The Math and AI of what I have done and the code. 😊

Linear regression is a type of machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. In our case, the **dependent variable** is the **final price of a house**, and the **independent features** are the **asked price** and other characteristics of the house.

The goal of linear regression is to find the best linear equation that can predict the value of the dependent variable based on the independent variables. The equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).

The equation of a linear function with one independent variable is:

f(x)=β0+β1x

where β0 is the intercept and β1 is the slope. The intercept is the value of the dependent variable when the independent variable is zero, and the slope is the rate of change of the dependent variable with respect to the independent variable.

The equation of a linear function with multiple independent variables is:

f(x)=β0+β1x1+β2x2+...+βnxn

Where β0 is the intercept and βi are the coefficients for each independent variable xi. The coefficients represent how much the dependent variable changes for a unit change in each independent variable, holding all other variables constant.

To find the best linear equation, we need to find the optimal values for the intercept and coefficients that minimise the error between the predicted and actual values of the dependent variable.

The error is also known as the residual, and it is calculated as:

ei=yi−f(xi)

where ei is the error for the i-th observation, yi is the actual value of the dependent variable, and)f(xi) is the predicted value of the dependent variable.

One way to measure the total error for all observations is to use the sum of squared errors (SSE), which is calculated as:

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

where **n is the number of observations**. The SSE measures how much variation in the dependent variable is not explained by the linear equation. The lower the SSE, the better the fit of the linear equation.

To minimize the SSE, we can use different methods, such as the normal equation for gradient descent. The normal equation is an analytical solution that directly computes the optimal values for the intercept and coefficients by solving a system of linear equations. The gradient descent is an iterative solution that gradually updates the values for the intercept and coefficients by moving in the direction of steepest descent of SSE.

In our code, we have used both methods to find the best linear equation for our data set. We have also used scikit-learn, which is a popular Python library for machine learning, to implement linear regression in a few lines of code. **We have evaluated our model performance using mean squared error (MSE) and R-squared score (R2), which are common metrics for regression problems. The MSE measures how close our predictions are to the actual values, and it is calculated as:**

$$MSE = \frac{1}{n}\sum_{i=1}^{\cdot\cdot}(y_i - f(x_i))^2$$

The **R2 measures how well our model explains the variation in the dependent variable**, and it is calculated as:

R2 = 1- SSE / SST

where **SST is the total sum of squares, which is calculated as:**

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

where **y is the mean value of the dependent variable. The R2 ranges from 0 to 1, where 0 means no fit and 1 means perfect fit.**