Session 33

Regression

Table of Content
What will We Learn Today?

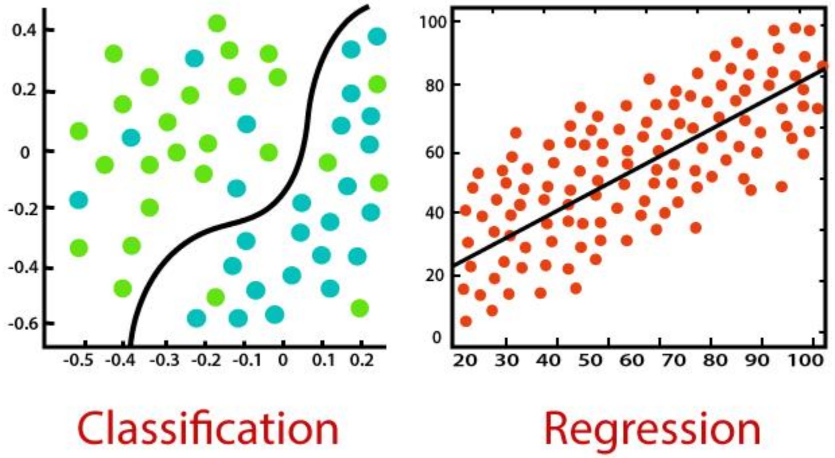
Regression

Regression

- **Regression**: metode yang mencoba untuk menentukan kekuatan dan karakter hubungan antara satu variabel dependen dan serangkaian variabel lainnya (dikenal sebagai independent variables).

- *Regression algorithms = continuous values (such as price, salary, age, etc).*
- *Classification algorithms = discrete values (such as stroke or normal, spam or not spam, etc)*

- Keduanya masuk dalam kategori *supervised learning*



Classification, regression, clustering

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built
221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955
538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951
180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933
604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965
510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987

Regression (house price dataset)

	ID	Sex	Marital status	Age	Education	Income	Occupation
0	100000001	0	0	67	2	124670	1
1	100000002	1	1	22	1	150773	1
2	100000003	0	0	49	1	89210	0
3	100000004	0	0	45	1	171565	1
4	100000005	0	0	53	1	149031	1

Clustering (customer dataset)

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

Classification (stroke dataset)

Linear Regression

Linear Regression

- Membangun hubungan diantara dua variables dengan garis lurus.
- Variabel independen merupakan variabel yang memengaruhi atau menyebabkan perubahan.
- Variabel dependen adalah variabel yang dipengaruhi atau yang menjadi akibat karena adanya variabel independen.

- **Simple linear regression:** $Y = a + bX + u$
- **Multiple linear regression:** $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_tX_t + u$

Where:

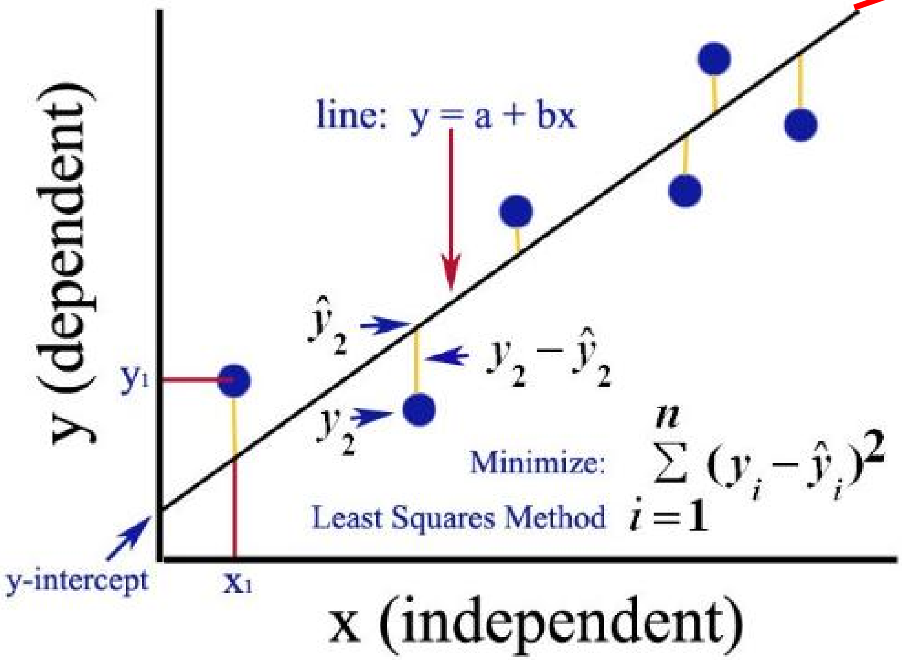
- Y = the variable that you are trying to predict (dependent variable).
- X = the variable that you are using to predict Y (independent variable).
- a = the intercept.
- b = the slope.
- u = the regression residual.

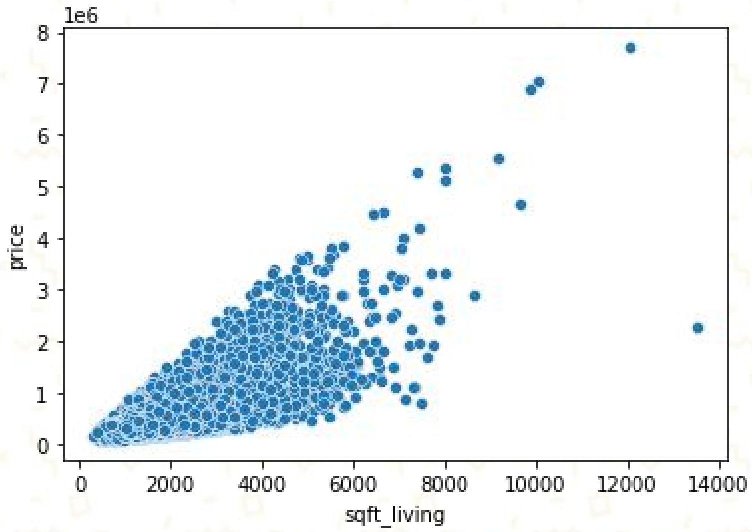
Linear Regression

- *Linear regression* (regresi linier) mencoba menggambar garis yang paling dekat dengan data dengan menemukan *slope* dan *intercept* dan meminimalkan *regression errors*.
- *Ordinary Least Squares (OLS)* adalah metode estimasi yang paling umum untuk model linier

Garis optimal yang memberikan nilai *sum of squared errors (SSE)* terendah

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

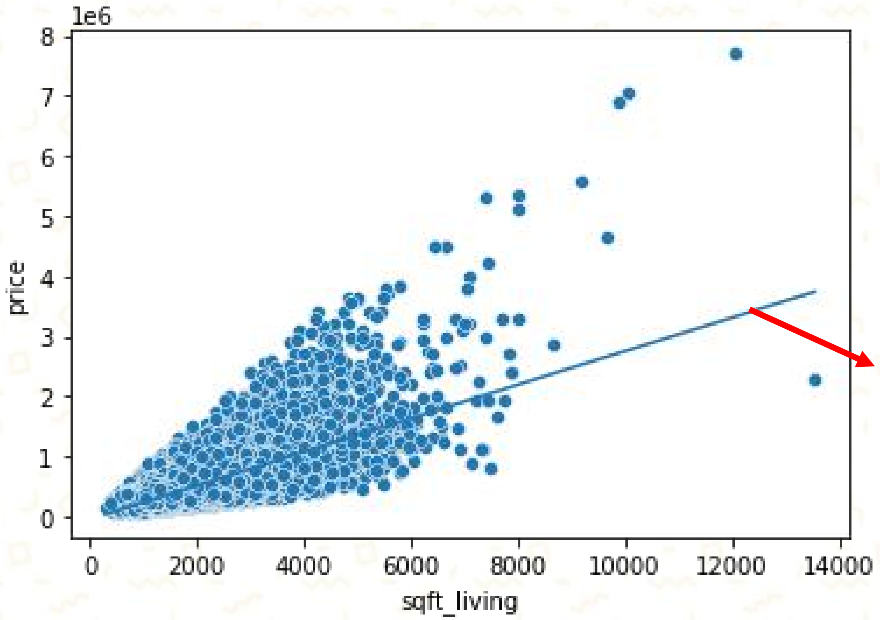




DigitalSkola
Uncover The World Of Digital Skills With Us

Linear Regression

- Example
  - y (*dependent variable*) = *price* (harga rumah)
  - x (*independent variable*) = *sqft_living* (luas rumah)




```
price = 279.51011741*sqft_living + -41947.45401876257
```

Q = Rumah dengan luas1000 *square feet*, berapa harganya kira kira?
A = USD 237562.663

Example

- *House Sales in King County, USA.*
- Dataset ini berhubungan dengan harga rumah di King County, yang termasuk juga Seattle. Berhubungan dengan rumah yang dijual dari Mei 2014 sampai Mei 2015.
- Source : https://www.kaggle.com/harlfoxem/housesalesprediction

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built
221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955
538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951
180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933
604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965
510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987

Linear Regression

- Kita bisa menggunakan library sklearn

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

df_X = df.drop(['id','date','price'],axis=1)
df_y = df['price']
X = df_X.astype(float).values
y = df_y.astype(float).values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
reg = LinearRegression()
reg.fit(X_train, y_train)
print('coefficient of determination of training set')
print(reg.score(X_train, y_train))
print('coefficient of determination of testing set')
print(reg.score(X_test, y_test))
print('coefficient')
print(reg.coef_)
print('intercept')
print(reg.intercept_)
print('prediction')
y_pred = reg.predict(X_test)
print(y_pred[:10])
print('real value')
print(y_test[:10])
```

```
coefficient of determination of training set
0.6995155846436758
coefficient of determination of testing set
0.6994627057969862
coefficient
[-3.43081477e+04  4.03129700e+04  1.12001375e+02  9.91841247e-02
  5.27154218e+03  5.43877177e+05  5.50830616e+04  2.31460673e+04
  9.49081794e+04  7.22190669e+01  3.97823083e+01 -2.59441847e+03
  2.19209734e+01 -5.56358731e+02  5.95216324e+05 -1.96904658e+05
  1.62077488e+01 -3.30430480e-01]
intercept
6641646.708113588
prediction
[ 458597.0676416   748993.75994814 1243303.75799055 1665116.95095444
  737302.05741739  283239.58524974  831732.87582315  495383.02095338
  385779.81919026  474179.42285135]
real value
[ 365000.  865000. 1038000. 1490000.  711000.  211000.  790000.  680000.
  384500.  605000.]
```

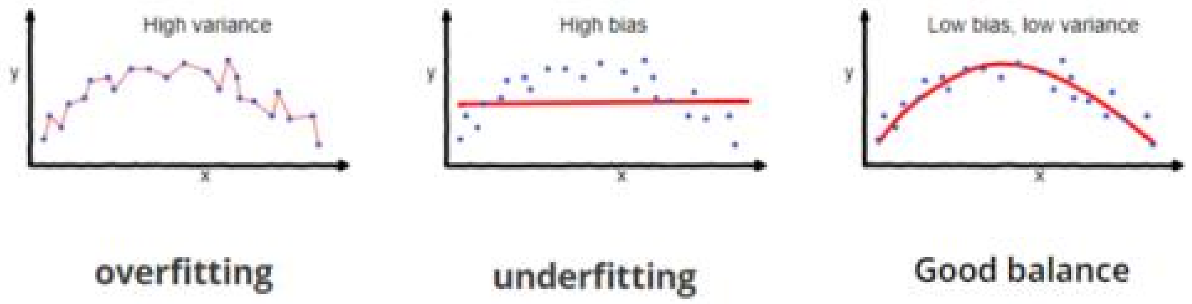
```
price = 279.51011741*sqft_living + -41947.45401876257
```

coefficient/slope/
kemiringan

intercept

Bias and variance

- *Linear regression* mencari nilai *coefficient* yang meminimalkan nilai *sum of squared errors (SSE)*.
- Tetapi mungkin ini bukan model terbaik, karena akan memberikan *coefficient* untuk semua features.
- Termasuk feature yang mempunyai “kemampuan prediksi yang rendah”.
- Ini akan menghasilkan model yang “high-variance, low bias”.
- Solusi = *regularization*
  - Kita bisa memodifikasi *cost function* untuk memberi batasan nilai *coefficients*.



https://towardsdatascience.com/bias-variance-and-regularization-in-linear-regression-lasso-ridge-and-elastic-net-8bf81991d0c5

Lasso and Ridge

L1 Regularization

- *Lasso (least absolute shrinkage and selection operator) regression*
- Lasso memberi tambahan “absolute value of magnitude” dari *coefficient* sebagai penalti untuk *loss function*
- Menambahkan *sum of the coefficient values* (the L-1 norm) dan mengalikan dengan *constant lambda*.

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

→ Loss function Lasso

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

→ Loss function Linear Regression

L2 Regularization

- *Ridge regression*
- Ridge regression menambahkan “squared magnitude” dari *coefficient* sebagai penalti untuk *loss function*
- Menambahkan *sums the squares of coefficient values* (the L-2 norm) dan mengalikan dengan *constant lambda*.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

→ Loss function Ridge

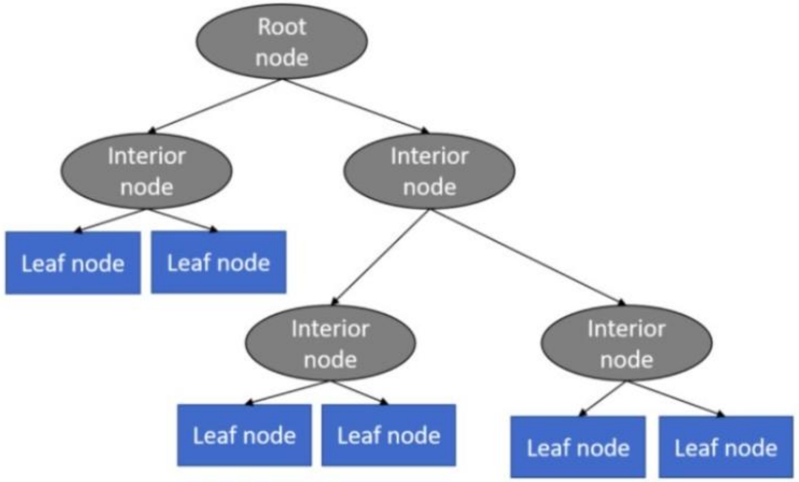
$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

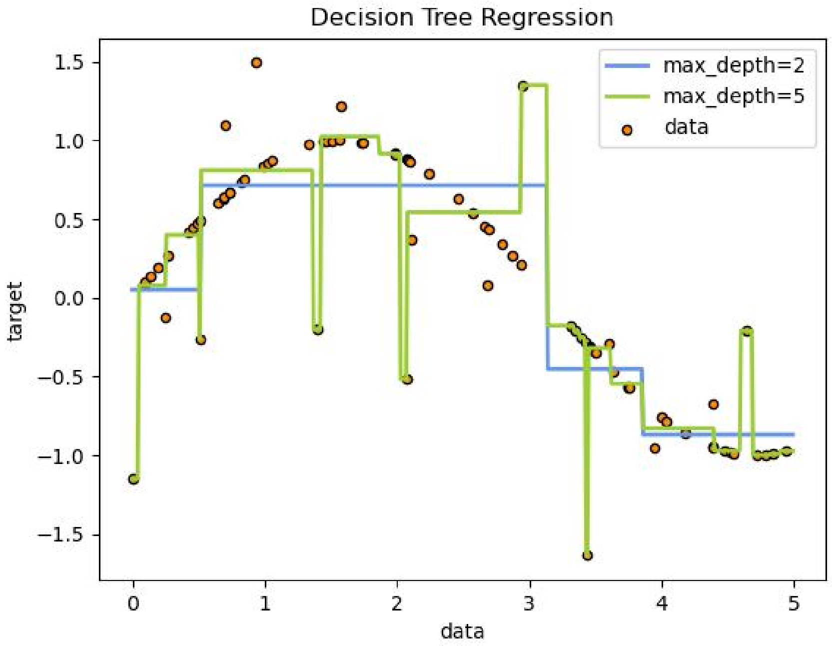
→ Loss function Linear Regression

DT and RF Regresion

Decision Tree Regression

- Decision trees bisa diaplikasikan pada kasus classification dan regression

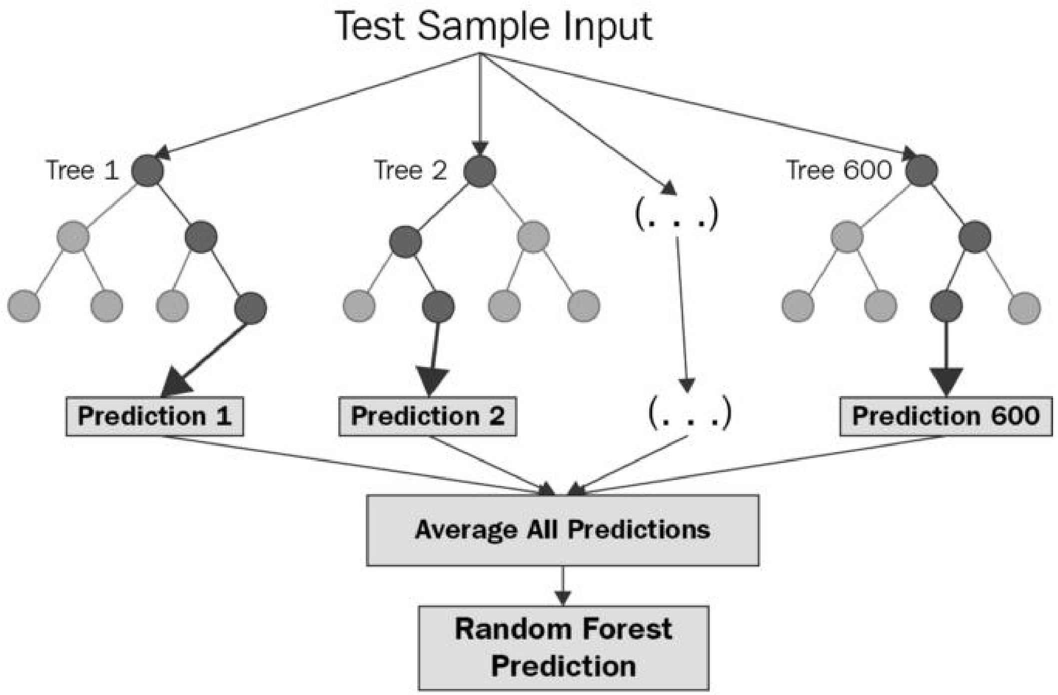


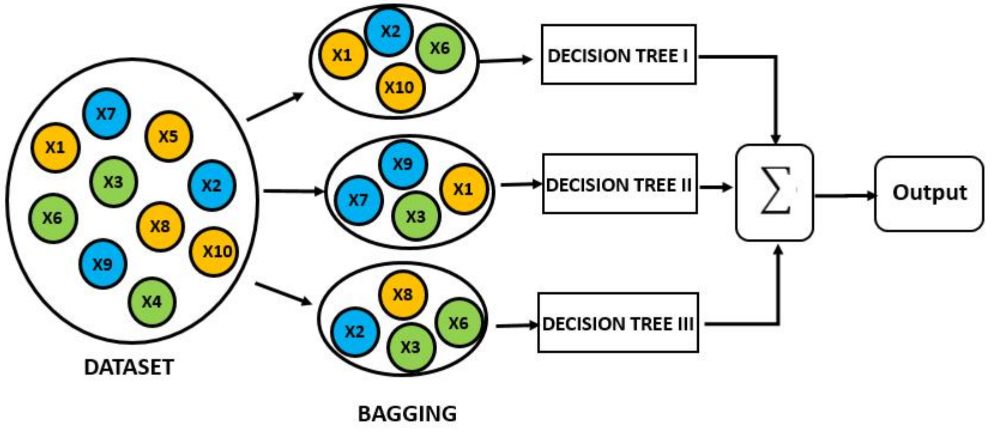


- Keuntungan
  - Mudah dipahami dan di-interpretasikan.
- Kerugian
  - Bisa membuat “over-complex trees” yang tidak bisa *generalise* terhadap data baru. Atau disebut dengan *overfitting*.
  - Solusi : *pruning*

Random Forest Regression

- Random forest adalah algoritma dalam Supervised Learning yang menggunakan *ensemble learning method* untuk kasus classification dan regression.
- Hasil prediksi adalah label terbanyak (untuk kasus classification) atau rata rata hasil prediksi (untuk kasus regression) dari model tree yang banyak.





Evaluation metrics for Regression

Evaluation metrics

- *Pearson correlation coefficient (r)* = mengukur kekuatan dan arah hubungan linier antara dua variabel (-1 to 1).
- *Coefficient determination (r^2 or r square)* = memberikan proporsi varians (fluktuasi) dari satu variabel yang diprediksi dari variabel lainnya (0 to 1).
- *Root mean square error (RMSE)* = merupakan besarnya tingkat kesalahan hasil prediksi. Semakin kecil (mendekati 0) semakin baik (*prediction errors*).

Performance Metric	Formula
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Pearson correlation coefficient (r)	$\frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)^2}}$
Coefficient determination (r^2)	$r^2 = [\text{Correlation Coefficient}]^2$

Performance comparison

- Hasil perbandingan dari model regresi yang diaplikasikan pada *house price dataset*

Model	RMSE	r2
Linear regression	208296	0.69
Lasso	208297	0.69
Ridge	208297	0.69
DT regression	192962	0.74
RF regression	144539	0.85

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU