**Simple linear Regression:-**

Using a straight line, a simple linear regression model estimates the relationship between one independent variable and one dependent variable. Both variables should be quantitative.

Simple linear regression is to find the distance between the minimum distance between the variables and the straight fit line.

The aim of Simple Linear regression is to find the best-fit line.

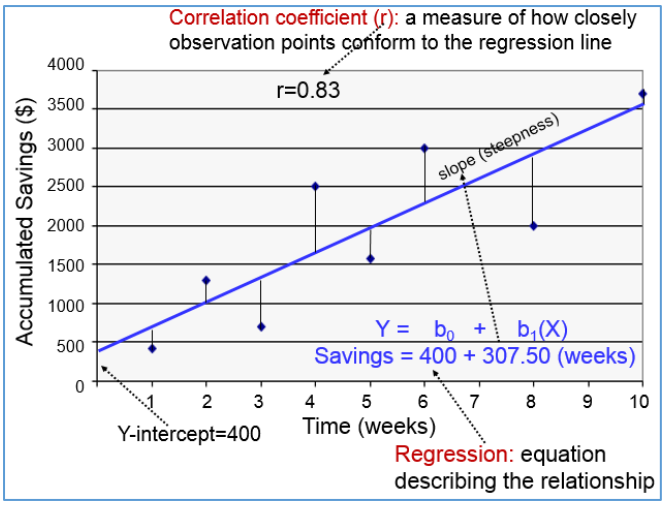
Technical Term:

Error/Residual: The difference between a real point and a projected point is called Error/Residual.

| Equation of the straight line | |
|---|---|
| $y = m x + c$ | Or |
| $y = \beta_0 + \beta_1 X_1$ | Or |
| $h_\theta(x) = \theta_0 + \theta_1 X_1$ | Or |

$\theta_0$ or $\beta_0$ OR **c** = Intercept (where the best-fit line meets the Y axis).
$\theta_1$ or $\beta_1$ or **m** (Slope or Coefficient) = if there is a unit movement on the x-axis then, what the movement on the y-axis is called slope.
$h_\theta(x)$ or **y** = is the predicted point.
$X_1$ or **x** = is the actual point.

To find the best-fit line we should be changing the intercept $\beta_0$ and Coefficient $\beta_1$ Values.




**Cost functions.**

A cost function, also known as a loss function or objective function, is a mathematical measure that evaluates the **discrepancy(error)** between predicted points and the true values or labels in the training data and serves as the basis for optimizing the model during training. The **objective** is to **minimize the Error** by adjusting the model's **parameters(intercept $\beta_0$ and Coefficient $\beta_1$)**, enabling the model to make more accurate predictions and improve its overall performance on the given task.

Here we are going to study three types of cost functions as below used in linear regression:

1) **MSE (Meas Square Error)**
2) **MAE (Mean Absolute Error)**
3) **RMSE (Root Means Square Error)**

**Mean Squared Error:**

Mean Squared Error:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Predicted Data Point
Actual Data Point
Total number of Data point(m)

MSE works by calculating the squared differences between each predicted value and the corresponding actual value, summing them up, and then dividing them by the total number of data points. By squaring the differences, negative and positive errors are both considered, and larger errors are penalized more heavily, giving more weight to larger deviations between predictions and actual values.

MSE is particularly popular due to its mathematical properties, including convexity, which makes it easier to optimize when training machine learning models. However, it is sensitive to outliers and may not be an ideal metric in cases where the data is heavily skewed or contains heteroscedasticity (varying levels of error across the range of the target variable).

A lower value of MSE indicates that the model's predictions are closer to the actual values, and thus, a lower MSE is desirable as it signifies better model performance.

| Advantages | Disadvantages |
|---|---|
| - This equation is differentiable (differentiable is required to calculate slope, to update the $\theta_0$ and $\theta_1$ )<br>- This equation creates a convex function.<br>- Since this function creates a convex function, it has one global minimum | - Since residuals are penalized by squaring it up, it is not robust to Outliers.<br>- No Unit of Measurement: MSE is expressed in the square of the units of the target variable, which can make it difficult to relate to the original data units. |

**Mean Absolute Error:**

Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

Predicted Data Point
Actual Data Point
Total number of Data point(n)

MAE works by taking the absolute difference between each predicted value and the corresponding actual value, summing them up, and then dividing by the total number of data points. MAE does not square the differences between predicted and actual values. This means that MAE treats positive and negative errors equally and does not penalize larger errors more heavily.

MAE is particularly useful in scenarios where outliers may have a significant impact on the model's performance. Since it takes the absolute value of the errors, extreme deviations between predictions and actual values do not have an inflated effect on the overall MAE value.

This function uses the subgradient concept to calculate the derivative.

| Advantages | Disadvantages |
|---|---|
| - Robust to outliers<br>- Remains in the same unit, since residuals are not penalized by squaring it. | - Conversion takes usually takes a lot of time since the same unit is used.<br>- Optimization is usually a complex task |

## Root Mean Squared Error

Root Mean Squared Error:

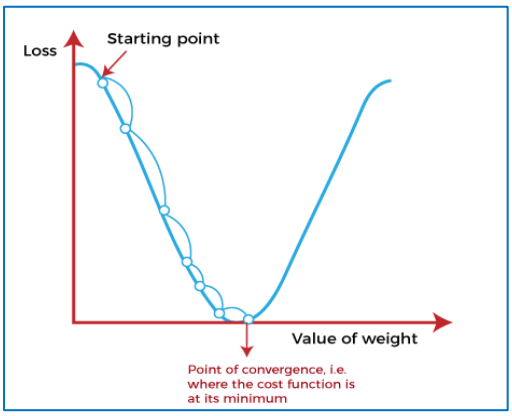
$$RMSE = \sqrt{MSE}$$

RMSE is similar to Mean Squared Error (MSE), but RMSE is taken as the square root of MSE to make the units of the error metric consistent with the original target variable. By taking the square root, RMSE is in the same unit as the target variable, making it more interpretable and easier to compare with the scale of the original data.

It penalizes larger errors more heavily, giving a higher weight to larger deviations between predictions and actual values.

| Advantages | Disadvantages |
|---|---|
| - Sensitivity to Large Errors: RMSE gives higher weight to larger errors due to squaring the differences between predicted and actual values. This sensitivity makes it suitable for applications where large errors are more critical and need to be penalized appropriately.<br>- RMSE is widely used and easy to understand. The metric is expressed in the same units as the target variable, which makes it more interpretable and relatable to the problem domain. | - Since residuals are penalized by squaring it up, it is not robust to Outliers. |

**Gradient descent:**




**Convergence Algorythem:**

$$\theta_j = \theta_j - a \frac{\partial J(\theta_j)}{\partial \theta_j}$$

**For –ve Slope**

$\theta_j = \theta_j - a$ (-ve)

**For +ve Slope**

$\theta_j = \theta_j - a$ (+ve)

**Performance Metrix:** Performance Metrix is used to know the accuracy of the model and they tell you if you're making progress or not. Here we will discuss two types of performance metrics.

1) **R Square**
2) **Adjusted R Square**

**R Square**

R-squared, also known as the coefficient of determination, is a performance metric used to evaluate the goodness of fit of a linear regression model. It provides an indication of how well the model explains the variation in the dependent variable based on the independent variables.

R-squared is a value between 0 and 1. A value of 0 indicates that the model does not explain any of the variability in the dependent variable, while a value of 1 indicates that the model perfectly explains all the variability.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

$R^2$ = coefficient of determination
$RSS$ = sum of squares of residuals
$TSS$ = total sum of squares

R Squared = Small Number / Bigger Number — Then the the output will be in +ve

R Squared = Bigger Number / Small Number — Then the output will be in -ve, and also consider that the worst model is created

**Adjusted R Square:**

There may be instances where certain features within a model are not correlated with the output feature. Logically, these features should decrease the accuracy of the model, but in reality, they actually improve the accuracy. This can lead to an unclear understanding of the model's performance and can result in poor real-time performance if deployed. To avoid such scenarios, the "adjusted R squared" formula is used. This ensures that the accuracy of the model decrease when non-correlated features are added and these features are removed during the training phase.

Adjusted R-squared is a modified version of the R-squared (coefficient of determination) that takes into account the number of independent variables in the model. It is used to assess the goodness of fit of a regression model while penalizing the inclusion of unnecessary independent variables.

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

Where
$R^2$ Sample R-Squared
$N$ Total Sample Size
$p$ Number of independent variable

**Overfilling and Underfitting (Bias and Variance):**

**Bias:**
accuracy of the training data is mentioned by Bias.
**Low Bias :**
if Training Data has Very Good Accuracy then it is Low Bias.
**High Bias :**
If Training Data has Bad Accuracy then.

**Variance:**
accuracy of the test data is mentioned by Variance.
**Low Variance:**
if Testing Data Very Good Accuracy.
**High Variance:**
if Testing Data Bad or Good Accuracy.

**Overfitting:**
if Training Data has Very Good Accuracy(Low Bias) while Testing Data has Bad Accuracy (High Variance) then this condition is called overfitting. To solve such a problem need to perform hyperparameter tuning.

**Underfitting :**
if Training Data has Very Bad Accuracy(High Bias), while Testing Data has Bad or Good Accuracy (High Variance) or Good Accuracy (Low Variance) then this condition is called Underfitting.

**Ridge Regression:(L2 Regularization)**

The main purpose of the Ridge is to reduce **overfitting**.

Useful for dealing with **multicollinearity** and when you want to reduce the impact of all features without eliminating it entirely.

**Coefficients are shrunk towards zero** but rarely become exactly zero, allowing all features to remain in the model.

In ridge regression, the regularized loss function is defined by adding a penalty term based on the **squared values** of the model's coefficients (weights). The penalty term is controlled by a hyperparameter λ (lambda). The formula for the regularized loss function in ridge regression is as follows

L2 regularization discourages large parameter values and promotes smoother parameter estimates. It does not lead to sparse solutions like L1 regularization, as it **does not drive coefficients to exactly zero**. Instead, it shrinks the values of the coefficients towards zero, but they remain non-zero.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{m} (\text{Slope})^2$$

**Lasso Regression: (L1 Regularization)**

The main aim of the Lasso is to reduce the **number of features**, basically, it is used to do the **feature selection**.

Some coefficients are driven to exactly zero, performing feature selection and keeping only the most important features.

Useful for feature selection when you suspect some features are irrelevant or redundant, and you want a more interpretable model.

L1 regularization adds the absolute values of the model's parameters as the penalty term to the loss function. The penalty term is controlled by a hyperparameter λ (lambda), which determines the strength of the regularization. The L1 regularization term is proportional to the sum of the absolute values of the model's coefficients.

L1 regularization encourages sparsity in the model, meaning that it drives some coefficients to exactly zero. This results in a simpler model with fewer features, as some features are effectively ignored in the final model. L1 regularization is particularly useful for feature selection, as it automatically identifies and selects the most important features in high-dimensional datasets.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{m} |\text{Slope}|$$

**ElasticNet Regression:**

Elastic Net regression is an extension of linear regression that combines both L1 (lasso) and L2 (ridge) regularization techniques.

It aims to overcome some limitations of ridge regression and lasso regression by providing a balance between feature selection and coefficient shrinkage.

Elastic Net strikes a balance between the two techniques and is useful for high-dimensional datasets with correlated features, offering flexibility in handling multicollinearity while performing feature selection.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{m} (\text{Slope})^2 + \lambda \sum_{i=1}^{m} |\text{Slope}|$$