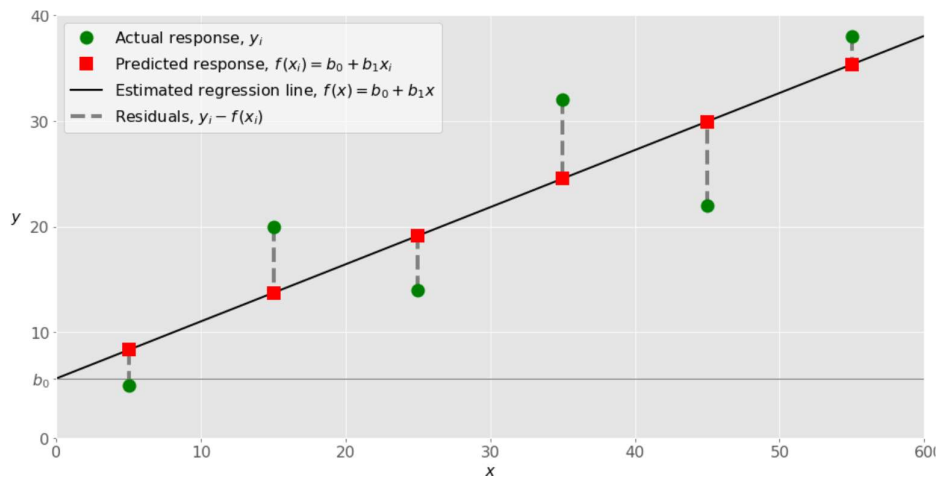


Experiment 3: Implement linear regression in Python on a given dataset.**Theory:****1. Linear Regression**

Regression is a technique from statistics that is used to predict values of a desired target quantity when the target quantity is continuous. When implementing linear regression of some dependent variable y on the set of independent variables $x = (x_1, \dots, x_r)$, where r is the number of predictors, you assume a linear relationship between y and x : $y = \beta_0 + \beta_1 x_1 + \dots + \beta_r x_r + \epsilon$. This equation is the regression equation. $\beta_0, \beta_1, \dots, \beta_r$ are the regression coefficients, and ϵ is the random error.

The following figure illustrates simple linear regression:

**Implementation:****1. sklearn.linear_model.LinearRegression**

Syntax:

```
class sklearn.linear_model.LinearRegression()
```

Gives ordinary least squares Linear Regression.

2. **corr()** - Compute pairwise correlation of columns, excluding NA/null values.

3. **reg.coef_** - Estimated coefficients for the linear regression problem.

4. **reg.intercept_** - Independent term in the linear model.

5. sklearn.impute.KNNImputer

Syntax-

```
class sklearn.impute.KNNImputer(*, missing_values=nan, n_neighbors=5, weights='uniform') ⓘ
```

Imputation for completing missing values using k-Nearest Neighbors.

6. `sklearn.model_selection.train_test_split`

Syntax:

```
sklearn.model_selection.train_test_split(*arrays, test_size=None, train_size=None, random_state=None, shuffle=True) ¶
```

Split arrays or matrices into random train and test subsets.

7. **fit()**-Fit linear model & **predict()**-Predict using the linear model.

8. `sklearn.metrics.r2_score`

Syntax:

```
sklearn.metrics.r2_score(y_true, y_pred) ¶
```

Computes R2 (coefficient of determination) regression score function.

About Dataset:

(Describe your dataset)

Conclusion: In this way, we have studied and implemented the linear regression algorithm and predicted the values of continuous-valued attribute.