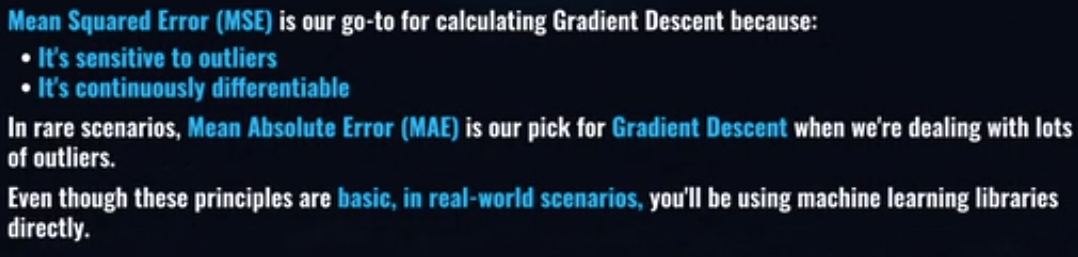
|  |  |
| --- | --- |
| **Mean Squared Error (MSE)** | **Mean Absolute Error (MAE)** |
| Smooth and differentiable, suitable for gradient-based methods. | Non-differentiable or derivative is undefined at x= zero (has a "kink"), Can’t update m and b |
| Penalizes large errors more heavily due to squaring the error term | Treats all errors linearly, with no disproportionate penalty for large errors |
| Easier to optimize with gradient-based methods (e.g., Gradient Descent). Derivatives help us to find gradient at any point. | More difficult to optimize due to non-differentiability at zero |
| Sensitive to outliers, as large errors have a bigger impact | Less sensitive to outliers, treats all errors equally |
| Strong connection to Maximum Likelihood Estimation if errors are normally distributed | No clear connection to statistical theory (not linked to likelihood maximization) |
| Minimizes the **mean** of the squared errors, leading to a model that focuses on reducing larger errors | Minimizes the **median** of the errors, providing a more robust solution to outliers |
| Generally computationally efficient, especially with smooth gradient-based methods | Computationally feasible, but less efficient for gradient-based optimization |
| Commonly used when minimizing larger errors is more important and smooth optimization is needed | Used when robustness to outliers is required or when errors are expected to be uniformly distributed |

****

**Hyperparameters**

Hyperparameters are not learned from data, they are manually set before training begins. They control the learning process and model structure.

**Key Examples of Hyperparameters:**

* **Learning Rate**: Determines how much the model's weights are adjusted with each update.
* **Number of Epochs**: The number of times the model is trained on the entire dataset.
* **Batch Size**: The number of training samples used in one iteration.
* **Tree Depth (for decision trees)**: Limits how deep decision trees can grow.
* **Regularization (L1, L2)**: Prevents overfitting by penalizing large weights.

**Hyperparameter Tuning:**

Hyperparameters are optimized using methods like:

* **Grid Search**: Testing all possible combinations of hyperparameters.
* **Random Search**: Randomly selecting combinations to test.
* **Bayesian Optimization**: A probabilistic method for selecting the best set of hyperparameters.

In summary, hyperparameters significantly influence a model's performance and must be carefully selected or tuned for optimal results.