Sanjivani Rural Education Society's

Sanjivani College of Engineering, Kopargaon-423603

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

NAAC 'A' Grade Accredited

Department of Information Technology

NBA Accredited-UG Programme

Machine Learning

Ms. K. D. Patil

Assistant Professor

Contents - Regression

- Linear Regression, Logistic Regression, Ridge Regression, Lasso Regression, Polynomial Regression, Types of Regression, Performance metrics, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2 (R-Squared)

Course Outcome

- **CO2:** To apply the Regression methods.

Linear Regression

- **Simple Linear Regression**
  - For simple linear regression, the form of the model is-

$$Y = A + BX$$

Here,

Y is the dependent variable that lies along the y-axis,

A is the y-intercept,

B is the slope of regression line,

X is the independent variable that lies along the x-axis,

Linear Regression

- **Simple Linear Regression**
  - For simple linear regression, the form of the model is-

$$Y = A + BX$$

$$A = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$B = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Linear Regression

- **Simple Linear Regression**
  - Let us consider, if

$$Y = \beta_0 + \beta_1 X$$

Where, Y is the dependent variable that lies along the y-axis

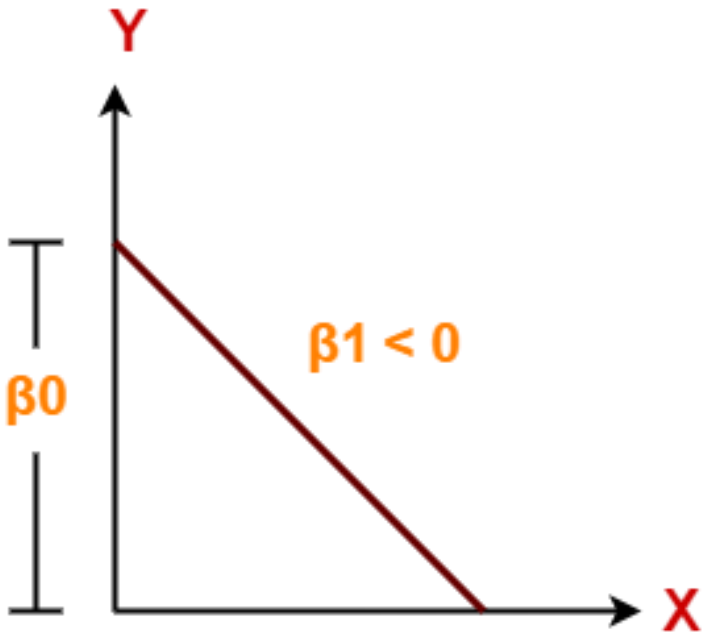
β0 is the y-intercept

β1 is the slope of regression line

X is the independent variable that lies along the x-axis

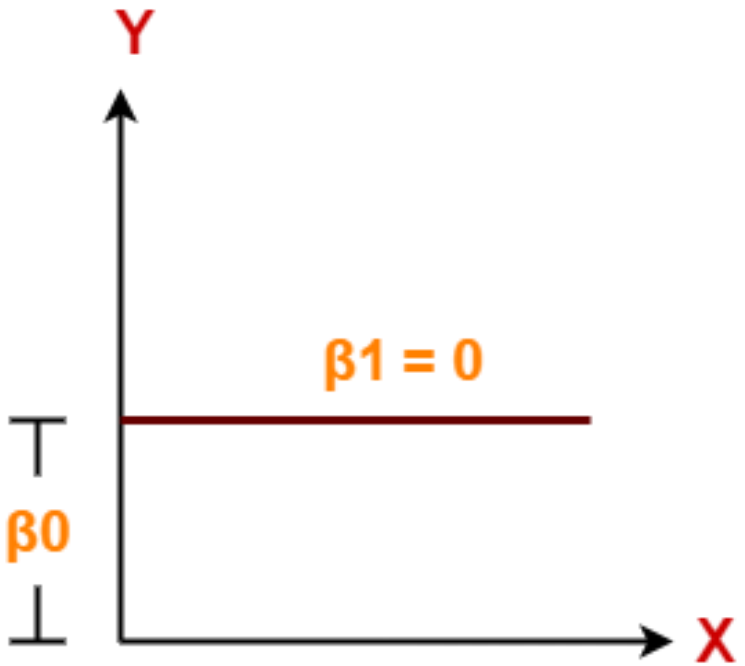
Linear Regression

- **Simple Linear Regression :** There are following 3 cases possible in
  - **Case-01: β1 < 0 and Y = β0 + β1X**
    - It indicates that variable X has negative impact on Y.
    - If X increases, Y will decrease and vice-versa.



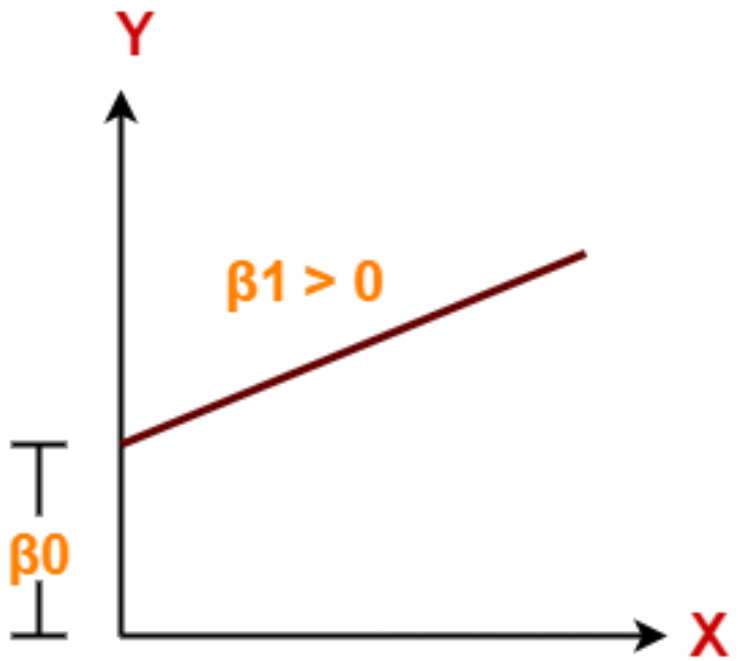
Linear Regression

- **Simple Linear Regression :** There are following 3 cases possible in
  - **Case-01: β1 = 0 and Y = β0 + β1X**
    - It indicates that variable X has no impact on Y.
    - If X changes, there will be no change in Y.



Linear Regression

- **Simple Linear Regression :** There are following 3 cases possible in
  - **Case-01: β1 > 0 and Y = β0 + β1X**
    - It indicates that variable X has positive impact on Y.
    - If X increases, Y will increase and vice-versa.



Linear Regression

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pandas as pd

cgpa = [6.89, 5.12, 7.82, 7.42, 6.94, 7.89, 6.73, 6.75, 6.09]
package = [3.26, 1.98, 3.25, 3.67, 3.57, 2.99, 2.6, 2.48, 2.31]
df = pd.DataFrame({'cgpa' : cgpa, 'package' : package})
y = df['package']
X = df.drop('package', axis = 1)

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)

lr = LinearRegression()
lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)
print(y_pred)
```

Evaluation metrics for a Linear Regression

- Machine learning model cannot have 100 % efficient otherwise the model is known as a biased model.
- Which further includes the concept of over-fitting and under-fitting.
- It is necessary to obtain the accuracy on training data, But it is also important to get a genuine and approximate result on unseen data otherwise Model is of no use.
- So to build and deploy a generalized model we require to Evaluate the model on different metrics which helps us to better optimize the performance, fine-tune it, and obtain a better result.

Evaluation metrics for a Linear Regression

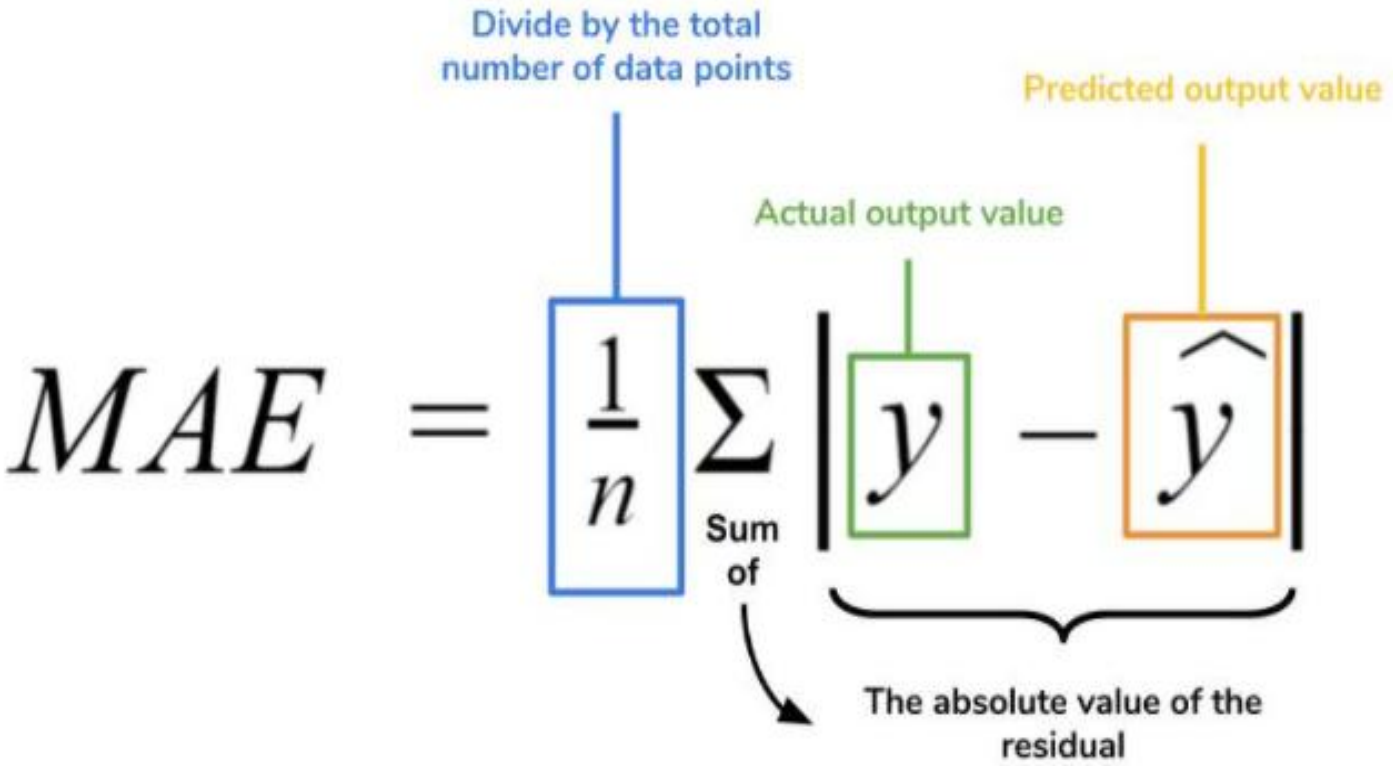
- Common Regression metrics are of Four Types:
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - R-squared (R^2) Score
  - Root Mean Squared Error (RMSE)

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**
  - MAE is a very simple metric which calculates the absolute difference between actual and predicted values.
  - It is basically a mistake made by the model known as an error.
  - Find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset.
  - So, Sum all the errors and divide them by a total number of observations and this is MAE.

Evaluation metrics for a Linear Regression

- **Mean Absolute Error(MAE):**
  - We aim to get a minimum MAE because this is a loss.



Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**
  - Not preferred in cases where outliers are prominent (easily noticable).
  - **MAE does not penalize large errors.**
  - This means that a large error contributes the **same amount** to the **overall MAE as a small error**, unlike Mean Squared Error **(MSE)**, which squares the errors and thus gives more weight to larger errors.

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**
  - **Advantages of MAE:**
    - The MAE you get is in the same unit as the output variable.
    - It is most Robust to outliers.
  - **Disadvantages of MAE:**
    - The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

```
from sklearn.metrics import mean_absolute_error
print("MAE", mean_absolute_error(y_test,y_pred))
```

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**

Problem Statement: Imagine you are a data scientist working for a startup that sells handmade crafts online. The company recently implemented a new pricing algorithm to predict the price of crafts based on various features like size, material, and complexity. You want to evaluate the performance of this new algorithm.

Here's a set of actual prices of crafts and the prices predicted by the algorithm:

Craft Item	Actual Price ($)	Predicted Price ($)
Necklace	25	28
Bracelet	15	14
Earrings	20	22
Ring	30	29
Brooch	40	38

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**

Solution:

Here, $n = 5$,

MAE = 1/5 * (|25-28| + |15-14| + |20-22| + |30-29| + |40-38|) = 1/5 * (3+1+2+1+2) = 1/5 * (9)

- **From the above result, on average the pricing algorithm is off by $ 1.8**

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**

Solution:

Here, $n = 5$,

MAE = 1/5 * (|25-28| + |15-14| + |20-22| + |30-29| + |40-38|) = 1/5 * (3+1+2+1+2) = 1/5 * (9)

- **From the above result, on average the pricing algorithm is off by $ 1.8**

Evaluation metrics for a Linear Regression

- **Mean Absolute Error (MAE):**

```
from sklearn.metrics import mean_absolute_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print("Mean Absolute Error:", mean_absolute_error(y_true, y_pred))

#OUTPUT
Mean Absolute Error: 0.5
```

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**
  - MSE is a most used and very simple metric with a little bit of change in mean absolute error.
  - Mean squared error states that finding the squared difference between actual and predicted value.
  - In MAE, we are finding the absolute difference and here we are finding the squared difference.
  - **What actually the MSE represents?**
  - It represents the squared distance between actual and predicted values for all data points.
  - **we perform squared to ensure that negative and positive differences don't cancel each other out.**

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**
  - **Advantages:**
    - Since it is differentiable and has a convex shape, it is easier to optimize.
  - **Disadvantages:**
    - **MSE penalizes large errors.** This means that it squares the errors and thus gives more weight to larger errors.
    - It is sensitive to outliers as large errors contribute significantly to the overall score.
    - The value you get after calculating MSE is a squared unit of output.
    - For example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared.

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**
  - Calculate MSE for Number of sales done for bikes in below months:

Month	Acutal Number of Sales	Forecasted Number of Sales
January	35	38
February	37	35
March	39	39
April	40	37
May	36	35
June	35	37
July	35	35
August	38	38
September	37	39

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**
  - Calculate MSE for Exam scores of students in exam:

Hours Studied	Exam Score	Predicted Score
1	68	79.03
1	78	79.03
1	75	79.03
2	83	82.11
2	80	82.11
2	78	82.11
2	89	82.11
2	93	82.11
3	90	85.19
3	91	85.19
4	94	88.27
5	88	91.35
5	84	91.35
5	90	91.35
6	94	94.43

Evaluation metrics for a Linear Regression

- **Mean Squared Error (MSE):**

  from sklearn.metrics import mean_squared_error

  print("MSE",mean_squared_error(y_test,y_pred))

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print("Mean Squared Error:", mean_squared_error(y_true, y_pred))

#OUTPUT
Mean Squared Error: 0.375
```

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - As RMSE is clear by the name itself, that it is a simple square root of mean squared error.
  - The Root Mean Squared Error (RMSE) has been used as a standard statistical metric to measure model performance in meteorology, air quality, and climate research studies.
  - RMSE functions similarly to MAE (that is, you use it to determine how close the prediction is to the actual value on average), but with a minor difference.

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - As RMSE is clear by the name itself, that it is a simple square root of mean squared error.
  - If you are concerned about large errors, RMSE is a good metric to use. If the model overestimated or underestimated some points in the prediction (because the residual will be square, resulting in a large error), you should use RMSE.
  - RMSE is a popular evaluation metric for regression problems because it not only calculates how close the prediction is to the actual value on average, but it also indicates the effect of large errors.
  - Large errors will **have an impact on the RMSE result**.

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - It quantifies the differences between predicted values and actual values, squaring the errors, taking the mean, and then finding the square root.
  - RMSE provides a clear understanding of the model's performance, with lower values indicating better predictive accuracy relative root mean square error.
  - It is computed by taking the square root of MSE.
  - RMSE is also called the **Root Mean Square Deviation.**

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - **Advantages:**
    - RMSE does not penalize the errors as much as MSE does due to the square root.
    - It serves as a heuristic (enable someone to discover something for themselves) for training models.
  - **Disadvantages:**
    - RMSE is dependent on the scale of the data. It increases in magnitude if the scale of the error increases.
    - One major drawback of RMSE is its sensitivity to outliers and the outliers have to be removed for it to function properly.
    - RMSE increases with an increase in the size of the test sample. This is an issue when we calculate the results on different test samples.

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values

$y_1, y_2, \ldots, y_n$ are observed values

n is the number of observations

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - Calculate RMSE for below data:

Predicted Points ($\hat{y}_i$)	Actual points (y_i)
14	12
15	15
18	20
19	16
25	20
18	19
12	16
12	20
15	16
22	16

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**
  - Calculate RMSE for below data:

Month	Actual	Forecasted
January	67	70
February	50	49
March	36	38
April	74	76
May	84	83
June	84	80
July	64	67
August	34	30
September	23	20
October	72	75
November	62	60
December	42	38

Evaluation metrics for a Linear Regression

- **Root Mean Squared Error (RMSE):**

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print("Root Mean Squared Error:", mean_squared_error(y_true, y_pred, squared=False))

#OUTPUT
Root Mean Squared Error: 0.6123724356957945
```

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - It measures the proportion of variance of the dependent variable explained by the independent variable.
  - R squared is a popular metric for identifying model accuracy.
  - It tells how close are the data points to the fitted line generated by a regression algorithm.
  - A **larger R squared value indicates a better fit**. This helps us to find the relationship between the independent variable towards the dependent variable.
  - MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - With help of R squared we have a baseline model to compare a model which none of the other metrics provides.
  - Basically R2 squared calculates how must regression line is better than a mean line.
  - R2 squared is also known as **Coefficient of Determination** or sometimes also known as **Goodness of fit.**
  -

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - **When to Use R Squared?**
    - Both independent and dependent variables must be continuous.
    - When the independent and dependent variables have linear relationship (+ve or -ve) between them.

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - **What is the significance of R squared**
    - R-squared values range from 0 to 1, usually expressed as a percentage from 0% to 100%.
    - This value of R square tells you how well the data fits the line you've drawn.
    - The higher the model's R-Squared value, the better the regression line fits the data.
    - R-squared values very close to 1 are likely over-fitting of the model and should be avoided.

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**

$$\text{R-Squared} = 1-\left(\frac{SSR}{SST}\right)=1-\frac{\sum_{i=1}^{n}(yi-\widehat{yi})^2}{\sum_{i=1}^{n}(yi-\overline{yi})^2}$$

where:

SSR is the sum of squared residuals (i.e., the sum of squared errors)

SST is the total sum of squares (i.e., the sum of squared deviations from the mean)

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - **What is the significance of R squared**
    - SSE is the sum of the square of the difference between the actual value and the predicted value

$$SSE = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**
  - **What is the significance of R squared**
    - SST is the total sum of the square of the difference between the actual value and the mean of the actual value.

$$SST = \sum_{i=1}^{m} (y_i - \bar{y})^2$$

    - yi is the observed target value, ŷi is the predicted value, and y-bar is the mean value, m represents the total number of observations.

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**

x	y
3	22
5	24
5	28
7	20
9	28
12	31
14	37
17	33

Evaluation metrics for a Linear Regression

- **R-squared (R^2) Score:**

```python
from sklearn.metrics import r2_score
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print("R2 Score:", r2_score(y_true, y_pred))

#OUTPUT
R2 score: 0.9486081370449679
```

Thank You!!!

Happy Learning!!!