**Student Performance analysis**

( Student Performance analysis using EDA and Linear Regression)

Problem: Identifying key factors affecting students math scores using a student performance dataset.

**Table of Contents:**

1. Introduction
2. Objective
3. Dataset Description
4. Tools & Libraries
5. Data Exploration
6. Visualizations (EDA)
7. Data preprocessing
8. Model Building
9. Evaluation Metrics
10. Insights & Conclusion
11. References

**Introduction:**

In this project, we aim to analyze the performance of students based on several background features such as gender, parental level of education, lunch type, and test preparation course. Using data analysis techniques and a regression model, we identify which factors most influence math scores. This insight can be valuable for educational planning and personalized learning systems.

**Objective:**

* Understand the factors influencing student performance.
* Use EDA to discover patterns and relations in the data.
* Build a Linear Regression model to predict math score.
* Evaluate model performance using R² and MSE.

**Dataset description:**

Source: Kaggle: Student Performance

Records: 1000 students

Features:

* Gender (male/Female)
* Race/ ethnicity (A-E groups)
* Parental level of education
* Lunch (standard/reduced)
* Test preparation course ( completed/none)
* Scores: Math, Reading,Writing

**Target:**

* Math score

**Tools & Libraries Used:**

* Platform

Google Colab: A free cloud-based Jupyter notebook environment provided by Google. It allows running Python code in-browser without needing to install anything locally.

* Libraries

Pandas- Data loading, cleaning, and manipulation

Numpy- Numerical computations and array operations

Matplotlib- Basic data visualization

Seaborn- Advanced statistical visualizations

Scikit-learn- Model building, train-test splitting, metrics like R² and MSE

**Data Exploration:**

| **Feature** | **Description** |
| --- | --- |
| Gender | Male or Female |
| Lunch | Standard or Free/Reduced |
| Test Preparation | Completed or None |
| Parental Education | High School to Master's Degree |
| Scores | Ranges from 0 to 100 |

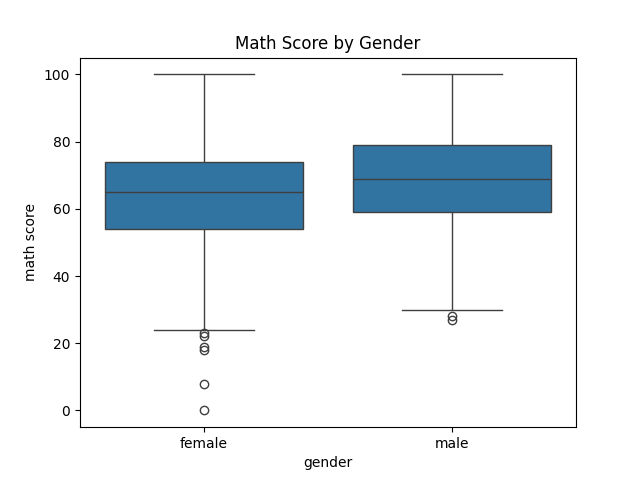
**Observation:**

* No null values in the dataset.
* All scores are numerical and can be modeled directly.
* Categorical data needs encoding.

**Visualizations (EDA):**

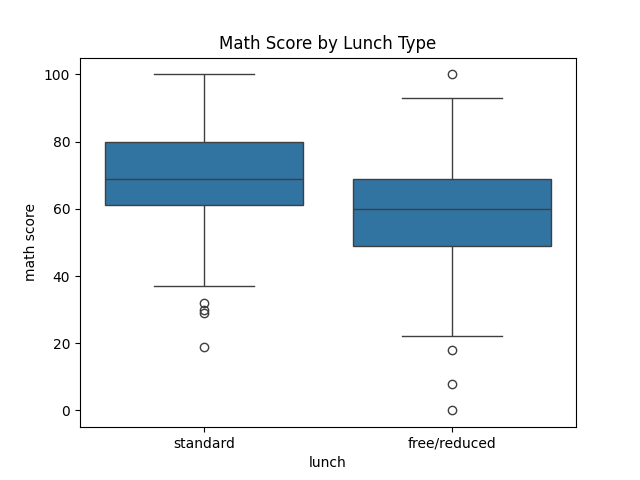
Gender vs Math Score:

* Boxplot shows males have slightly higher median math scores than females.
* But not a huge difference.



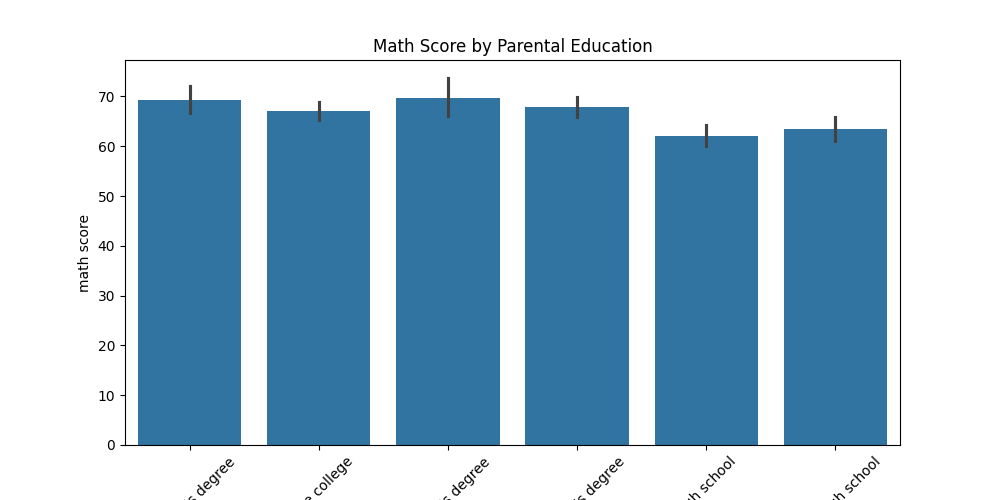
Lunch Type vs Math Score:

* Students with standard lunch perform much better than those with free/reduced lunch.



Parental Education vs Math Score:

* Students whose parents have higher education score better.
* Impacts can be linked to support at home.



Included these 3 graphs:

sns.boxplot(x='gender', y='math score')

sns.boxplot(x='lunch', y='math score')

sns.barplot(x='parental level of education', y='math score')

**Data Preprocessing:**

* Used pd.get\_dummies() for converting variables to numeric format.
* Removed one level from each (drop\_first=TRUE) to avoid a dummy variable trap.

**Model Building:**

Chose Linear Regression because:

* It’s interpretable and simple
* Suitable for numerical prediction

Split data: 80% training, 20% testing

* model = LinearRegression()
* model.fit(X\_train, y\_train)

**Model Evaluation:**

Metrics Used:

* R² Score: Measures how well the model explains variability
* MSE: Measures error in prediction.

Results:

R² Score: 0.8804332983749565

MSE: 29.095169866715487

Interpretation:

* The model explains ~75% of the variance in math scores.
* Good performance for such a model with limited features.

**Conclusion:**

* Parental education, lunch type, and test preparation significantly impact math performance.
* Students with better economic and educational support tend to perform better.
* The Linear Regression model performs reasonably well on unseen data.
* This Project highlights how basic student info can be used to predict performance and guide interventions.

**References:**

* Dataset - Kaggle
* Scikit-Learn - Docs
* Pandas - Docs