Regression Problem:

In statistics and machine learning, a regression problem involves predicting a continuous value based on one or more input features. The goal is to find a mathematical relationship that best describes the data. This relationship can then be used to make predictions on new data.

Examples of regression problems include predicting house prices based on features like size, number of bedrooms, and location; forecasting stock prices based on historical data; or estimating the demand for a product based on factors like price, advertising expenditure, and seasonality.

Linear Regression:

Linear regression is one of the simplest and most commonly used techniques for modeling the relationship between a dependent variable (the output we want to predict) and one or more independent variables (the inputs). It assumes that there's a linear relationship between the inputs and the output.

In simple linear regression, there's only one independent variable, while in multiple linear regression, there are multiple independent variables.

The equation for a simple linear regression model with one independent variable can be expressed as:

$$y=\beta0+\beta1\times x$$

Where:

y is the dependent variable (the output we want to predict).

x is the independent variable (the input feature).

β0 is the y-intercept (the value of y when x is zero).

β1 is the slope of the line (the change in y for a one-unit change in x).

The goal of linear regression is to estimate the values of

β0 AND β 1

that minimize the sum of squared differences between the observed and predicted values of y. This is typically done using methods like ordinary least squares (OLS) or gradient descent.

Once the model is trained, it can be used to make predictions on new data by plugging in the values of the independent variables into the equation.

Linear regression is widely used because it's simple, interpretable, and often provides a good baseline model for more complex techniques. However, it's important to note that it makes several assumptions about the data, such as linearity, independence of errors, constant variance of errors (homoscedasticity), and normality of errors. These assumptions should be checked before using linear regression and alternative techniques should be considered if they're violated.

Application of regression

- Stock Price Prediction

- Sales Forecasting
- Real Estate Price Prediction
- Player Performance Prediction
- Energy Consumption Prediction

How does linear regression works?

Mathematics Calculations:

Ads (X)	Sales (Y)
1	14
3	24
2	18
1	17
3	27

Intercept

- Equation of Line:

$Y = mx + c$ *or* $\hat{Y} = b_1(X) + b_0$

Slope:

"Unit of change in Y due to X"

- Formula for regression coefficient:

$$b_1 = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sum (X - \bar{X})^2}$$

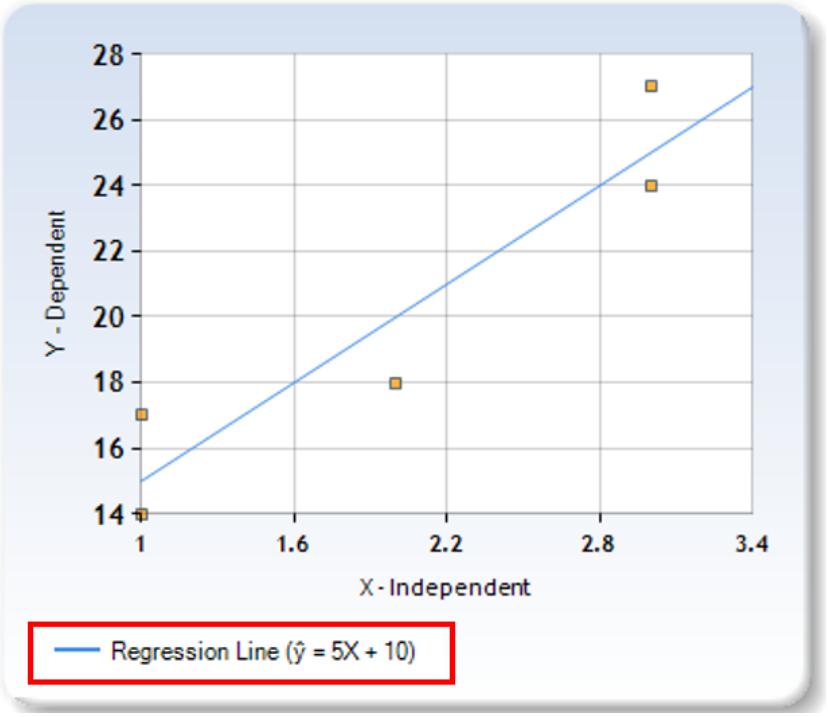
- Formula for Intercept:

$$b_0 = \bar{Y} - b_1(\bar{X})$$

Mathematics Calculations:

Ads (X)	Sales (Y)	$(X - \bar{X})$	$(Y - \bar{Y})$	$(X - \bar{X})(Y - \bar{Y})$	$(X - \bar{X})^2$
1	14	-1	-6	6	1
3	24	1	4	4	1
2	18	0	-2	0	0
1	17	-1	-3	3	1
3	27	-1	7	7	1
$\bar{X} = 2$	$\bar{Y} = 20$			$\sum (X - \bar{X})(Y - \bar{Y}) = 20$	$\sum (X - \bar{X})^2 = 4$

Regression Line (Chart)



Evaluation Metrics:

Evaluation metrics are used to assess the performance of a machine learning model. They provide a quantitative measure of how well the model is performing on unseen data. Different evaluation metrics are used depending on the nature of the problem (classification, regression, etc.) and the specific goals of the model.

Mean Squared Error (MSE): Mean squared error (MSE) is a commonly used metric for evaluating the performance of regression models. It measures the average squared difference between the actual values and the predicted values produced by the model. The formula for MSE is:

Mean squared error (MSE).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

- N is number of values
- Y_i is actual values
- $\hat{Y}$ is predicted value of Y

In simple terms, MSE calculates the average squared difference between the actual and

predicted values. A lower MSE indicates that the model is better at predicting the target variable.

Coefficient of Determination (R^2):

The coefficient of determination, often denoted as R^2, is a measure of how well the independent variables explain the variability of the dependent variable. It ranges from 0 to 1, where 0 indicates that the model does not explain any of the variability of the dependent variable, and 1 indicates that the model perfectly explains all the variability.

The formula for R^2 is:

Coefficient of determination (R^2):

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

- SSR: Sum of square due to regression
- SST: Total sum of squares
- $\overline{Y}$ is mean of Y
- $\widehat{Y}$ is predicted value of Y

R^2can be interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variables.

A higher R^2value indicates that a larger proportion of the variance is explained by the model, and thus the model fits the data better.

Relationship Between MSE and R^2:

MSE measures the average squared difference between actual and predicted values.

R^2 measures the proportion of variance explained by the model.

Typically, as the MSE decreases (i.e., the model becomes more accurate),

R^2increases. However, it's important to consider both metrics together to get a comprehensive understanding of the model's performance.

SKlearn and regression code

This Python code utilizes the LinearRegression class from the sklearn.linear_model module to perform linear regression. Here's an explanation of each part:

```
1: from sklearn.linear_model import LinearRegression
2: regressor = LinearRegression()
3: regressor.fit(X_train, y_train)
4: print(regressor.intercept_)
5: print(regressor.coef_)
```

1:This line imports the LinearRegression class from the sklearn.linear_model module. sklearn (also known as scikit-learn) is a widely-used Python library for machine learning, and LinearRegression is a class within this library that implements linear regression.

2: Creating a LinearRegression object: Here, an instance of the LinearRegression class is created and assigned to the variable regressor. This object will be used to perform linear regression.

3: Fitting the model to the training data: The fit() method of the LinearRegression object is called to fit the linear regression model to the training data. X_train represents the feature matrix (independent variables) and y_train represents the target vector (dependent variable).

4: Printing the intercept and coefficients: regressor.intercept_ prints the y-intercept of the linear regression model. This is the value of the dependent variable when all independent variables are zero. regressor.coef_ prints the coefficients of the independent variables in the linear regression model. These coefficients represent the change in the dependent variable for a one-unit change in the corresponding independent variable, while holding other variables constant.

Multiple linear regression

Multiple Linear Regression by Hand (Step-by-Step) - Statology

Task

- An attached file, Regression.ipynb, contains code. Please review and comprehend it, then proceed to implement the regression code on the provided datasets.
- Write code from scratch to execute single-variable linear regression on the student.csv dataset. Write code from scratch to calculate MSE and R^2
- write code from scratch to conduct multivariable regression on this dataset.

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11