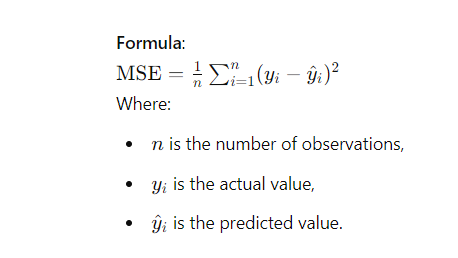
**Mean Squared Error (MSE)**

**Definition**:

* Mean Squared Error (MSE) is a widely used metric for evaluating the accuracy of a regression model. It represents the average of the squared differences between the predicted and actual values.

****

**Interpretation**:

* MSE measures the average squared error between the predicted and actual values.
* A lower MSE indicates a better fit of the model to the data.
* Since errors are squared, MSE heavily penalizes larger errors more than smaller ones.

**Advantages**:

* Emphasizes larger errors, making it useful when large deviations are particularly undesirable.
* Differentiable, making it suitable for optimization algorithms like gradient descent.
* Provides a useful loss function for many machine learning models, particularly in linear regression.

**Disadvantages**:

* Sensitive to outliers because errors are squared, leading to higher penalties for larger deviations.
* Less interpretable compared to metrics like MAE, since it does not represent the error in the same units as the original data.

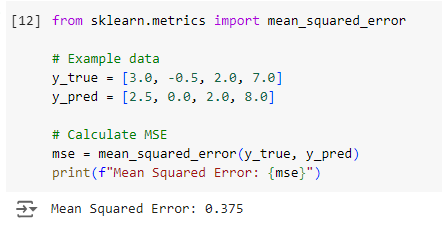
**Use Cases**:

* MSE is commonly used in regression problems where the goal is to minimize the error, especially when large errors are a concern.
* It is often employed in training and evaluating models like linear regression, decision trees, and neural networks.

**Comparison with Other Metrics**:

* **Mean Absolute Error (MAE)**: Unlike MAE, which takes the absolute value of errors, MSE squares the errors. MSE penalizes larger errors more heavily, making it more sensitive to outliers than MAE.
* **Root Mean Squared Error (RMSE)**: RMSE is the square root of MSE, which puts the error metric back in the same unit as the original data. RMSE is more interpretable than MSE but still penalizes larger errors more.
* **R-squared (R^2)**: R^2 measures how well the model explains the variance in the data, whereas MSE provides an absolute measure of error magnitude.

**Python Implementation Example**:



**When to Use MSE**:

* When large errors are particularly undesirable and should be penalized more than smaller ones.
* In applications where a differentiable loss function is needed for model training, such as in linear regression or neural networks.
* When comparing models, especially in cases where the magnitude of the error is important.