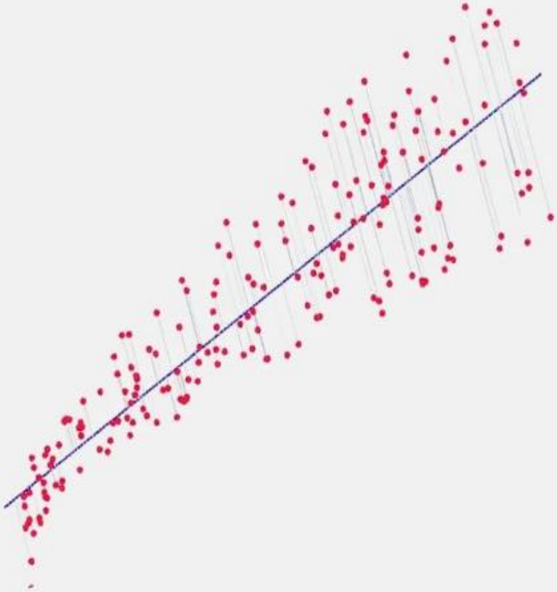
Linear **Regression**

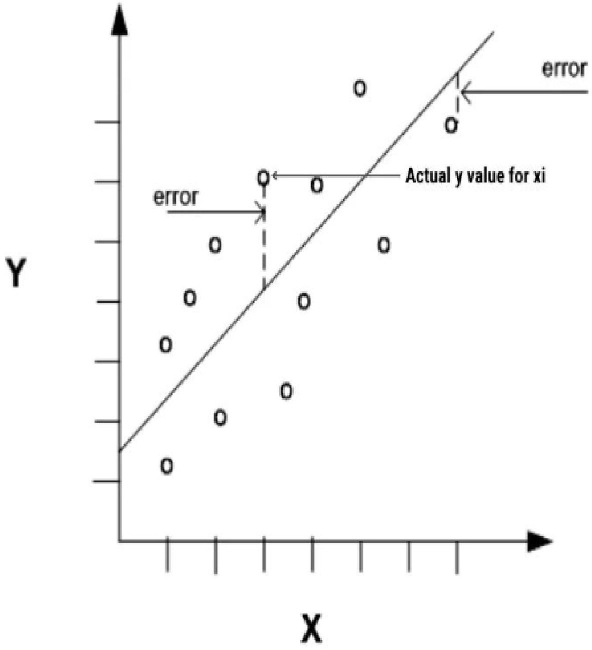
Explained with Code



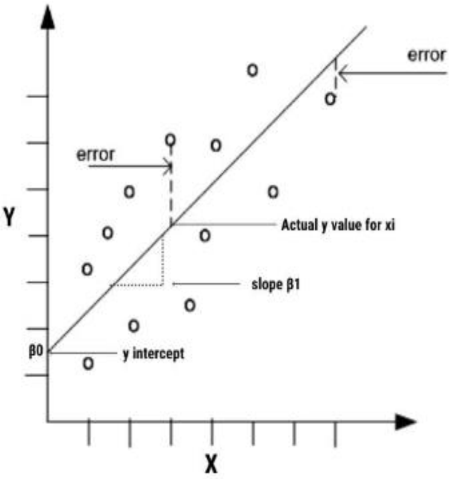
Swipe
to see in
detail

What is Linear regression

Linear regression is a statistical method used to **model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data.** The goal is to find the best-fitting line that minimizes the sum of the squared differences between the observed values and the values predicted by the line.



Equation

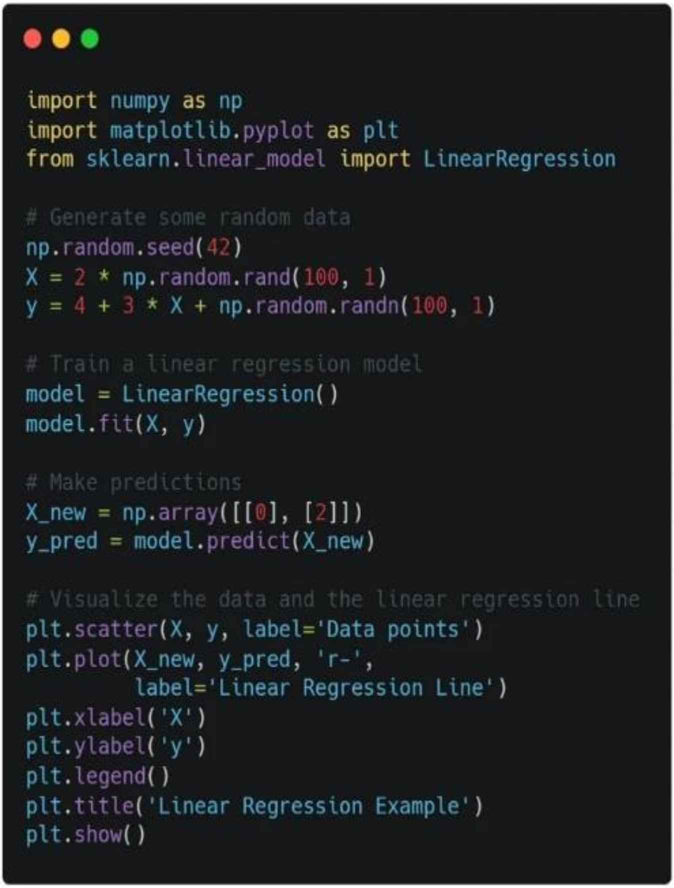


The equation for the line is: $Y=\beta 0+\beta 1X+\varepsilon$ Here,

- Y is the dependent variable,
- X is the independent variable,
- β0 is the y-intercept,
- β1 is the slope of the line, and
- ε represents the error term.

The goal is to find the best-fitting line by adjusting the values of $\beta 0$ and $\beta 1$ to minimize the sum of squared errors between the observed values and the values predicted by the lin

Code



```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Generate some random data
np.random.seed(42)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

# Train a linear regression model
model = LinearRegression()
model.fit(X, y)

# Make predictions
X_new = np.array([[0], [2]])
y_pred = model.predict(X_new)

# Visualize the data and the linear regression line
plt.scatter(X, y, label='Data points')
plt.plot(X_new, y_pred, 'r-',
         label='Linear Regression Line')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.title('Linear Regression Example')
plt.show()
```