Dataset:

https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression?utm_source=chatgpt.com

Task 1 - Load and Inspect the Dataset

1. Load the CSV file into a pandas DataFrame
2. Print:
   - Shape of the dataset
   - Column names
   - First 5 rows
   - Describe
   - Info
3. Check data types of all columns

Task 2 - Missing Value Analysis

1. Check the number of missing values per column
2. Decide how missing values should be handled:
   - Numerical features → mean or median
   - Categorical features → most frequent value

Task 3 - Exploratory Data Analysis (EDA)

Goal: Understand the data distribution, relationships, and potential issues.

Target Variable Analysis:

1. Plot the distribution of Performance Index
2. Check whether the distribution is:
   - Normal
   - Skewed
3. Identify possible outliers

Feature Relationships:

1. Compute the correlation matrix for numerical features
2. Visualize correlations using a heatmap
3. Identify the top features most correlated with the target variable

Feature-Level Analysis:

1. Analyze how study hours, sleep hours, and previous scores affect performance
2. Compare performance for students with and without extracurricular activities

----

Task 4 - Data Preprocessing

Goal: Prepare the data correctly for machine learning models.

Feature and Target Separation:

Separate the dataset into:

- Features (X)
- Target (y)

Feature Type Separation:

Identify:

- Numerical features
- Categorical features

Preprocessing Pipelines:

1. Create a **numerical pipeline** that:
   - Imputes missing values
   - Applies feature scaling
2. Create a **categorical pipeline** that:
   - Imputes missing values
   - Applies one-hot encoding
3. Combine both pipelines using ColumnTransformer

Task 5 - Train-Test Split

1. Split the dataset into training and testing sets
2. Use:
   - 80% training data
   - 20% testing data
3. Set random_state = 42

Task 6 - Linear Regression Model

1. Create a pipeline that includes:
   - Preprocessing
   - Linear Regression model
2. Train the model on the training set
3. Predict the target values for the test set

Task 7 - Model Evaluation

Evaluate the model using:

1. Mean Absolute Error (MAE)
2. Root Mean Squared Error (RMSE)
3. R^2 Score

Analyze whether the model:

- Fits well
- Underfits
- Overfits

Task 8 - Regularization

Train and compare the following models:

1. Ridge Regression (L2)

2. Lasso Regression (L1)
3. ElasticNet

Analyze:

- Effect on coefficients
- Effect on overfitting

Task 9 - Cross-Validation

1. Apply k-fold cross-validation (k = 5)
2. Evaluate model stability using RMSE
3. Compare cross-validation results with test-set results

Task 10 - Learning Curves

1. Plot learning curves for:
    - Training error
    - Validation error
2. Analyze:
    - Bias vs variance
    - Whether more data would help

Task 11 - Gradient Descent with SGDRegressor

Goal: Understand how Gradient Descent works in regression.

Feature Scaling:

1. Apply feature scaling to numerical features
2. Explain why scaling is important for Gradient Descent

Train SGDRegressor:

1. Train an SGDRegressor model
2. Experiment with different learning rates

3. Observe convergence behavior

Learning Rate Experiment:

1. Train multiple models with different learning rates
2. Compare their RMSE values
3. Identify:
    - Too small learning rate
    - Too large learning rate
    - Optimal learning rate

Task 12 - Hyperparameter Optimization

Grid Search:

1. Perform GridSearchCV for SGDRegressor
2. Tune:
    - Learning rate
    - Regularization strength
    - Penalty type

Randomized Search:

1. Perform RandomizedSearchCV
2. Compare results with Grid Search

Which model performed best ?