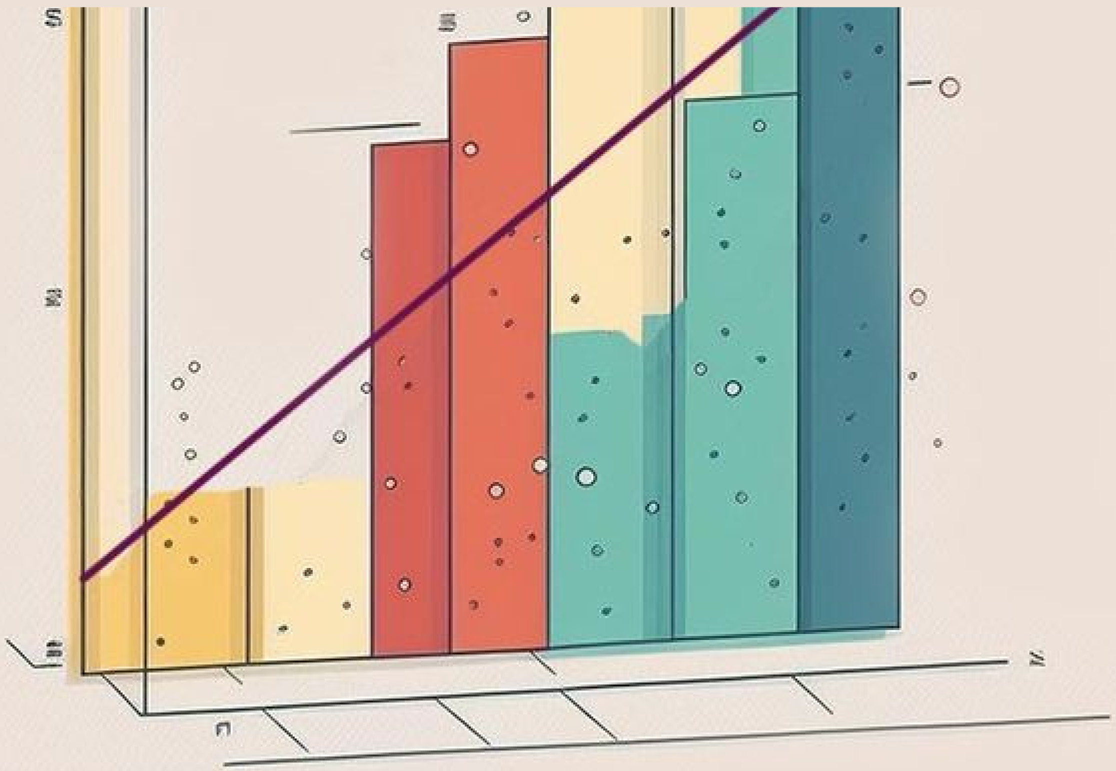
Linear Regression

Unveiling Patterns, Predicting Outcomes: Linear Regression for Data Insight



Linear Regression

Linear regression is a type of machine learning algorithm that is used to model the linear relationship between a dependent variable and one or more independent variables. It works by fitting a straight line (or hyperplane in higher dimensions) to the data that minimizes the distance between the points and the line. This line can then be used to make predictions on new data.

Linear Regression

Linear regression is a statistical modeling technique used to understand the relationship between a dependent variable (output) and one or more independent variables (inputs). It assumes a linear relationship between the variables, meaning that the dependent variable can be expressed as a linear combination of the independent variables.

Types of Linear Regression

1. Simple Linear Regression: Simple linear regression is used when we have a single independent variable (X) and a single dependent variable (y). It models the relationship between X and y as a straight line. The formula for simple linear regression is:

$$y = \beta_0 + \beta_1 X + \varepsilon$$

- y represents the dependent variable (output variable).
- X represents the independent variable (input variable).
- β_0 is the y-intercept, which represents the value of y when X is zero.
- β_1 is the slope of the line, indicating how much y changes for each unit change in X.
- ε represents the error term or residual, which accounts for the variability not explained by the line.

@neuralcode.io

Cost Function

The cost function in linear regression measures the average squared difference between the predicted values ($\hat{y}$) and the actual values (y). It quantifies the error and guides the model towards minimizing it. The formula for the Mean Squared Error (MSE) cost function is:

$$MSE = (1/n) * \Sigma(y - \hat{y})^2$$

- n is the number of data points.
- y represents the actual values of the dependent variable.
- $\hat{y}$ represents the predicted values of the dependent variable.

The goal is to minimize the MSE, finding the coefficients that lead to the smallest average squared difference between predicted and actual values.

Implementing a simple linear regression model from scratch without using the sklearn library in Python

Samsung

```python
import numpy as np

# Define the input variable (X) and the output variable (y)
X = np.array([1, 2, 3, 4, 5])  # Input variable
y = np.array([2, 4, 5, 8, 10])  # Output variable

# Calculate the mean of X and y
mean_X = np.mean(X)
mean_y = np.mean(y)

# Calculate the slope (m) and y-intercept (b) using the formula
numerator = np.sum((X - mean_X) * (y - mean_y))
denominator = np.sum((X - mean_X) ** 2)
m = numerator / denominator
b = mean_y - m * mean_X

# Predict the output values
y_pred = m * X + b

# Print the coefficients and predicted values
print("Slope (m):", m)
print("Intercept (b):", b)
print("Predicted values:", y_pred)
```

@neuralcode.io

Unlock the power of code with Python!
Join me on a journey into artificial intelligence and programming wonders. 🚀

@neuralcode.io

LIKE ❤️
&
FOLLOW 👍 OUR PAGE FOR MORE CONTENT LIKE THIS🔥