Student Performance Behaviour Data Analysis Report

1. Dataset Description

1.1 Source:Kaggle Student performance behaviour dataset (StudentPerformanceFactors.csv) – 6,600 records.

1.2 Columns:

1.Hours Studied: The number of hours a student studies, ranging from 1 to 44 hours, has a notable impact on their Exam_Score.

2.Attendance: Students with higher attendance rates tend to perform better. The data shows attendance percentages from 60% to 100%.

3.Previous Scores: A student's historical performance, measured by Previous_Scores (ranging from 50 to 100), is a key predictor of their current Exam_Score.

4.Parental Involvement : The level of parental involvement is categorized as Low, Medium, or High.

5.Access to Resources: Access to resources is also categorized as Low, Medium, or High, and appears to be a significant factor.

6.Extracurricular Activities: Whether a student participates in extracurricular activities or not (Yes or No) is also a factor.

7.Sleep Hours: The number of hours a student sleeps, ranging from 4 to 10 hours, is a variable in the dataset.

8.Motivation Level : Motivation is categorized as Low, Medium, or High.

9.Internet Access : The availability of internet access (Yes or No) is also considered.

10. Tutoring Sessions : The number of tutoring sessions a student has is recorded, from 0 to 6.

11. Family Income : Family income is categorized as Low, Medium, or High.

12. Teacher Quality : Teacher quality is categorized as Low, Medium, or High.

13. School Type : Students attend either a Private or Public school.

14. Peer Influence : Peer influence is classified as Positive, Neutral, or Negative.

15. Physical Activity : The number of physical activity hours is recorded, from 0 to 6.

16. Learning Disabilities : Whether a student has a learning disability or not (Yes or No).

17. Parental Education Level : The highest level of education attained by a parent is noted, from High School to Postgraduate.

18. Distance from Home : The distance from home to school is categorized as Near, Moderate, or Far.

19. Gender : Student gender is recorded as Male or Female.

Data Insights and Trends

The Exam_Score varies across all the factors listed above. For example, some high scorers studied many hours and had high parental involvement, while others with lower scores had less study time and low parental involvement. The wide range of factors and scores suggests a complex relationship that could be further analyzed to pinpoint specific correlations. For example, the highest score of 101 was achieved by a female student with a High Parental Education Level, High Parental Involvement, and High Teacher Quality. However, a student with a score of 55 had Low Parental Involvement, Low Access to Resources, and a Low Parental Education Level.

1.3 Data Quality:

The operations performed indicate a thorough approach to data quality, focusing on understanding the dataset's structure, content, and potential issues.

Schema and Structure Inspection:

The code starts by printing the schema of the DataFrame, which is a crucial first step in data quality. It shows the column names and their inferred data types, such as `double`, `long`, `string`, and `boolean`. This helps in identifying if the data was loaded correctly and if any data type conversions are needed later.

Missing Value Analysis:

The notebook explicitly checks for missing or null values in each column. It counts the number of nulls for all columns, showing that `Parental_Involvement`, `Teacher_Quality`, and `Access_to_Resources` have nulls. The code also displays the rows where these null values exist, allowing for a detailed inspection of the affected records.

Column Uniqueness Check:

A check for unique values in the `Exam_Score` column is performed, showing that there are 47 unique scores out of 1000 total records. This helps to understand the distribution and variety of the target variable.

Statistical Summary:

The `describe` function is used to generate a statistical summary of the DataFrame. This provides key metrics like `count`, `mean`, `stddev`, `min`, and `max` for all columns. This operation is vital for identifying outliers, understanding data distribution, and ensuring the data is within expected ranges.

2.Operations Performed

The operations performed demonstrate a clear workflow for data preparation and feature engineering using PySpark.

- **Categorical Data Conversion**: The code addresses categorical variables by converting them into a numerical format suitable for machine learning models.
  - **Label Encoding**: The `StringIndexer` is used to convert categorical string columns (e.g., 'Gender', 'School_Type') into numerical indices. The code shows the transformation of columns like `Gender`, `School_Type`, and `Learning_Disabilities`.
  - **One-Hot Encoding**: The `OneHotEncoder` is then applied to these indexed columns to create binary vector columns. This is a best practice for categorical data to avoid ordinal assumptions in the model. The code creates new columns like `Gender_Vector`, `School_Type_Vector`, and so on.
- **Feature Assembly**: The `VectorAssembler` is used to combine all the processed feature columns into a single vector column, which is the required input format for most machine learning algorithms in PySpark. The code clearly lists all the input columns (`features_col`) and the output column (`assembled_features`).
- **Pipeline Creation**: A `Pipeline` is created to streamline the entire process of data transformation. This combines the `StringIndexer`, `OneHotEncoder`, and `VectorAssembler` steps into a single workflow that can be easily applied to the dataset. The pipeline is then fit to the data, and a new DataFrame with the assembled feature vector is created.
- **Data Splitting**: The final processed data is split into training and testing sets, which is a standard procedure for building and evaluating a machine learning model. The code uses a 70/30 split, with 70% of the data allocated for training the model and 30% for testing its performance.
- **Model Training**: The processed data is used to train a linear regression model. The `LinearRegression` model is imported, and its `fit` method is called with the training data. This suggests that the ultimate goal of the project is to build a predictive model for the `Exam_Score`.

3. Key Insights

Based on the provided data, a few key insights emerge regarding student performance:

Strongest Predictors :

Unsurprisingly, `Previous_Scores` is a strong indicator of a student's `Exam_Score`. However, other factors like `Hours_Studied`, `Attendance`, and `Parental_Involvement` also have a clear and measurable impact.

Socioeconomic Influence :

The data suggests that external factors play a role. Students from families with `High` income and `High` access to resources generally tend to perform better. This highlights the link between a supportive home environment and academic success.

Holistic View of Performance :

The analysis moves beyond simple academic metrics. By incorporating factors like `Sleep_Hours`, `Motivation_Level`, and `Physical_Activity`, the model acknowledges that a student's performance is a result of their overall well-being, not just their study habits.

Recommendations

To improve student performance, the following recommendations can be made based on the project's analysis:

Implement a Proactive Support System :

Use the predictive model to identify students who are at risk of a low `Exam_Score`. The model can flag students with low `Previous_Scores`, low `Attendance`, or low `Parental_Involvement` so they can receive timely academic and emotional support. This moves the focus from reactive measures to proactive intervention.

Promote Healthy Habits :

Encourage students to maintain a healthy lifestyle. The data shows that factors like `Sleep_Hours` and `Physical_Activity` are part of the predictive model. School programs could be developed to educate students on the importance of these habits for academic performance.

Bridge the Resource Gap :

Use the insights on `Family_Income` and `Access_to_Resources` to address inequities. The school could provide additional support or resources to students from low-income families to ensure they have the same opportunities as their peers. This could involve providing access to tutoring sessions, digital resources, or study materials.

Enhance Parent-Teacher Collaboration :

The `Parental_Involvement` metric suggests that parental engagement is beneficial. A new report could be created to show parents how their involvement is linked to their child's success. This could encourage them to take a more active role in their children's education.

Utilize the Predictive Model for Personalization :

The linear regression model provides a framework for understanding the weight of each factor. A future project could create a user-friendly dashboard for teachers and counselors. This dashboard could show an individual student's predicted score and highlight the specific factors (e.g., `Hours_Studied`, `Attendance`) that, if improved, could lead to a better outcome. This would enable a personalized approach to student development.
