**TASK 4:**

**Data Science Lifecycle Example:**

**1. Problem Definition**

* **Objective**: The goal is to predict the variable Y (target) based on the feature X (input).

**2. Data Collection**

* In this case, the data is synthetically generated, but in a real-world scenario, we would collect data from a variety of sources such as sensors, online databases, business records, or APIs.
* **Example**: Data could be collected about house prices where X could represent house features like size, and Y could represent the price.

**3. Data Preparation**

* **Cleaning**: We checked for missing values and handled any missing or erroneous data (in this case, we simulated clean data).
* **Feature Engineering**: This stage would involve creating new variables or transforming existing ones. For example, in a housing dataset, X might include derived features like price per square foot.
* **Normalization/Scaling**: If necessary, the feature data could be scaled (e.g., using min-max scaling) to improve model performance.

**4. Exploratory Data Analysis (EDA)**

* **Understand the Data**: Visualize the data to understand the relationship between X and Y. In this case, a scatter plot between X and Y might be helpful to see if there's a linear trend.
* **Identify Patterns**: This stage involves using summary statistics and visualizations to understand the distribution, correlations, and outliers in the data.

**5. Modeling**

* **Algorithm Selection**: We selected the **Linear Regression** model, which is suitable for predicting a continuous target variable based on one or more input features.
* **Training**: We split the data into training and testing datasets. We used the training set to teach the model the relationship between X and Y.
* **Hyperparameter Tuning**: In more advanced scenarios, you would experiment with different hyperparameters (e.g., regularization terms in regression) to improve model performance.

**6. Evaluation**

* **Model Testing**: After training the model, we used the test set to evaluate its performance. In our case, we calculated the **Mean Squared Error (MSE)** as a measure of prediction accuracy.
* **Metrics**: Depending on the problem, you might also use metrics like **R²** for regression tasks, or **precision**, **recall**, and **accuracy** for classification tasks.

**7. Deployment**

* **Integration into Production**: After the model is validated, it would be deployed in a real-world system. For example, this regression model could be used to predict house prices for new listings based on their features.
* **Deployment Technologies**: You might deploy the model using a web API, integrate it into a software application, or use cloud-based services like AWS, Azure, or Google Cloud for scalability.
* **Model Monitoring**: After deployment, the model's performance should be monitored over time. If it starts to degrade, you may need to retrain the model with updated data.

**8. Model Maintenance**

* **Updates**: As new data comes in (e.g., more real estate listings or sales data), the model might need to be retrained to maintain accuracy.
* **Feedback Loops**: Continuous monitoring and feedback loops ensure that the model adapts to new trends and maintains its predictive power.