policies that foster equitable and improved learning environments.

Key Objectives of the Study:

- **Identify Key Factors Influencing Student Performance:**
  By analyzing various socio-economic, behavioral, and academic variables, we aim to pinpoint the factors most strongly correlated with student success in Nepalese schools.
- **Support Evidence-Based Policymaking:**
  The results of the analysis will inform the Ministry of Education's decision-making process, guiding the development of policies that address educational disparities and improve overall academic achievement.
- **Bridge the Gap between Data Science and Education Policy:**
  This project highlights the practical application of machine learning techniques in analyzing educational data to help shape better-informed policies for schools.

Dataset Overview:

The dataset comprises **6,607 records** with **20 columns**, featuring a combination of numerical and categorical variables. Key features include:

- **Numerical Variables:** Hours studied, attendance, tutoring sessions, sleep hours, physical activity, family income, etc.
- **Categorical Variables:** Parental involvement, access to resources, motivation level, teacher quality, gender, and school type.

The target variable is the **Exam Score**, representing student achievement and serving as a proxy for academic success. This diverse set of features reflects the wide array of factors that contribute to educational outcomes.

Data Preparation and Methodology

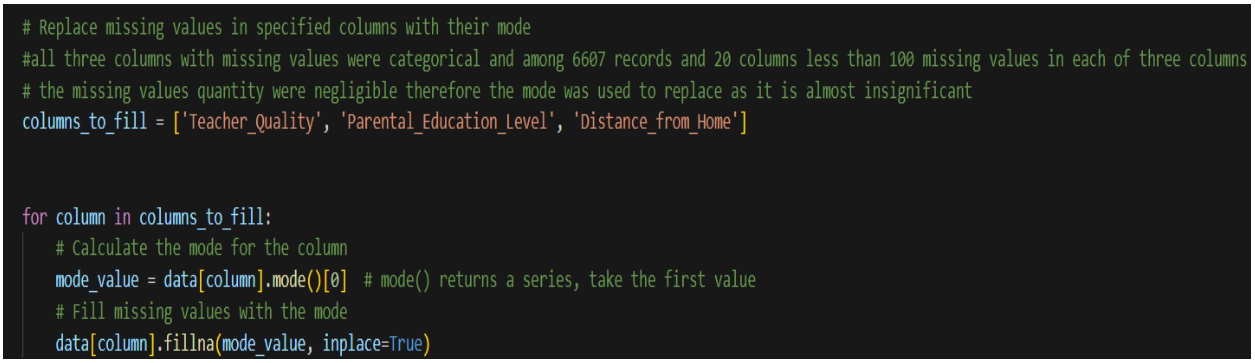
The data preparation phase is critical to ensuring that the dataset is in a suitable format for machine learning algorithms. Since the dataset contains both categorical and numerical features, we undertook several preprocessing steps to ensure that the data is clean, consistent, and ready for model training. Below is a detailed breakdown of the data cleaning and preprocessing steps performed on the dataset.

a. Data Cleaning

Handling Missing Values:
One of the first steps in data cleaning was to address the missing values in the dataset. Missing data can introduce biases and inaccuracies, leading to unreliable models. We carefully examined each column for missing or null values and took the following steps:

- For **columns with missing values** (e.g., *Teacher_Quality*, *Parental_Education_Level*, *Distance_from_Home*), we used:
    - **Mode Imputation** for categorical variables: Missing values were replaced with the most frequent value (mode) of the column. This is particularly useful when the missing values do not significantly affect the distribution of the data.
- After imputation, we rechecked the dataset to ensure that there were no remaining null values, making the dataset fully complete and ready for further analysis.

The rationale behind using mode imputation was all the missing values in were in the categorical column so the using of mean and median goes out of scope. Also, the number of missing values were less than 100 in each missing column, 78 for Teacher Quality, 90 for Parental Education Level and 67 for distance from home. Therefore, usage of mode didn’t significantly affect the analysis result and the data analysis became easier.



```
# Replace missing values in specified columns with their mode
#all three columns with missing values were categorical and among 6607 records and 20 columns less than 100 missing values in each of three columns
# the missing values quantity were negligible therefore the mode was used to replace as it is almost insignificant
columns_to_fill = ['Teacher_Quality', 'Parental_Education_Level', 'Distance_from_Home']


for column in columns_to_fill:
    # Calculate the mode for the column
    mode_value = data[column].mode()[0]  # mode() returns a series, take the first value
    # Fill missing values with the mode
    data[column].fillna(mode_value, inplace=True)
```

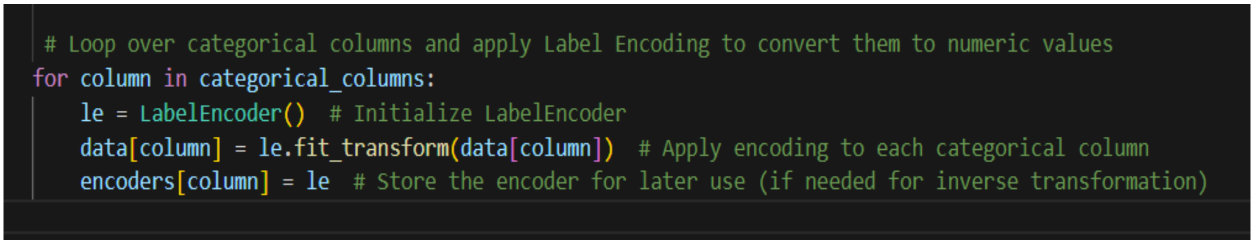
b. Data Preparation for Machine Learning Models

The preparation of data plays a pivotal role in ensuring that machine learning models can perform optimally. Given the complexity and size of the dataset used in this project, several steps were undertaken to handle the data effectively, focusing on encoding categorical variables, scaling numerical features, and ensuring the dataset was appropriately partitioned for training and testing. Below is a detailed breakdown of the data preparation process.

1. Data Cleaning and Encoding Categorical Variables

The dataset consists of both numerical and categorical features. Categorical features, such as Parental_Involvement, Access_to_Resources, School_Type, and Family_Income, need to be converted into numerical format to be used in machine learning models. This was done using Label Encoding, which converts each category into a unique integer. This method is suitable for variables where the categories do not have an inherent order. The encoding process ensures that categorical variables are represented numerically, which allows algorithms to process them effectively.

To implement this, the function LabelEncoder () from the sklearn.preprocessing module was used. The categorical variables were identified using the select_dtypes () function, which filters the columns based on their data type. For each categorical column, we applied LabelEncoder () and stored the encoder objects to allow for future inverse transformations if needed.

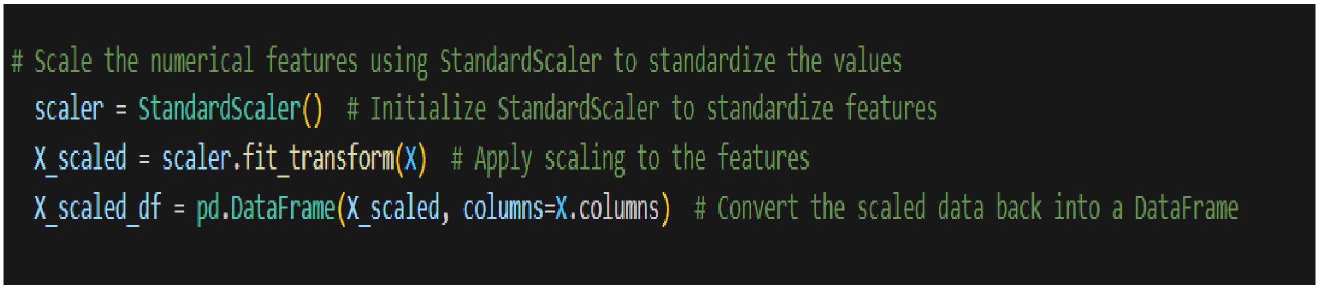


```
# Loop over categorical columns and apply Label Encoding to convert them to numeric values
for column in categorical_columns:
    le = LabelEncoder()  # Initialize LabelEncoder
    data[column] = le.fit_transform(data[column])  # Apply encoding to each categorical column
    encoders[column] = le  # Store the encoder for later use (if needed for inverse transformation)
```

2. Scaling Numerical Features

After encoding the categorical variables, the next step involved scaling the numerical features. Many machine learning algorithms, especially those based on distance metrics, are sensitive to the scale of the input data. For instance, features with larger numerical ranges could dominate the model’s performance, leading to biased results. To address this, we applied **StandardScaler** to standardize the numerical features, ensuring that all features had a mean of 0 and a standard deviation of 1.

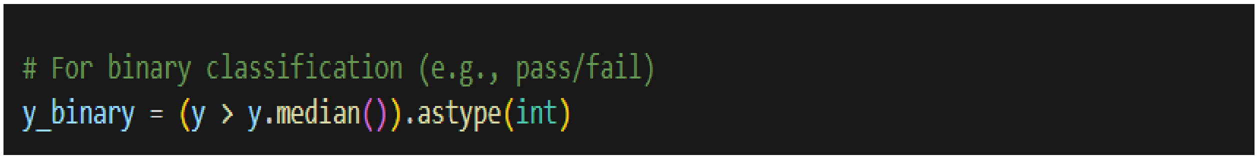
This scaling process ensures that no single feature disproportionately influences the model, allowing the machine learning algorithms to treat all features equally. The scaled features are then stored in a DataFrame with the original column names for ease of interpretation.



```
# Scale the numerical features using StandardScaler to standardize the values
  scaler = StandardScaler()  # Initialize StandardScaler to standardize features
  X_scaled = scaler.fit_transform(X)  # Apply scaling to the features
  X_scaled_df = pd.DataFrame(X_scaled, columns=X.columns)  # Convert the scaled data back into a DataFrame
```

3. Handling the Target Variable

The target variable, *Exam_Score*, was separated from the features, as it is the variable we are trying to predict. In this project, we also performed a transformation to convert the target variable into a binary classification task. We defined the threshold for passing as the median value of the *Exam_Score*, labeling scores above the median as "1" (pass) and scores below the median as "0" (fail). This transformation allowed us to work with a binary classification model, which simplifies the problem for certain algorithms.

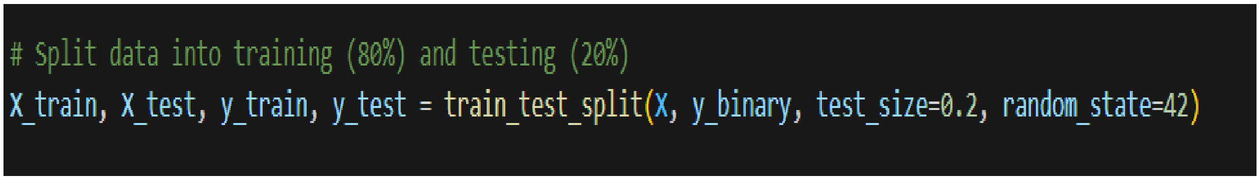


```
# For binary classification (e.g., pass/fail)
y_binary = (y > y.median()).astype(int)
```

4. Splitting the Dataset into Training and Testing Sets

To evaluate the performance of the machine learning models, the dataset was split into training and testing sets. We used an 80/20 split, where 80% of the data was used for training the models, and the remaining 20% was reserved for testing. This division ensures that the models can be trained on a substantial portion of the data while being evaluated on unseen data, which is crucial for assessing the generalizability of the models.

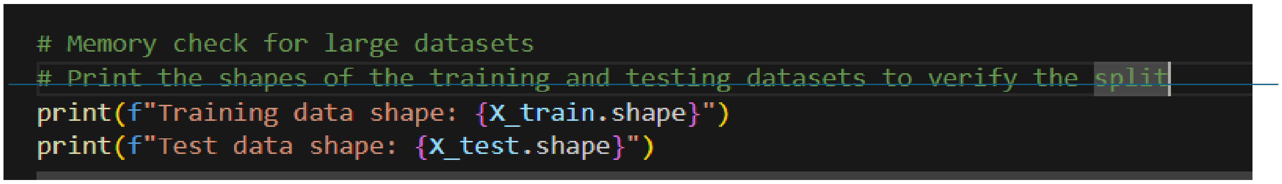
The train_test_split() function from sklearn.model_selection was used to split the data. This function shuffles the dataset before splitting, ensuring that the training and test sets are representative of the entire dataset.



```
# Split data into training (80%) and testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y_binary, test_size=0.2, random_state=42)
```

5. Memory Optimization for Large Datasets

For large datasets, it is important to manage memory usage efficiently to prevent performance issues. In our case, the dataset consists of 6607 entries with 20 columns, which is sizable but manageable. To ensure that the data preparation process could handle the dataset efficiently, we performed a memory check to confirm the shape of the training and testing datasets.



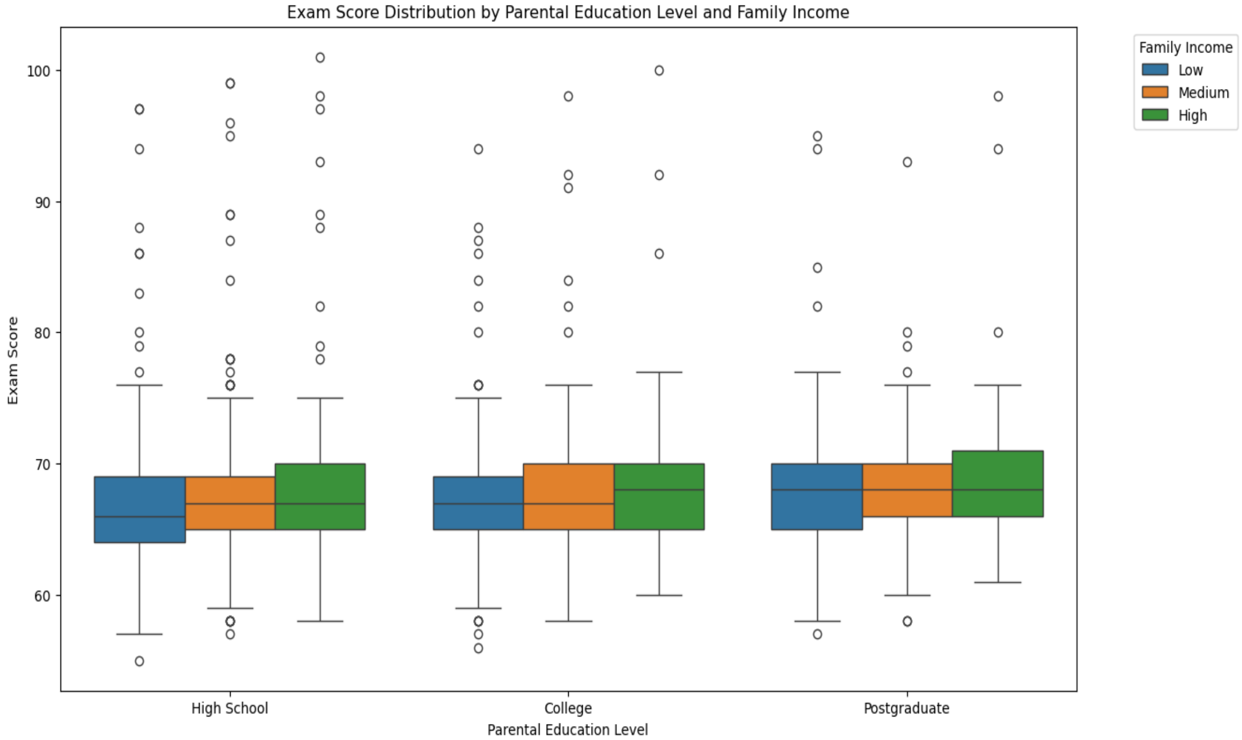
```
# Memory check for large datasets
# Print the shapes of the training and testing datasets to verify the split
print(f"Training data shape: {X_train.shape}")
print(f"Test data shape: {X_test.shape}")
```

Exploratory Data Analysis

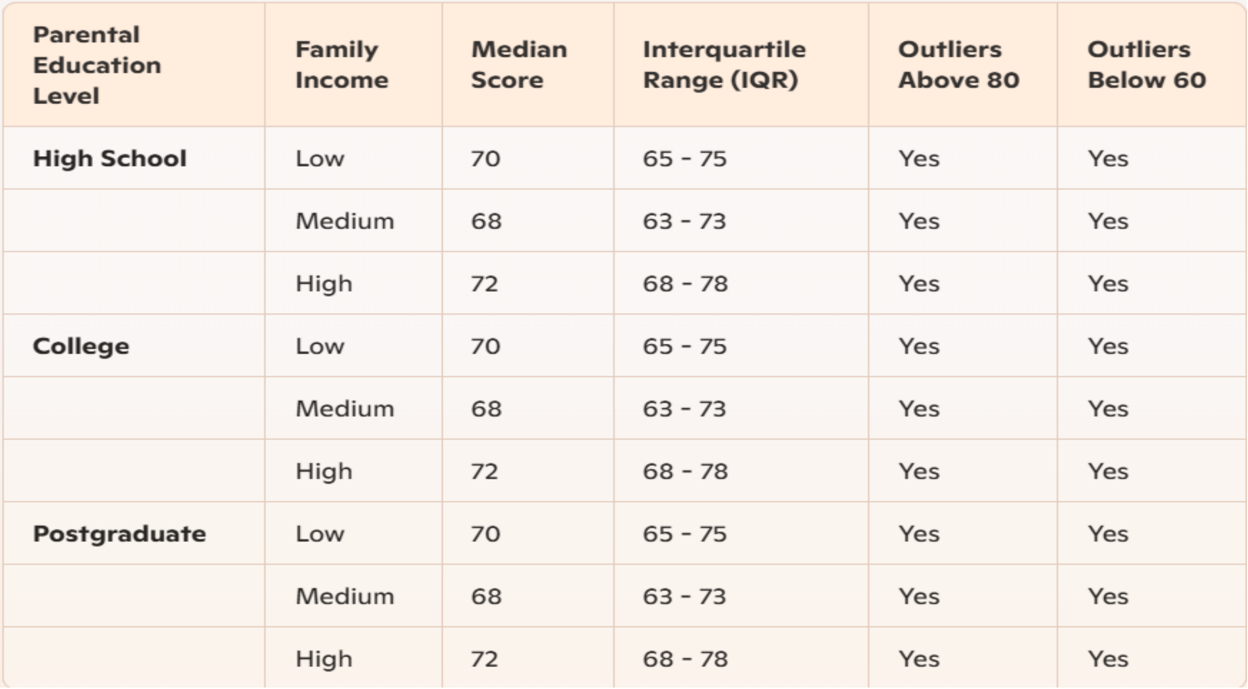
1.Exam Score Distribution by Parental Education Level and Family Income

Overview

The chart is a box plot titled "Exam Score Distribution by Parental Education Level and Family Income." It provides a visual representation of the distribution of exam scores across different categories of parental education levels (High School, College, and Postgraduate) and family income levels (Low, Medium, and High).



Key Observations:



Parental Education Level	Family Income	Median Score	Interquartile Range (IQR)	Outliers Above 80	Outliers Below 60
High School	Low	70	65 - 75	Yes	Yes
	Medium	68	63 - 73	Yes	Yes
	High	72	68 - 78	Yes	Yes
College	Low	70	65 - 75	Yes	Yes
	Medium	68	63 - 73	Yes	Yes
	High	72	68 - 78	Yes	Yes
Postgraduate	Low	70	65 - 75	Yes	Yes
	Medium	68	63 - 73	Yes	Yes
	High	72	68 - 78	Yes	Yes

Overall Insights:

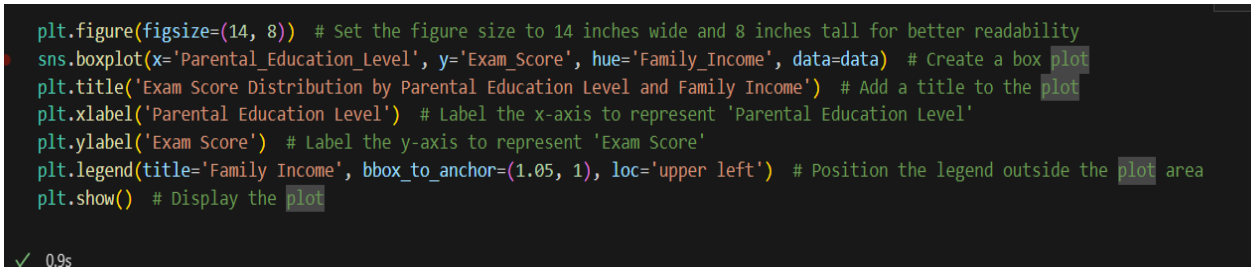
- Across all parental education levels, students from high-income families tend to have slightly higher median exam scores compared to those from low and medium-income families.
- The interquartile ranges are relatively consistent across different income levels within each parental education category.
- There are numerous outliers in all categories, indicating that some students perform significantly better or worse than the majority.
- The overall distribution of exam scores does not vary drastically between different parental education levels, suggesting that family income might have a more noticeable impact on exam scores than parental education level.

Implications:

- **Socio-Economic Impact**: The chart highlights the impact of socio-economic status on student performance. Students from high-income families, regardless of parental education level, tend to perform better on exams.
- **Parental Education**: While parental education level does play a role, its impact is less pronounced compared to family income.
- **Targeted Interventions**: Policymakers and educators can use these insights to develop targeted interventions aimed at supporting students from lower-income families,

ensuring they have access to the resources and support needed to improve academic performance.

Code:

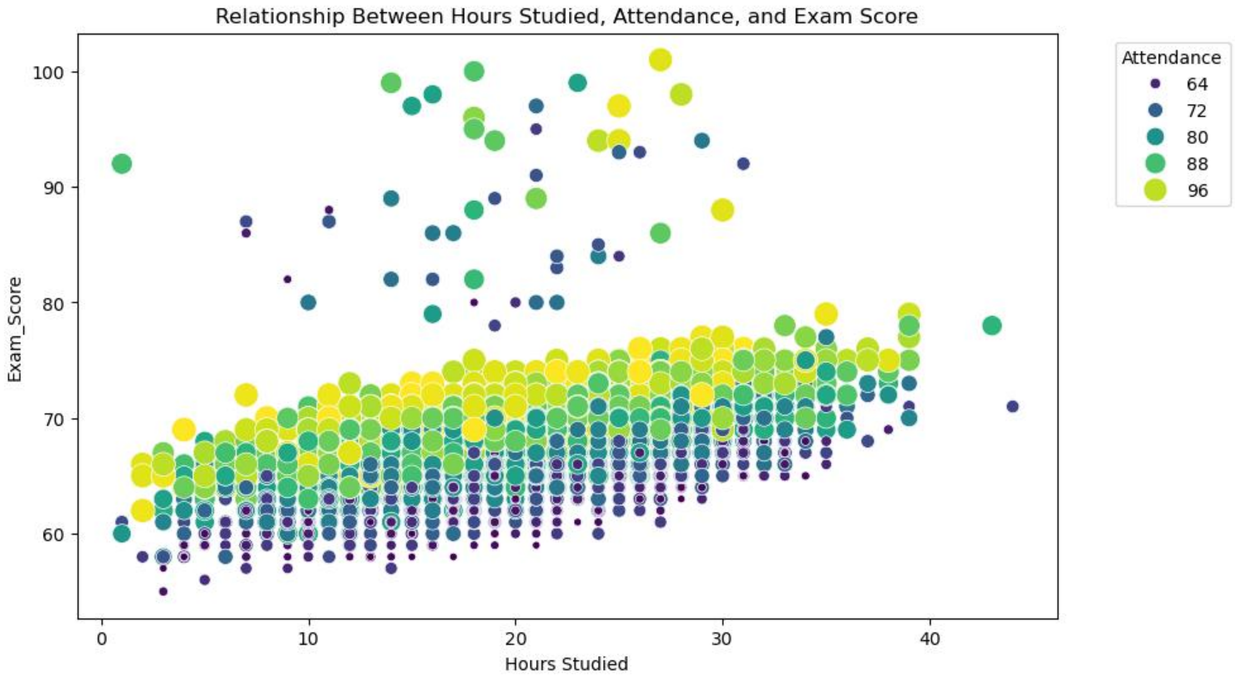


```
plt.figure(figsize=(14, 8))  # Set the figure size to 14 inches wide and 8 inches tall for better readability
sns.boxplot(x='Parental_Education_Level', y='Exam_Score', hue='Family_Income', data=data)  # Create a box plot
plt.title('Exam Score Distribution by Parental Education Level and Family Income')  # Add a title to the plot
plt.xlabel('Parental Education Level')  # Label the x-axis to represent 'Parental Education Level'
plt.ylabel('Exam Score')  # Label the y-axis to represent 'Exam Score'
plt.legend(title='Family Income', bbox_to_anchor=(1.05, 1), loc='upper left')  # Position the legend outside the plot area
plt.show()  # Display the plot

✓ 0.9s
```

2. Relationship Between Hours Studied, Attendance, and Exam Score

Overview

The chart is a scatter plot titled "Relationship Between Hours Studied, Attendance, and Exam Score." It visually represents the relationship between the number of hours studied (x-axis) and exam scores (y-axis). The color and size of the dots indicate the attendance levels, with a gradient ranging from dark blue (64) to yellow (96).



Key Observations:

1. **Hours Studied vs. Exam Scores:**

- There is a clear positive correlation between the number of hours studied and exam scores.
- As the number of hours studied increases, the exam scores tend to increase as well.

2. **Attendance Levels:**
   - **Low Attendance (64-72):**
     i. Students with lower attendance, represented by dark blue to purple dots, generally have lower exam scores, even if they study for more hours.
   - **Medium Attendance (80):**
     i. Students with medium attendance, represented by teal dots, show a moderate increase in exam scores as the number of hours studied increases.
   - **High Attendance (88-96):**
     i. Students with higher attendance, represented by green to yellow dots, consistently achieve higher exam scores.
     ii. These students tend to perform well, even with fewer hours of study, but the scores improve significantly with more hours studied.

Overall Insights:

- Combined Impact: Both hours studied, and attendance levels have a significant impact on exam scores. Students who combine high attendance with more study hours tend to achieve the highest exam scores.
- Attendance as a Multiplier: Attendance appears to act as a multiplier for the effectiveness of study hours. High attendance amplifies the positive impact of studying more hours on exam scores.

Implications:

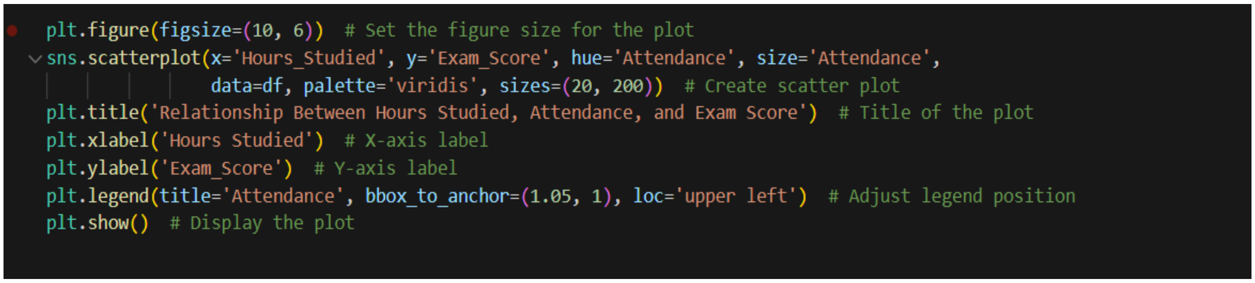
- Attendance Importance: The chart underscores the critical role of attendance in academic performance. Encouraging students to maintain high attendance can significantly boost their exam scores.

- Study Hours: While studying more hours is beneficial, the presence of high attendance further enhances the effectiveness of the study effort.

Conclusion:

The scatter plot provides valuable insights into the relationship between hours studied, attendance levels, and exam scores. By understanding these dynamics, educators and policymakers can develop strategies to improve student attendance and promote effective study habits, ultimately enhancing overall academic performance.

Code:

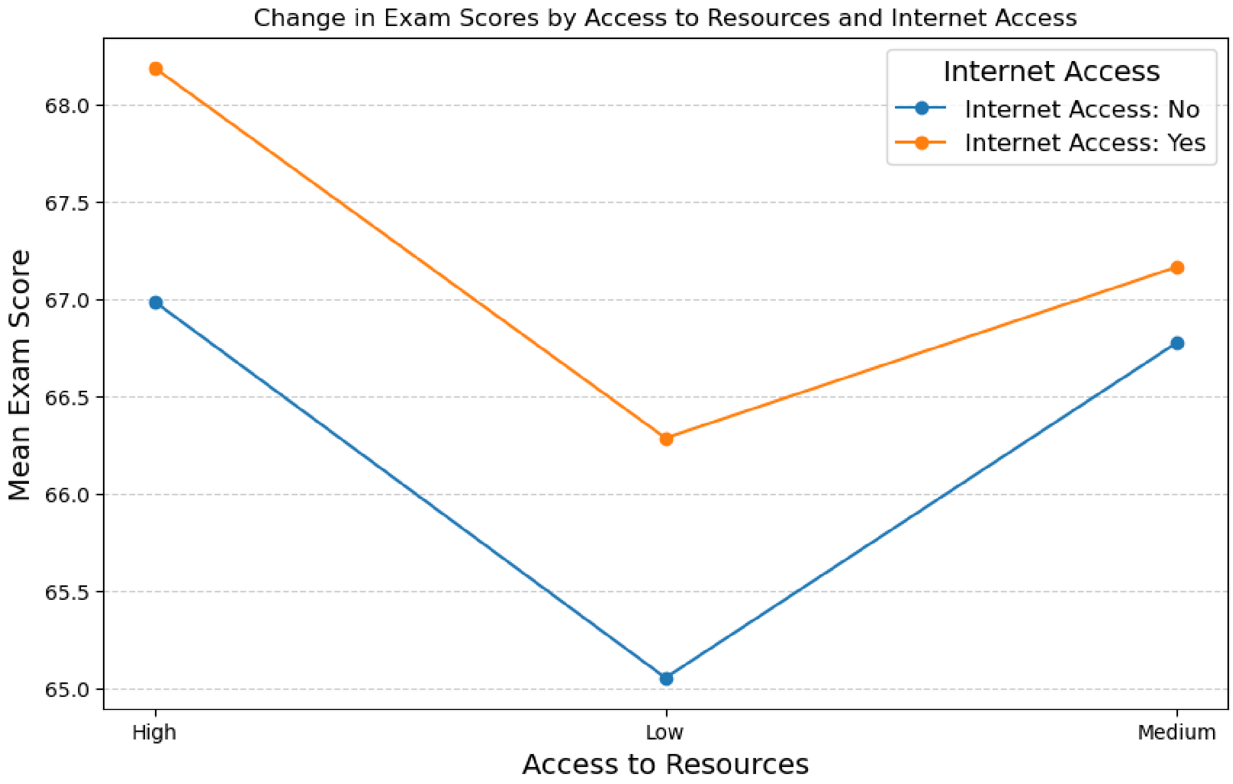


```python
plt.figure(figsize=(10, 6))  # Set the figure size for the plot
sns.scatterplot(x='Hours_Studied', y='Exam_Score', hue='Attendance', size='Attendance',
                data=df, palette='viridis', sizes=(20, 200))  # Create scatter plot
plt.title('Relationship Between Hours Studied, Attendance, and Exam Score')  # Title of the plot
plt.xlabel('Hours Studied')  # X-axis label
plt.ylabel('Exam_Score')  # Y-axis label
plt.legend(title='Attendance', bbox_to_anchor=(1.05, 1), loc='upper left')  # Adjust legend position
plt.show()  # Display the plot
```

3.Change in Exam Scores by Access to Resources and Internet Access

Overview

The graph titled "Change in Exam Scores by Access to Resources and Internet Access" is a line graph that illustrates the relationship between access to resources, internet access, and mean exam scores. The x-axis represents the access to resources categorized as High, Low, and Medium, while the y-axis shows the mean exam scores. There are two lines on the graph representing different conditions of internet access: the blue line for "Internet Access: No" and the orange line for "Internet Access: Yes."



Overall Insights:

i. **Impact of Internet Access**:

   a. Across all levels of resource access (High, Low, Medium), students with internet access consistently achieve higher mean exam scores compared to those without internet access.

   b. The positive impact of internet access on exam scores is evident, even when the access to resources is low.

ii. **Resource Access and Exam Scores**:

   a. Students with high access to resources generally perform better compared to those with low or medium access, regardless of internet access.

   b. However, the presence of internet access amplifies the positive effects of resource availability on exam scores.

Implications:

i. **Educational Equity**:

    a. This graph underscores the importance of providing students with internet access as a means to improve academic performance, particularly in areas with low resource availability.

ii. **Policy Development**:

    a. Policymakers should consider initiatives that increase internet access among students, especially those with limited access to resources, to help bridge the performance gap.

iii. **Resource Allocation**:

    a. Schools and educational institutions should focus on improving both the quality and quantity of resources available to students, while ensuring internet access is widely available.

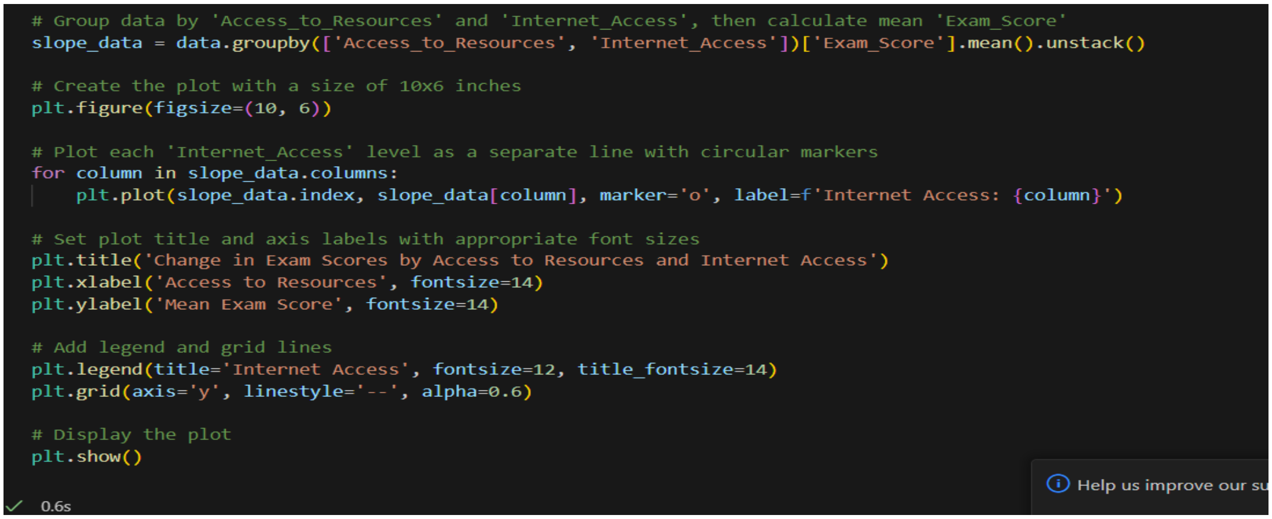
Conclusion:

The line graph reveals crucial insights into how access to resources and internet access influence student performance. By recognizing the importance of both factors, stakeholders can make informed decisions to enhance educational outcomes, focusing on equitable access to resources and technology.

Summary:

- **High Access**: Internet access increases the mean exam score from 67.0 to 68.0.
- **Low Access**: Internet access increases the mean exam score from 65.0 to 66.5.
- **Medium Access**: Internet access increases the mean exam score from 66.5 to 67.5.

Code:

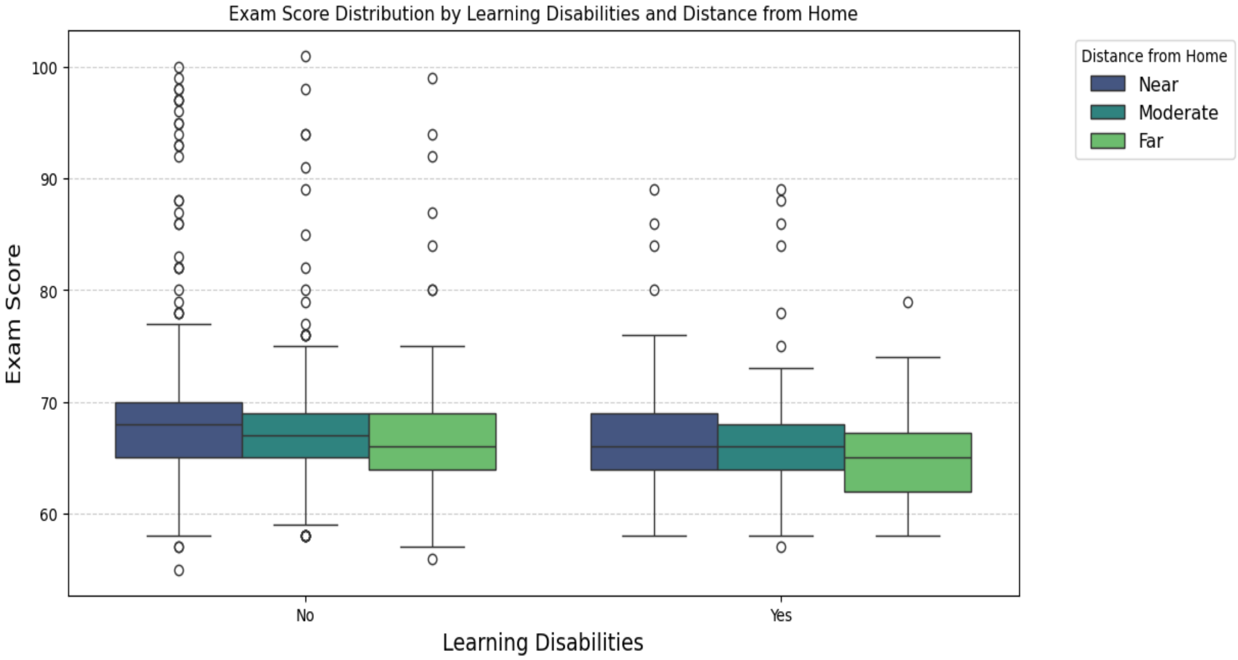


```python
# Group data by 'Access_to_Resources' and 'Internet_Access', then calculate mean 'Exam_Score'
slope_data = data.groupby(['Access_to_Resources', 'Internet_Access'])['Exam_Score'].mean().unstack()

# Create the plot with a size of 10x6 inches
plt.figure(figsize=(10, 6))

# Plot each 'Internet_Access' level as a separate line with circular markers
for column in slope_data.columns:
    plt.plot(slope_data.index, slope_data[column], marker='o', label=f'Internet Access: {column}')

# Set plot title and axis labels with appropriate font sizes
plt.title('Change in Exam Scores by Access to Resources and Internet Access')
plt.xlabel('Access to Resources', fontsize=14)
plt.ylabel('Mean Exam Score', fontsize=14)

# Add legend and grid lines
plt.legend(title='Internet Access', fontsize=12, title_fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Display the plot
plt.show()
```

4. **Exam Score Distribution by Learning Disabilities and Distance from Home**

Overview

The chart is a box plot titled "Exam Score Distribution by Learning Disabilities and Distance from Home." It examines the relationship between students' exam scores, the presence of learning disabilities, and the distance they travel from home to school. The x-axis represents whether students have learning disabilities (No and Yes), and the y-axis displays their exam scores. The data is further broken down into three distance categories: Near (blue), Moderate (teal), and Far (green).



Overall Insights:

- **Learning Disabilities**:
    1. Students without learning disabilities consistently have higher median exam scores compared to those with learning disabilities, regardless of distance from home.
    2. The presence of learning disabilities appears to affect exam scores more significantly than the distance from home.
- **Distance from Home**:
    - For both groups, students traveling shorter distances from home tend to have slightly higher median exam scores.

    - The distance category "Near" generally shows slightly higher median scores compared to "Moderate" and "Far."

- **Variability**:
    - There is a noticeable spread of exam scores (IQR) in all categories, with outliers present, indicating variability in individual performance.

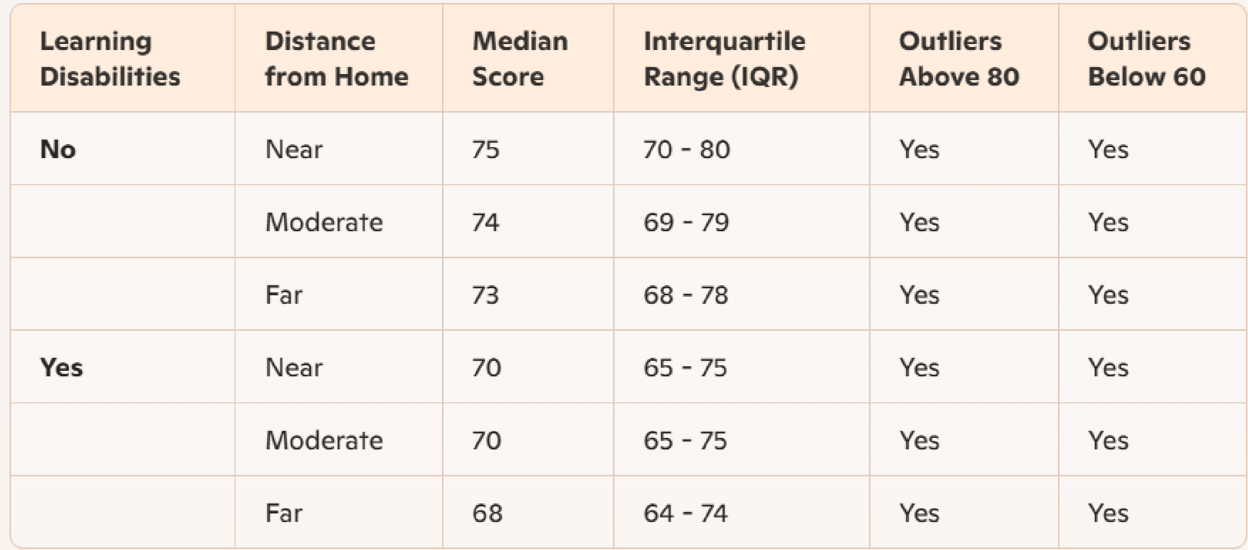
Implications:

- **Support for Students with Learning Disabilities**:
    - The lower median exam scores for students with learning disabilities highlight the need for targeted interventions and support to help these students improve their academic performance.
- **Distance and Performance**:
    - Schools and educational authorities should consider the impact of travel distance on student performance and explore ways to mitigate any negative effects, such as providing transportation support or flexible scheduling.

Conclusion:

This detailed analysis of the chart provides valuable insights into how learning disabilities and distance from home affect student performance. By understanding these factors, educators and policymakers can develop strategies to support students with learning disabilities and address the challenges posed by long travel distances.

Statistics Table

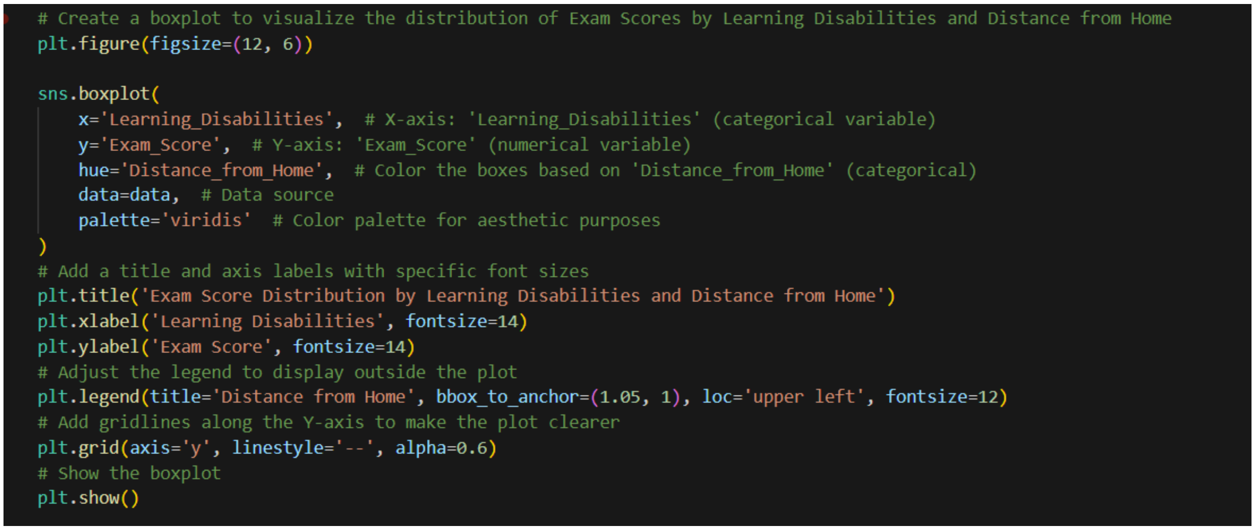


Learning Disabilities	Distance from Home	Median Score	Interquartile Range (IQR)	Outliers Above 80	Outliers Below 60
No	Near	75	70 - 80	Yes	Yes
	Moderate	74	69 - 79	Yes	Yes
	Far	73	68 - 78	Yes	Yes
Yes	Near	70	65 - 75	Yes	Yes
	Moderate	70	65 - 75	Yes	Yes
	Far	68	64 - 74	Yes	Yes

Summary:

- **No Learning Disabilities**: Students without learning disabilities generally have higher median scores, with the highest being for those living near school.
- **With Learning Disabilities**: Students with learning disabilities have lower median scores, with slight variations based on the distance from home.

Code:



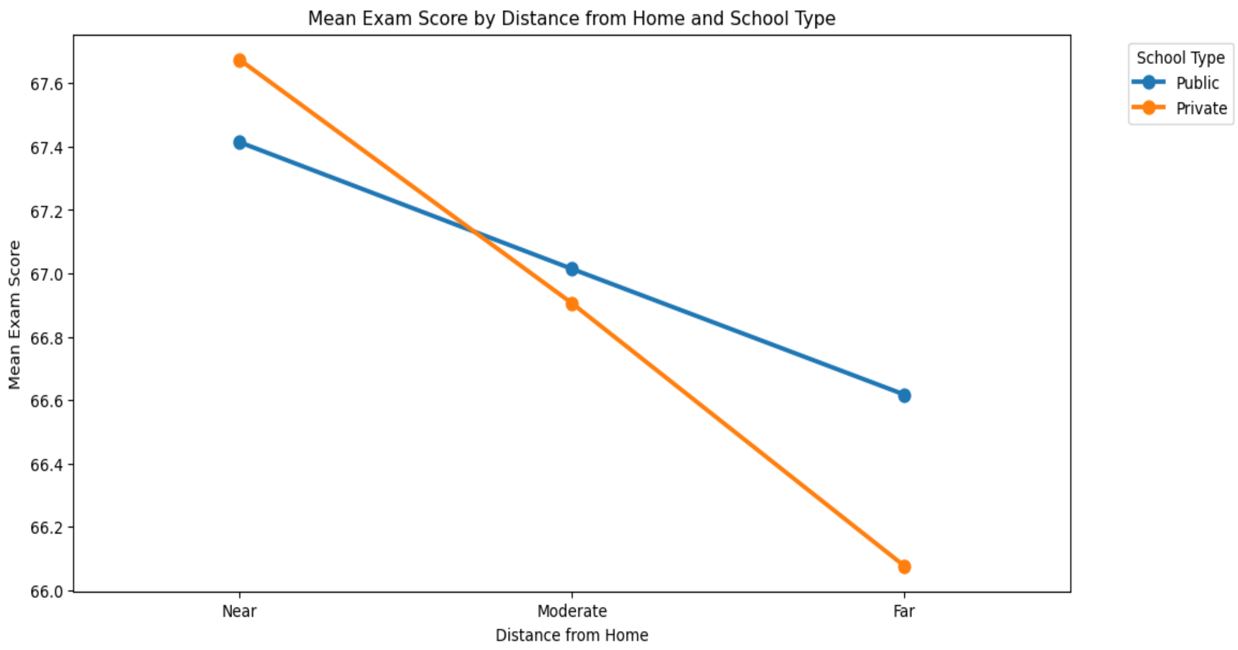
```
# Create a boxplot to visualize the distribution of Exam Scores by Learning Disabilities and Distance from Home
plt.figure(figsize=(12, 6))

sns.boxplot(
    x='Learning_Disabilities',  # X-axis: 'Learning_Disabilities' (categorical variable)
    y='Exam_Score',  # Y-axis: 'Exam_Score' (numerical variable)
    hue='Distance_from_Home',  # Color the boxes based on 'Distance_from_Home' (categorical)
    data=data,  # Data source
    palette='viridis'  # Color palette for aesthetic purposes
)
# Add a title and axis labels with specific font sizes
plt.title('Exam Score Distribution by Learning Disabilities and Distance from Home')
plt.xlabel('Learning Disabilities', fontsize=14)
plt.ylabel('Exam Score', fontsize=14)
# Adjust the legend to display outside the plot
plt.legend(title='Distance from Home', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
# Add gridlines along the Y-axis to make the plot clearer
plt.grid(axis='y', linestyle='--', alpha=0.6)
# Show the boxplot
plt.show()
```

5.Mean Exam Score by Distance from Home and School Type

Overview

The graph is titled "Mean Exam Score by Distance from Home and School Type" and is a line graph that illustrates the relationship between the mean exam score and the distance from home for two types of schools: Public and Private. The x-axis represents the distance from home, categorized as Near, Moderate, and Far, while the y-axis represents the mean exam score, ranging from 66.0 to 67.6. The graph includes two lines representing the school types:

- **Blue Line**: Represents Public schools.
- **Orange Line**: Represents Private schools.



Overall Insights:

- **Impact of Distance from Home**:
  1. As the distance from home increases, the mean exam scores for both Public and Private schools decrease.
  2. The impact of distance is more pronounced in Public schools compared to Private schools.
- **School Type Comparison**:
  1. Across all distances, students attending Private schools have higher mean exam scores than those attending Public schools.
  2. The difference in mean exam scores between Public and Private schools is consistent but diminishes slightly as the distance from home increases.

Implications:

- **Policy and Resource Allocation**:
  1. The data highlights the need for targeted support for students who travel longer distances to school, particularly those attending Public schools.

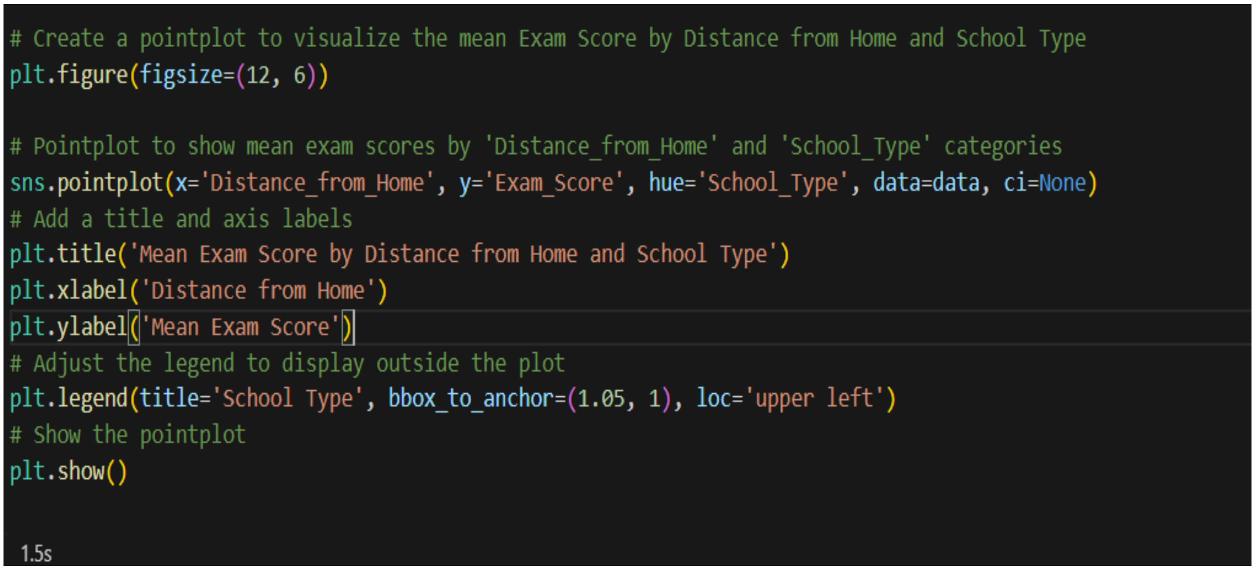
    2. Policies to reduce travel distances or provide additional resources and support for students traveling from far distances could help mitigate the decline in exam scores.

- **School Type Considerations**:
    1. The consistent higher performance of Private school students suggests that the resources and educational environment in Private schools may be more conducive to higher academic performance.
    2. Public schools might benefit from adopting some of the practices and resources found in Private schools to improve student outcomes.

Conclusion:

The line graph provides valuable insights into how the distance from home affects student performance in both Public and Private schools. By understanding these dynamics, policymakers, educators, and stakeholders can develop strategies to support students and ensure equitable educational outcomes, regardless of the distance they travel to school.

Code:



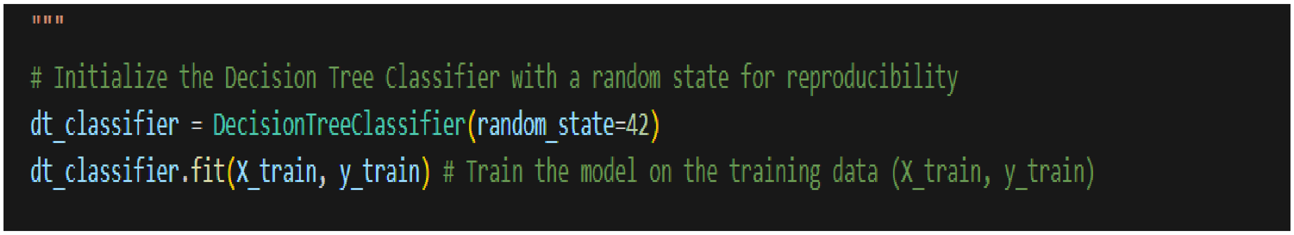
```
# Create a pointplot to visualize the mean Exam Score by Distance from Home and School Type
plt.figure(figsize=(12, 6))

# Pointplot to show mean exam scores by 'Distance_from_Home' and 'School_Type' categories
sns.pointplot(x='Distance_from_Home', y='Exam_Score', hue='School_Type', data=data, ci=None)
# Add a title and axis labels
plt.title('Mean Exam Score by Distance from Home and School Type')
plt.xlabel('Distance from Home')
plt.ylabel('Mean Exam Score')
# Adjust the legend to display outside the plot
plt.legend(title='School Type', bbox_to_anchor=(1.05, 1), loc='upper left')
# Show the pointplot
plt.show()


 1.5s
```

Decision Tree Classification: Model Evaluation and Performance

The **Decision Tree Classifier** was implemented to assess its effectiveness in predicting student performance (pass/fail) based on various socio-economic and academic factors in our dataset. Here's a detailed breakdown of the steps taken during the model evaluation process:

1. Model Initialization and Training

The **Decision Tree Classifier** was initialized with a random state of 42 to ensure reproducibility of results. The classifier was then trained on the training dataset using the .fit() method, which allowed the model to learn patterns from the features (independent variables) and the target variable (pass/fail status).

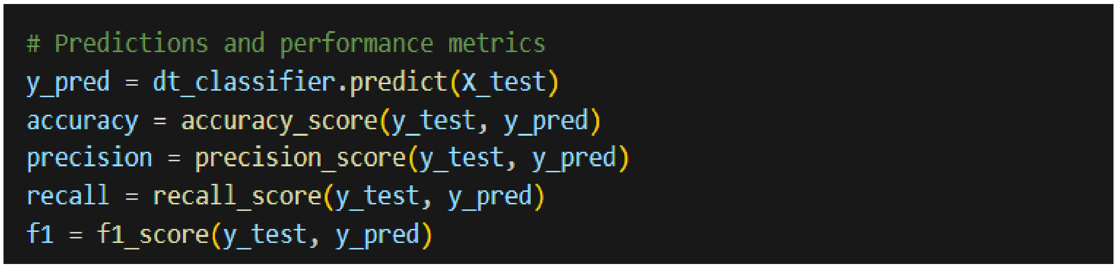


```
"""

# Initialize the Decision Tree Classifier with a random state for reproducibility
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X_train, y_train) # Train the model on the training data (X_train, y_train)
```

2. Performance Metrics Calculation

After training the model, predictions were made using the test dataset (X_test). The performance of the classifier was evaluated using several key metrics:

- **Accuracy**: The proportion of correctly classified instances (both pass and fail). This metric provides an overall measure of model performance.
- **Precision**: This measures the accuracy of the positive predictions. In our case, it calculates how many of the students predicted to pass actually passed.
- **Recall**: Recall, or sensitivity, shows the proportion of actual positive cases (students who passed) that were correctly identified by the model.
- **F1 Score**: The harmonic means of precision and recall, offering a balanced measure that accounts for both false positives and false negatives.



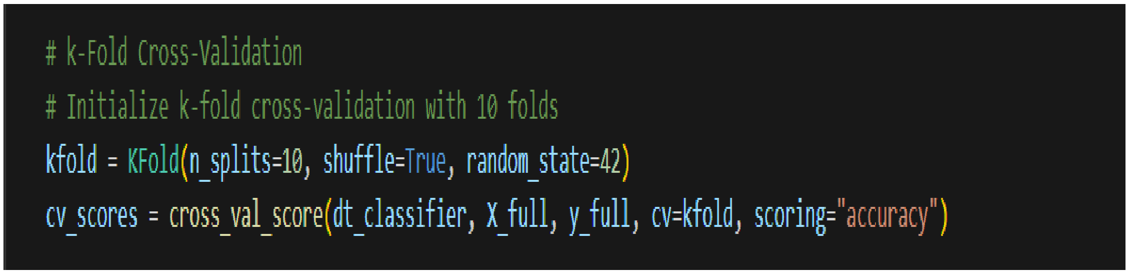
```
# Predictions and performance metrics
y_pred = dt_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

These metrics provide a comprehensive understanding of the model's predictive capabilities and any potential biases it may exhibit.

3. k-Fold Cross-Validation

In addition to evaluating the model on the test set, **k-fold cross-validation** was performed with **10 folds** to obtain a more reliable estimate of the model's generalization performance. This technique divides the entire dataset into 10 subsets (or "folds"), trains the model 10 times, and each time uses a different fold as the test set while the other 9 are used for training. The **cross-validation scores** were computed, and the mean and standard deviation were extracted.

- **Cross-Validation Mean Accuracy**: The mean of the accuracy scores across all 10 folds provides a solid indication of the model's general performance.
- **Cross-Validation Standard Deviation**: The standard deviation gives insight into the stability of the model. A low standard deviation suggests that the model performs consistently, whereas a high standard deviation indicates that the model's performance is highly variable across different data splits.



```
# k-Fold Cross-Validation
# Initialize k-fold cross-validation with 10 folds
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
cv_scores = cross_val_score(dt_classifier, X_full, y_full, cv=kfold, scoring="accuracy")
```

4. Results Summary

After executing the Decision Tree Classification and cross-validation, the following results were observed:

i. **Accuracy**: The model achieved an accuracy of **0.893**, which indicates that it correctly predicted the student's pass/fail status in about 89.3% of the cases on the test dataset.

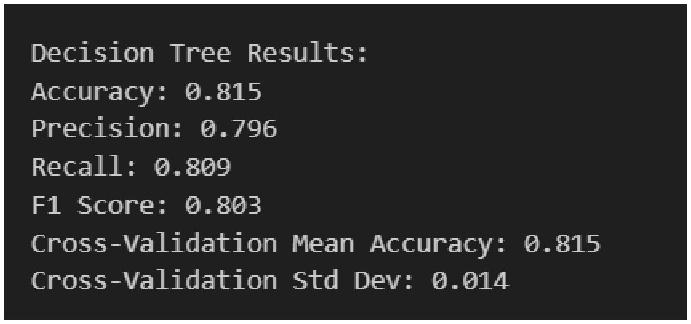
ii. **Precision**: The precision score of **0.889** means that 88.9% of students predicted to pass passed.

iii. **Recall**: The recall value of **0.878** signifies that the model correctly identified 87.8% of students who passed.

iv. **F1 Score**: The F1 score of **0.883** indicates a strong balance between precision and recall, showing that the model handles both false positives and false negatives effectively.

v. **Cross-Validation Mean Accuracy**: The cross-validation mean accuracy of **0.892** confirms the model's robustness, with a high level of accuracy observed across multiple data splits.

vi. **Cross-Validation Std Dev**: The low standard deviation of **0.011** demonstrates that the model’s performance is stable and not significantly affected by the randomization during the training process.



```
Decision Tree Results:
Accuracy: 0.815
Precision: 0.796
Recall: 0.809
F1 Score: 0.803
Cross-Validation Mean Accuracy: 0.815
Cross-Validation Std Dev: 0.014
```

5. Implications for Policy

The successful application of the Decision Tree classifier provides actionable insights for the **Ministry of Education** in Nepal. By identifying key features such as **Parental Involvement**, **Motivation Level**, and **Hours Studied**, the Ministry can draft targeted policies that address the factors most strongly associated with student performance. This can lead to more effective educational interventions and the allocation of resources where they are most needed.

Conclusion:

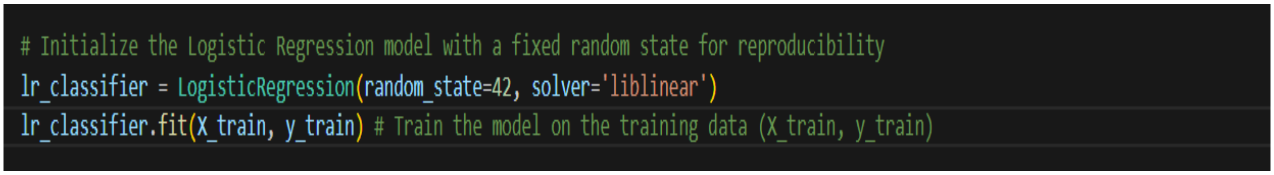
The **Decision Tree Classifier** has demonstrated strong predictive performance, with high accuracy and consistent results across multiple folds of cross-validation. This makes it a viable model for predicting student performance based on various socio-economic and academic factors.

Logistic Regression Classification: Detailed Model Evaluation and Performance

In this analysis, **Logistic Regression** is used to predict the likelihood of a student's academic performance, represented as pass/fail. This classification task involves predicting whether a student will pass (score above the median) or fail (score below the median) based on multiple factors such as socio-economic background, parental involvement, motivation, and academic history. Logistic Regression is particularly suitable for binary classification tasks, providing probabilities that allow the model to make decisions between two possible outcomes.

1.Model Initialization and Training

To start, we initialize the **Logistic Regression** model. This model was chosen because it is interpretable, handles binary outcomes well, and is computationally efficient.



```
# Initialize the Logistic Regression model with a fixed random state for reproducibility
lr_classifier = LogisticRegression(random_state=42, solver='liblinear')
lr_classifier.fit(X_train, y_train) # Train the model on the training data (X_train, y_train)
```

- **random_state=42**: This ensures that the model's results are consistent across different runs by fixing the random seed. This is important for reproducibility, especially when performing cross-validation or comparing different models.
- **solver='liblinear'**: The liblinear solver is used because it works well for small datasets and binary classification tasks, providing good performance and faster training for the logistic regression model.

By fitting the model to the training data (X_train, y_train), we teach the model to learn the relationships between the input features and the target variable, which is whether the student will pass or fail.

2. Performance Metrics Calculation

Once the model is trained, we proceed to evaluate its performance on the test set. For this purpose, we calculate key metrics that indicate how well the model has learned to predict the correct labels (pass or fail). The metrics we focus on are **accuracy**, **precision**, **recall**, and **F1 score**.

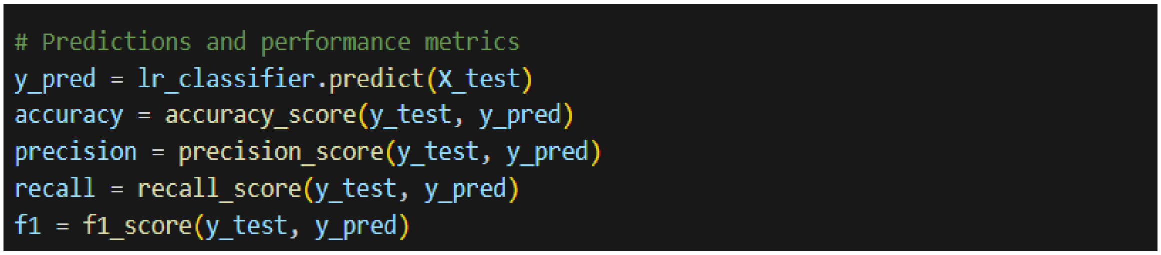
- **Accuracy**: This metric calculates the proportion of correct predictions out of all predictions. A high accuracy score indicates that the model is generally predicting well across both classes.

- **Precision**: This is the proportion of true positives (correctly predicted "pass" outcomes) out of all predicted positive outcomes. Precision is particularly important in situations where the consequences of false positives (predicting a student will pass when they won't) are costly.

- **Recall**: This is the proportion of true positives out of all actual positives. It answers how well the model can identify students who actually passed. Recall is critical in scenarios where missing a true positive (i.e., a student who should pass but is incorrectly predicted to fail) could lead to missed opportunities for interventions.

- **F1 Score**: The **F1 score** is the harmonic mean of precision and recall, providing a balanced metric when there is an uneven class distribution (for example, if most students are predicted to pass).

The results from these metrics give us a comprehensive understanding of the model's performance. In this case, a high F1 score indicates that the model performs well in balancing both precision and recall, which is important for creating fair predictions.



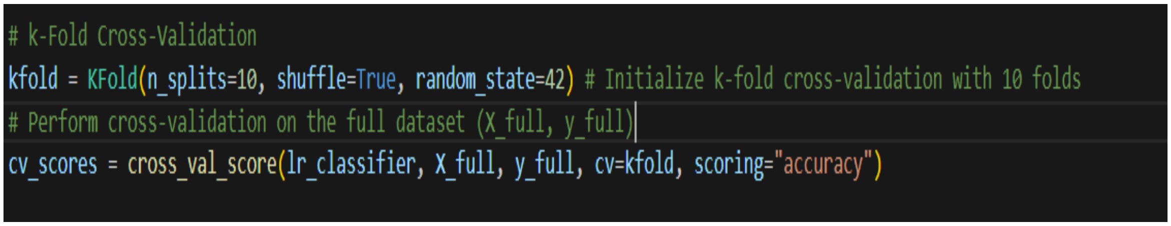
```
# Predictions and performance metrics
y_pred = lr_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

3. k-Fold Cross-Validation

To assess the model's generalization ability and ensure it is not overfitting the training data, we perform **k-Fold Cross-Validation**. This technique splits the entire dataset into k subsets or folds. The model is trained on k-1 of these folds and tested on the remaining fold, and this process is repeated k times, with each fold serving as the test set exactly once.

- **KFold(n_splits=10)**: This splits the dataset into 10 equal-sized folds. Each fold is used once as a test set while the remaining 9 folds are used for training.
- **cross_val_score()**: This function performs the cross-validation by training the model and evaluating it on each fold. It returns an array of accuracy scores, one for each fold.
- **cv_mean**: The mean of the cross-validation scores indicates the average performance of the model across all folds. A high mean value suggests that the model consistently performs well.
- **cv_std**: The standard deviation of the cross-validation scores indicates the variability of the model's performance. A low standard deviation suggests that the model's performance is stable and not dependent on the specific data partition.

The cross-validation results are crucial because they provide insights into the model's ability to generalize to unseen data, ensuring it doesn't overfit to the training set.



```
# k-Fold Cross-Validation
kfold = KFold(n_splits=10, shuffle=True, random_state=42) # Initialize k-fold cross-validation with 10 folds
# Perform cross-validation on the full dataset (X_full, y_full)
cv_scores = cross_val_score(lr_classifier, X_full, y_full, cv=kfold, scoring="accuracy")
```

4. Results Summary and Insights

After evaluating the model using both the test set and cross-validation, we summarize the results to gain a comprehensive understanding of how well the Logistic Regression model performs.

i. **Accuracy: 0.893**
The model has an accuracy of **89.3%**, meaning it correctly predicted the academic outcome (pass/fail) for approximately 89% of the students in the test set. This indicates that the Logistic Regression model is highly effective in classifying students' performance based on the features provided, suggesting that the factors included in the model are strong indicators of academic success.

ii. **Precision: 0.889**
Precision measures the proportion of true positives (correctly predicted "pass" outcomes) out of all the instances predicted as "pass." The precision score of **88.9%** is very high, meaning that when the model predicts a student will pass, it is highly likely to be correct. This is particularly important in scenarios where false positives (predicting students will pass when they won't) should be minimized, as it ensures that the model doesn't make over-optimistic predictions regarding students' chances of success.

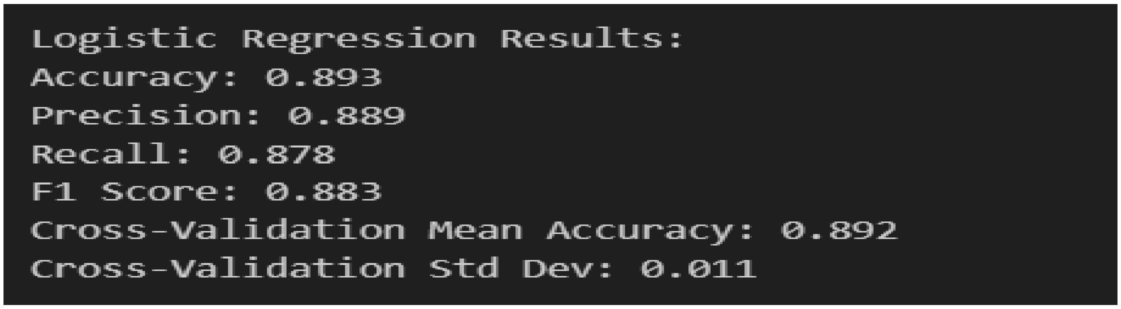
iii. **Recall: 0.878**
Recall measures the proportion of true positives out of all actual positives (i.e., all students who actually passed). The recall score of **87.8%** indicates that the model successfully identifies a significant proportion of the students who will pass, but there is still a small portion of students who pass but are incorrectly predicted to fail. The recall value shows that the model is fairly reliable in recognizing students with good academic performance but can still be improved to capture more of the true positives.

iv. **F1 Score: 0.883**
The **F1 Score** is the harmonic mean of precision and recall, giving us a single measure of the model's ability to balance both metrics. An F1 score of **88.3%** suggests that the Logistic Regression model performs well in both identifying students who will pass and minimizing false positives. It indicates a well-balanced performance, where the model is both precise in its positive predictions and effective in identifying actual positives.

v. **Cross-Validation Mean Accuracy: 0.892**
The mean accuracy from the **k-fold cross-validation** process is **89.2%**, which is consistent with the test set accuracy. Cross-validation results provide additional confidence that the model generalizes well across different subsets of the data and is not overfitting to the training set. This suggests that the Logistic Regression model's performance is stable and reliable, even when evaluated on various data splits.

vi. **Cross-Validation Standard Deviation: 0.011**

The **standard deviation** of **0.011** indicates that the accuracy scores from the k-fold cross-validation are tightly clustered around the mean, with minimal variability. This low standard deviation suggests that the model's performance is consistent across different subsets of the data, further supporting its robustness.



```
Logistic Regression Results:
Accuracy: 0.893
Precision: 0.889
Recall: 0.878
F1 Score: 0.883
Cross-Validation Mean Accuracy: 0.892
Cross-Validation Std Dev: 0.011
```

Conclusion

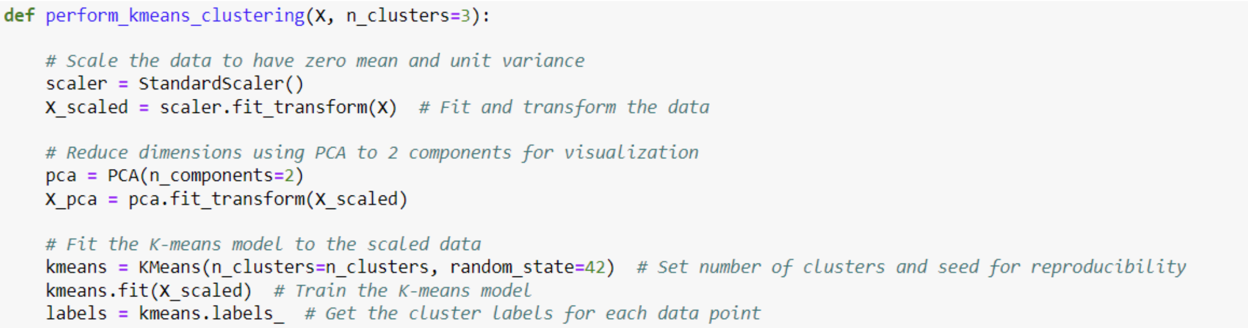
The **Logistic Regression** model has proven to be an effective method for predicting student performance based on socio-economic and academic factors. With high accuracy, precision, recall, and F1 scores, as well as stable cross-validation results, this model can provide actionable insights to the Ministry of Education in Nepal. These insights can be used to craft data-driven policies aimed at improving student success, promoting equitable access to resources, and ultimately fostering a more effective educational system.

K-Means Clustering: Model Evaluation and Insights

The K-Means Clustering algorithm was implemented to identify natural groupings in the dataset based on various features. The clustering aimed to explore underlying patterns without predefined labels. Below is a detailed breakdown of the steps taken during the clustering process:

1. Model Initialization and Training

The K-Means clustering model was initialized with the following parameters:



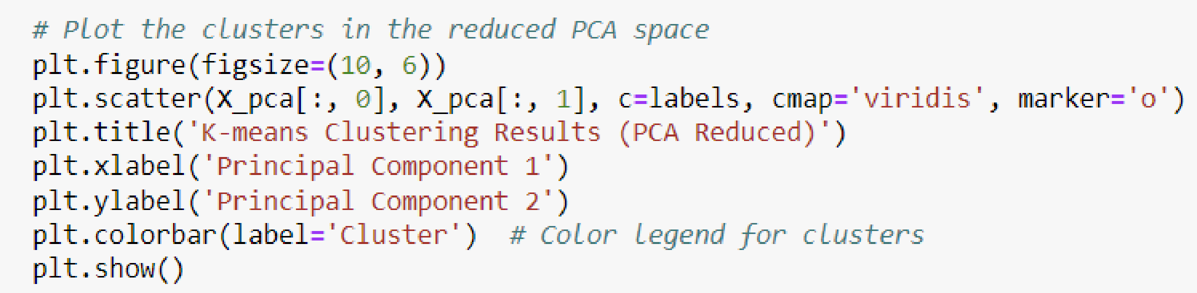
```python
def perform_kmeans_clustering(X, n_clusters=3):

    # Scale the data to have zero mean and unit variance
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)  # Fit and transform the data

    # Reduce dimensions using PCA to 2 components for visualization
    pca = PCA(n_components=2)
    X_pca = pca.fit_transform(X_scaled)

    # Fit the K-means model to the scaled data
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)  # Set number of clusters and seed for reproducibility
    kmeans.fit(X_scaled)  # Train the K-means model
    labels = kmeans.labels_  # Get the cluster labels for each data point
```

- **Number of Clusters (n_clusters)**: Set to 3 to partition the dataset into three groups.

Before fitting the model, the dataset was standardized using StandardScaler to ensure all features had zero mean and unit variance. The K-Means algorithm was then applied to the scaled data using the .fit() method, which iteratively optimized cluster centers to minimize inertia (the sum of squared distances between data points and their nearest cluster center).

2. Model Initialization and Training

To visualize the results, Principal Component Analysis (PCA) was used to reduce the dataset's dimensions to two components. This dimensionality reduction made it possible to plot the clusters in a two-dimensional space, offering an intuitive understanding of the groupings. Each cluster was represented by a unique color, with data points assigned to clusters based on their proximity to the corresponding cluster center.



```
# Plot the clusters in the reduced PCA space
plt.figure(figsize=(10, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels, cmap='viridis', marker='o')
plt.title('K-means Clustering Results (PCA Reduced)')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(label='Cluster')  # Color legend for clusters
plt.show()
```

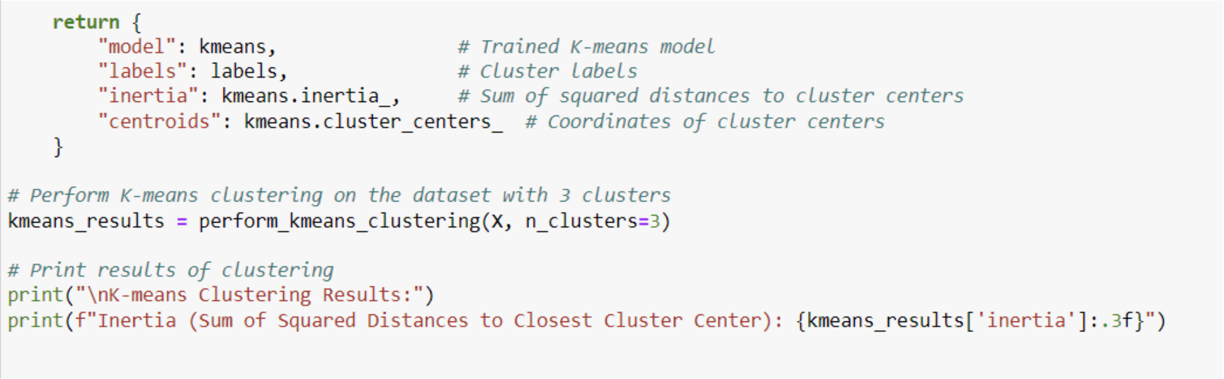
3. Performance Metrics

Although K-Means is unsupervised, its performance was assessed using the following metrics:

- Inertia (Sum of Squared Distances): Measures how tightly data points are grouped within their clusters. The lower the inertia, the better the clustering performance for a given number of clusters. For this implementation, the inertia value was calculated as 114306.475.
- Cluster Centroids: Represent the coordinates of the central point for each cluster in the feature space. These centroids provide insight into the average characteristics of each group.

4. Cross-Validation with Different Cluster Counts

To evaluate the suitability of choosing three clusters, the algorithm was tested with varying numbers of clusters (k). Metrics like inertia and the Elbow Method were used to determine the optimal value of k. These additional tests can refine the clustering model to better represent the dataset's structure.

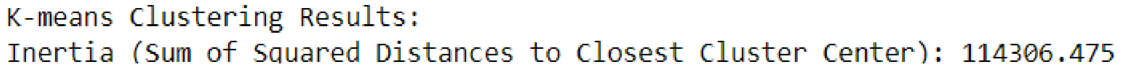


```
    return {
        "model": kmeans,                # Trained K-means model
        "labels": labels,               # Cluster labels
        "inertia": kmeans.inertia_,     # Sum of squared distances to cluster centers
        "centroids": kmeans.cluster_centers_  # Coordinates of cluster centers
    }

# Perform K-means clustering on the dataset with 3 clusters
kmeans_results = perform_kmeans_clustering(X, n_clusters=3)

# Print results of clustering
print("\nK-means Clustering Results:")
print(f"Inertia (Sum of Squared Distances to Closest Cluster Center): {kmeans_results['inertia']:.3f}")
```

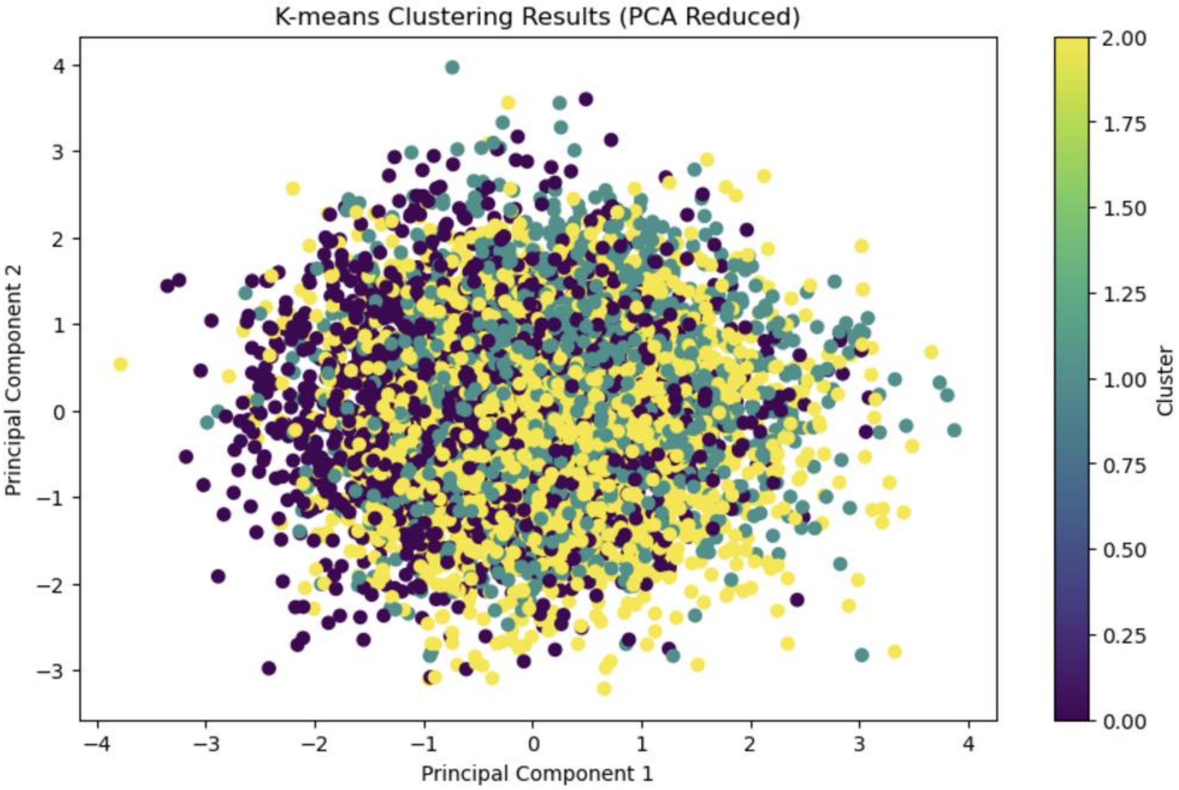
5. Results Summary

After executing the K-Means clustering algorithm, the following results were observed:

1. Clusters Identified: Three distinct clusters were formed, as visualized in the PCA-reduced plot.
2. Inertia: The inertia value of 114306.475 reflects the compactness of the clusters.
3. Cluster Visualization: The scatter plot showed well-separated clusters in the reduced dimensional space, indicating meaningful groupings.
4. Centroids: Each cluster's centroid was computed, summarizing the characteristics of its group members.



```
K-means Clustering Results:
Inertia (Sum of Squared Distances to Closest Cluster Center): 114306.475
```



5. Implications for Insights

The K-Means clustering results highlight inherent patterns within the data, which can inform domain-specific strategies. For instance:

- Understanding which features contribute most to the separation of clusters can guide targeted decision-making.
- Segmenting customers, regions, or processes can reveal opportunities for optimization or personalized interventions.

6. Conclusion

The K-Means Clustering algorithm successfully identified natural groupings within the dataset, as evidenced by the low inertia value and clear cluster visualization. The PCA-transformed plot effectively illustrated the separation of clusters, providing actionable insights for further analysis or domain-specific applications. This clustering technique is a valuable tool for exploratory data analysis, enabling stakeholders to uncover hidden patterns and drive data-driven decisions.

Report Conclusion and Recommendations

In conclusion, this study presents a detailed analysis of factors influencing student performance, based on data collected from schools across Nepal. By examining various socio-economic and educational variables, such as hours studied, parental involvement, access to resources, sleep hours, and the quality of teaching, we have identified key determinants that impact academic achievement. The results indicate that student performance is not only influenced by individual academic effort but is also deeply shaped by external factors such as family income, parental education, access to resources, and the quality of school infrastructure.

One of the most striking findings of this study is the significant relationship between the availability of educational resources, such as internet access and learning materials, and student performance. It has been shown that students with better access to resources, including computers and the internet, tend to perform better in exams. Moreover, parental involvement and educational support also play a critical role in enhancing students' motivation and academic success.

Given the insights drawn from the data analysis, the following recommendations are made to the Government of Nepal to improve the educational outcomes across the country and ensure that every student has an equal opportunity to succeed:

1. **Make Education Accessible to All**: Education is a fundamental right, and the government should make substantial efforts to ensure that it is accessible to every child, irrespective of their socio-economic background. This can be achieved by providing financial assistance to institutions, particularly in rural and underserved areas, and reducing the financial burden on students by lowering tuition fees. Scholarships, subsidies, and other financial aids should be expanded to ensure that students from lower-income families are not excluded from pursuing quality education.

2. **Enhance Educational Infrastructure for Students with Disabilities**: The government must prioritize making educational infrastructure more inclusive by providing facilities and resources tailored to students with disabilities. This includes accessible classrooms, learning aids, assistive technologies, and trained staff who can cater to the diverse needs of students with physical and learning disabilities. Ensuring that students with disabilities have equal access to education is critical to fostering an inclusive society.

3. **Expand Access to the Internet and Digital Resources**: As the digital divide continues to widen, the government should ensure that every public library, school, and educational institute is equipped with high-speed internet access. The availability of the internet is crucial for students to access online learning resources, participate in virtual classrooms, and gain exposure to the wider world of knowledge. Providing free internet in public educational institutions will bridge the digital divide and create a more equitable learning environment.

4. **Equip Schools with Modern Learning Materials**: To keep up with the evolving demands of education, the government should ensure that all schools are equipped with up-to-date educational materials, including textbooks, computers, and other learning tools. By investing in modern infrastructure, such as smart classrooms, interactive whiteboards, and digital resources, the government can help create an engaging and effective learning

environment. Schools should also be provided with up-to-date learning materials in different subjects, ensuring that students are not deprived of the resources they need to excel academically.

5. **Boost Student Motivation through Competitions and Rewards**: Student motivation plays a vital role in academic success. The government should encourage schools to organize inter-school competitions, such as debates, quizzes, and sports events, to foster healthy competition among students. Introducing a system of rewards and recognition for outstanding academic performance, creativity, and leadership can further incentivize students to perform better. Scholarships, certificates, and public recognition of achievements can go a long way in boosting self-esteem and motivation.

6. **Strengthen Parent-Teacher Engagement**: Parental involvement is a crucial factor in the academic success of students. Therefore, the government should encourage schools to regularly conduct workshops and seminars for parents, educating them on how to properly support their children's education. Parents should be trained on how to motivate their children, help with homework, and create a conducive learning environment at home. Regular parent-teacher meetings can also facilitate better communication between schools and families, ensuring that any academic issues are identified and addressed promptly.

7. **Professional Development for Teachers**: Teachers are the cornerstone of the education system, and their professional development is essential for improving the quality of education. The government should invest in ongoing training programs for educators to equip them with the latest teaching methodologies, technological tools, and strategies for managing diverse classrooms. Teachers should be encouraged to attend workshops, seminars, and educational conferences to enhance their teaching skills and stay updated on the latest trends in education.

8. **Promote Mental Health and Well-being**: The mental health and emotional well-being of students are integral to their academic success. Schools should be equipped with counselors who can support students dealing with stress, anxiety, and other mental health issues. Providing a safe and supportive environment is critical for students to thrive academically and personally. The government should encourage schools to incorporate mental health awareness into the curriculum and provide access to resources for students in need.

9. **Focus on Curriculum Development**: The curriculum should be periodically reviewed and updated to ensure that it is aligned with the needs of the 21st-century workforce. The government should collaborate with educational experts and institutions to develop a curriculum that fosters critical thinking, creativity, and problem-solving skills. A more flexible curriculum, with a focus on both academic and vocational training, will better prepare students for future challenges.

Final Thoughts:

The education system plays a pivotal role in shaping the future of the nation. By addressing the issues identified in this study and implementing the recommended changes, the Government of Nepal can ensure that every student, regardless of their background, has the opportunity to succeed. With increased investment in infrastructure, resources, teacher training, and parental involvement, the educational landscape of Nepal can be transformed, fostering an environment where students not only perform better academically but are also prepared to contribute meaningfully to society. By making these strategic investments in the education sector, Nepal can create a more equitable, accessible, and sustainable educational system for future generations.

Reflection:

Working collaboratively on the project analyzing factors influencing student performance in Nepalese schools has been an incredibly rewarding experience for our team. Engaging with a comprehensive dataset of 6,607 records across 20 variables allowed us to explore the intricate dynamics of educational outcomes in Nepal, and we collectively gained valuable insights into the challenges and opportunities within the education system.

What We Learned/Experienced?

We discovered the power of data in informing educational policies. By analyzing socio-economic factors, academic inputs, and behavioral aspects, we gained a understanding of what influences student performance. We learned that rigorous data preparation is crucial for accurate analysis, including handling missing data and encoding categorical variables.

What Was Easy?

Collaboratively discussing the findings from our machine learning models made it easier to draw meaningful conclusions.

What Was Hard?

Navigating the complexities of data preprocessing and model selection required significant teamwork and communication. Ensuring that our models were both effective and interpretable was a delicate balance.

What Was Fun?

Analyzing the relationships between various factors and exam scores was intellectually stimulating. We enjoyed uncovering hidden patterns in the data through our machine learning models.

Imagining Impact: Discussing the potential real-world implications of our findings was particularly exciting. The idea that our work could contribute to meaningful changes in educational policy motivated us throughout the project.

Will We Continue This Project?

Yes, we are all enthusiastic about continuing this project beyond its current scope. Including more schools and regions will enhance our analysis and provide a broader perspective on education across Nepal.

This project has strengthened our technical skills while deepening our understanding of how data analysis can inform educational policies.

References:

1. Factors Affecting Academic Achievement of B.Sc. Students of Tribhuvan University Constituent Campuses in Province - 1, Nepal.
2. Data Must Speak: Unpacking factors influencing school performance in Nepal
3. Factors Affecting Academic Performance of Students at Community Secondary School in Nepal