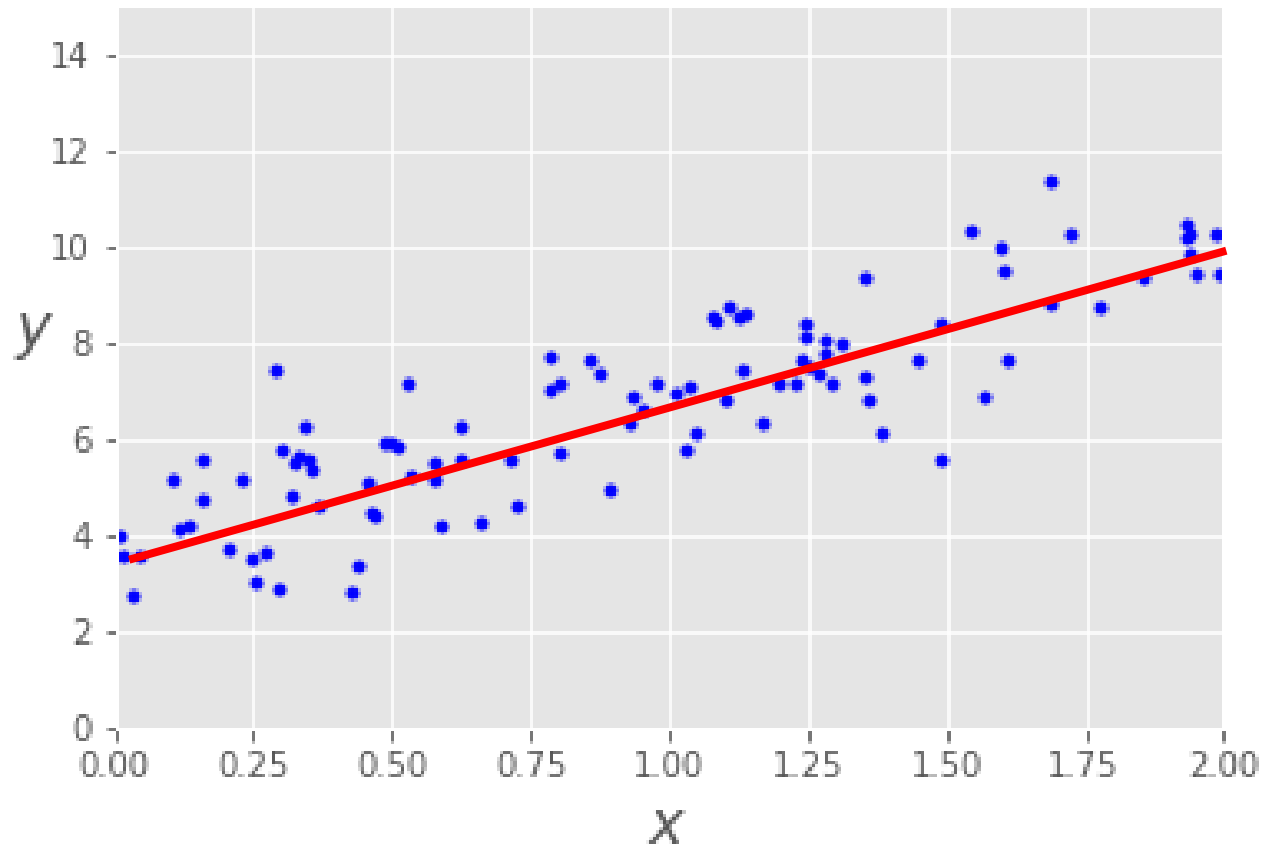


# Regresi Linear

Made Satria Wibawa, M.Eng.  
2020

# PENDAHULUAN

# Regresi



Proses untuk memperkirakan hubungan antara variabel tergantung (dependent variable) dengan variabel bebas (independent variable)

*jika nilai  $x$  diketahui berapakah nilai  $y$ ?*

# Asumsi Linearitas

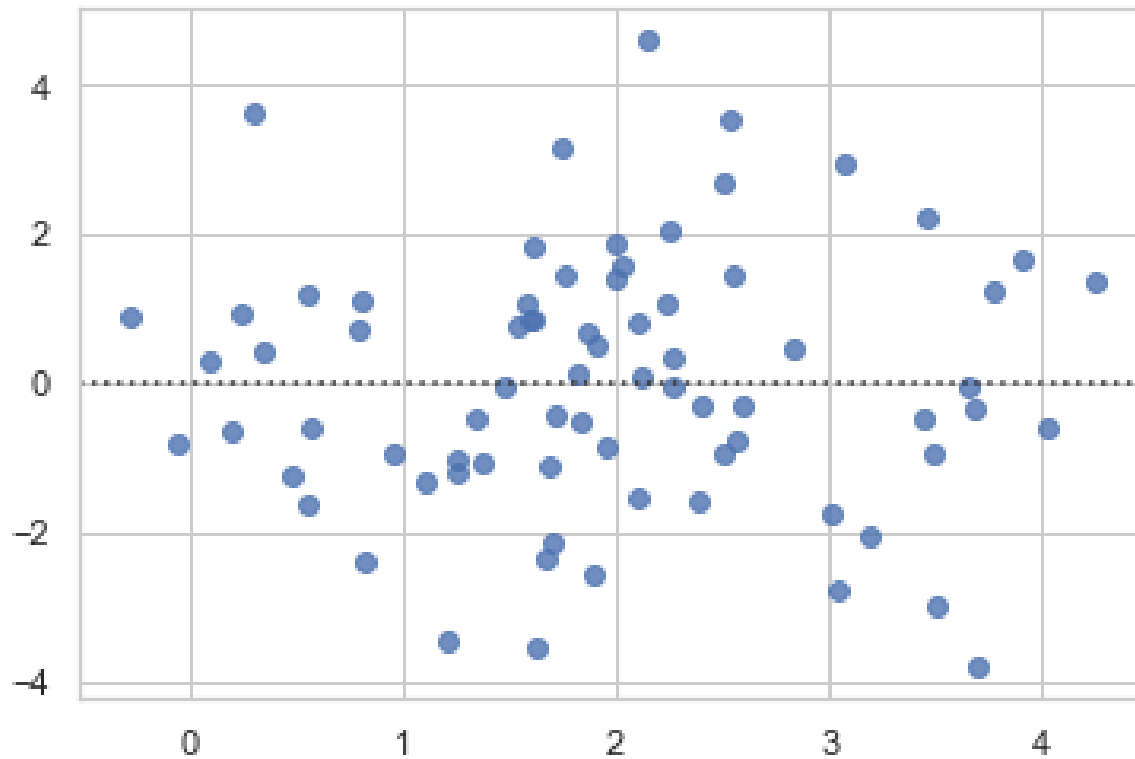
- Variabel tergantung ( $y$ ) harus memiliki hubungan linear terhadap variabel bebas ( $x$ ). (gunakan scatter plot)
- Untuk setiap nilai  $x$ 
  - Nilai  $y$  bersifat independen, ditunjukkan dengan pola acak pada plot residual\*
  - Nilai  $y$  memiliki distribusi normal. Skewness dapat ditoleransi jika ukuran data besar

*\*anda dapat menggunakan seaborn untuk membuat residual plot (sns.residplot)*

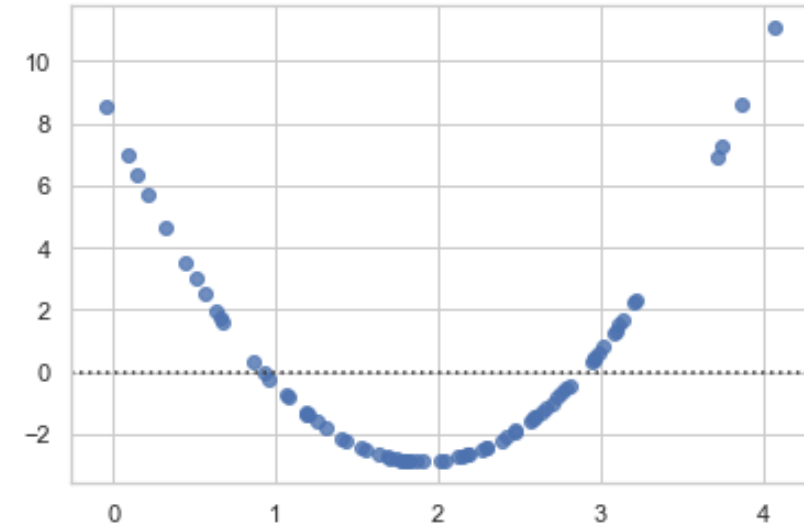
# Residual Plot

```
1 import seaborn as sns
2 sns.residplot(x,y)
```

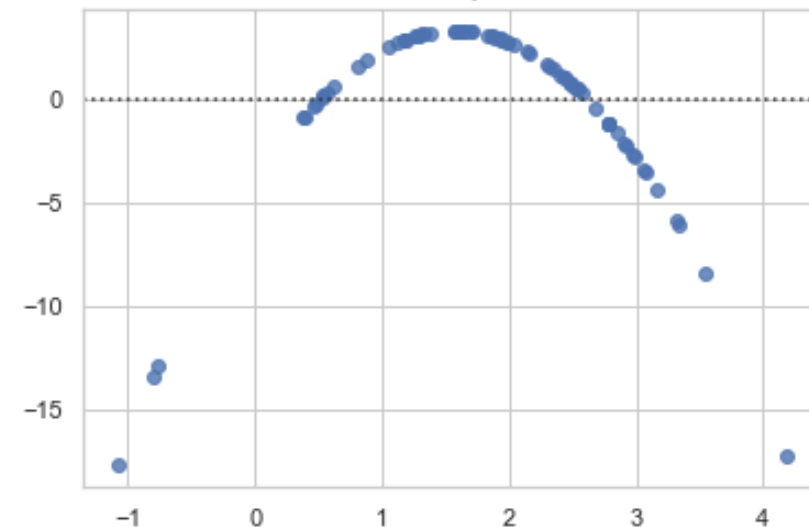
Random Pattern



U-Shaped Pattern



Inverted U-Shaped Pattern



# ORDINARY LEAST SQUARE (PERSAMAAN KUADRAT TERKECIL)

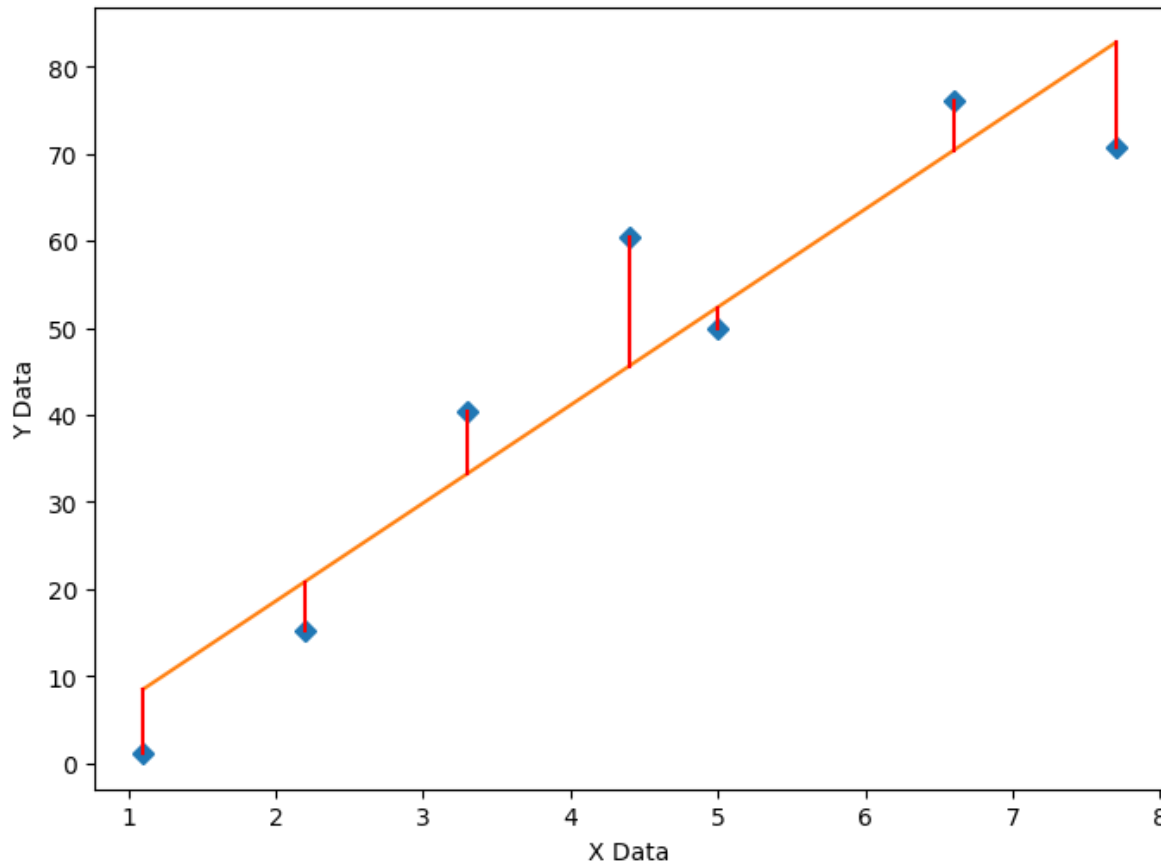
# Persamaan

$$y = a + bx$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$a = \frac{\sum y - b(\sum x)}{n}$$

# Evaluasi



1. mean squared error

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

2. root mean squared error

$$RMSE = \sqrt{MSE}$$

3. koefisien determinasi

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2}$$



# Koefisien Determinasi

- Merupakan korelasi antara nilai  $y$  (nilai asli) dengan nilai  $\hat{y}$  (hasil prediksi)
- Mempunyai rentang nilai 0-1
- 0 berarti variabel tergantung tidak dapat diprediksikan dari variabel bebas
- 1 berarti variabel bebas dapat memprediksikan variabel tergantung tanpa error
- Nilai di antar 0-1 berarti sejauh mana variabel tergantung dapat diprediksikan. Misalnya nilai 0.1 berarti 10 persen dari hasil dapat diprediksi, dst

# CONTOH PERHITUNGAN

# Contoh

$x$	$y$
20	64
16	61
34	84
23	70
27	88
32	92
18	72
22	77



	$y$	$x$	$xy$	$x^2$	$y^2$
	64	20	1280	400	4096
	61	16	976	256	3721
	84	34	2856	1156	7056
	70	23	1610	529	4900
	88	27	2376	729	7744
	92	32	2944	1024	8464
	72	18	1296	324	5184
	77	22	1694	484	5929
$\Sigma$	608	192	15032	4902	47094

# Perhitungan

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} = \frac{8(15032) - (192)(608)}{8(4902) - (192)^2} = 1.497$$

$$a = \frac{\sum y - b(\sum x)}{n} = \frac{(608) - 1.497(192)}{8} = 40.082$$

$$\mathbf{y = 40.082 + 1.497x}$$

# Perhitungan-Evaluasi

$y$	$\hat{y}$	$(y - \hat{y})^2$
64	70.01	36.16
61	64.03	9.16
84	90.97	48.52
70	74.50	20.28
88	80.49	56.04
92	87.97	16.22
72	67.02	24.79
77	73.01	15.95
$\Sigma$		227.49


$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{8} (227.49) = 28.43$$

$$RMSE = \sqrt{MSE} = \sqrt{28.43} = 5.33$$

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2} = 1 - \frac{227.49}{886} = 0.74$$

# IMPLEMENTASI PYTHON

# Implementasi Python



Home Installation Documentation ▾ Examples

Google Custom Search

Previous  
1. Supervised...

Next  
1.2. Linear...

Up  
1. Supervised...

scikit-learn v0.21.3  
Other versions

Please cite us if you use the software.

1.1. Generalized Linear Models

1.1.1. Ordinary Least Squares

- 1.1.1.1. Ordinary Least Squares Complexity
- 1.1.2. Ridge Regression
  - 1.1.2.1. Ridge Complexity
  - 1.1.2.2. Setting the regularization parameter: generalized Cross-Validation
- 1.1.3. Lasso
  - 1.1.3.1. Setting regularization parameter
    - 1.1.3.1.1. Using cross-validation
    - 1.1.3.1.2. Information-criteria based model selection
    - 1.1.3.1.3. Comparison with the regularization parameter of SVM
- 1.1.4. Multi-task Lasso
- 1.1.5. Elastic-Net
- 1.1.6. Multi-task Elastic-Net
- 1.1.7. Least Angle Regression
- 1.1.8. LARS Lasso
  - 1.1.8.1. Mathematical formulation
- 1.1.9. Orthogonal Matching Pursuit (OMP)
- 1.1.10. Bayesian Regression
  - 1.1.10.1. Bayesian Ridge Regression
  - 1.1.10.2. Automatic Relevance Determination - ARD
- 1.1.11. Logistic regression
- 1.1.12. Stochastic Gradient Descent - SGD
- 1.1.13. Perceptron
- 1.1.14. Passive Aggressive Algorithms
- 1.1.15. Robustness regression:

## 1.1. Generalized Linear Models

The following are a set of methods intended for regression in which the target value is expected to be a linear combination of the features. In mathematical notation, if  $\hat{y}$  is the predicted value.

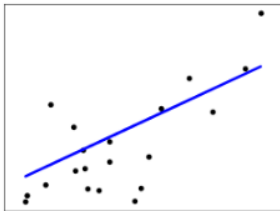
$$\hat{y}(w, x) = w_0 + w_1x_1 + \dots + w_px_p$$

Across the module, we designate the vector  $w = (w_1, \dots, w_p)$  as `coef_` and  $w_0$  as `intercept_`.

To perform classification with generalized linear models, see [Logistic regression](#).

### 1.1.1. Ordinary Least Squares

`LinearRegression` fits a linear model with coefficients  $w = (w_1, \dots, w_p)$  to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Mathematically it solves a problem of the form:

$$\min_w ||Xw - y||_2^2$$


`LinearRegression` will take in its `fit` method arrays  $X$ ,  $y$  and will store the coefficients  $w$  of the linear model in its `coef_` member:

```
>>> from sklearn import linear_model
>>> reg = linear_model.LinearRegression()
>>> reg.fit([[0, 0], [1, 1], [2, 2]], [0, 1, 2])
...
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
>>> reg.coef_
array([0.5, 0.5])
```

[https://scikit-learn.org/stable/modules/linear\\_model.html](https://scikit-learn.org/stable/modules/linear_model.html)

