Pertemuan 7: Machine Learning Supervised Learning for Regression

In [25]:

```
import numpy as np
import pandas as pd
```

IMPORT DATASET

In [26]:

```
data = "https://raw.githubusercontent.com/stedy/Machine-Learning-with-R-datasets/master/
df = pd.read_csv(data)
```

In [27]:

```
df
```

Out[27]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [28]:

```
# Check for missing values
print("Missing values in the dataset:")
print(df.isnull().sum())
```

In [29]:

```
# Convert categorical variables to numerical
df = pd.get_dummies(df, columns=['sex','smoker', 'region'])
```

In [30]:

```
# Memisahkan fitur dan target
X = df.drop(columns=['charges'])
y = df['charges']
```

Train Test Split

In [31]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import mean_absolute_error
```

In [32]:

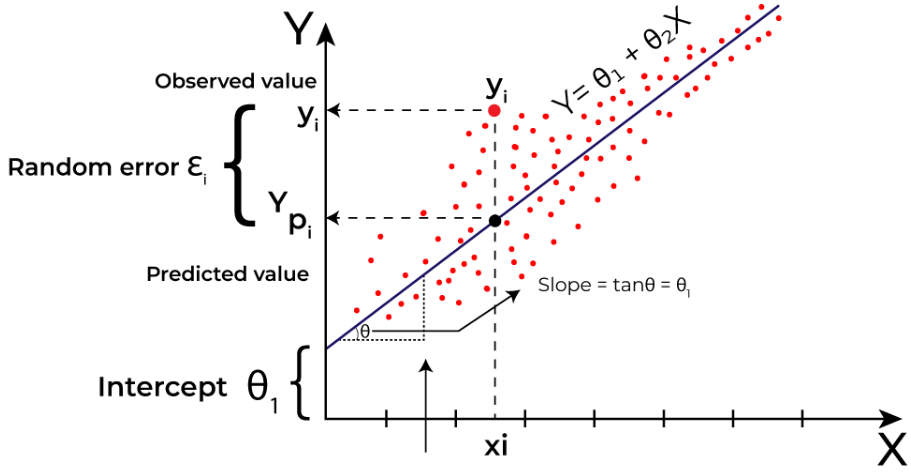
```
# Membagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

Evaluation Function

In [33]:

```python
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

def train_and_evaluate_regression(model, X_train, X_test, y_train, y_test):
    # Melatih model dengan data latih
    model.fit(X_train, y_train)

    # Melakukan prediksi menggunakan data uji
    y_pred = model.predict(X_test)

    # Menghitung Mean Absolute Error (MAE)
    mae = mean_absolute_error(y_test, y_pred)

    # Menghitung Mean Squared Error (MSE)
    mse = mean_squared_error(y_test, y_pred)

    # Menghitung Root Mean Squared Error (RMSE)
    rmse = mean_squared_error(y_test, y_pred, squared=False)

    # Menghitung R-squared (Koefisien Determinasi)
    r2 = r2_score(y_test, y_pred)

    # Menyusun hasil evaluasi
    result = {'Mean Absolute Error': mae, 'Mean Squared Error': mse, 'Root Mean Squared

    # Menampilkan hasil evaluasi
    print("Mean Absolute Error:", mae)
    print("Mean Squared Error:", mse)
    print("Root Mean Squared Error:", rmse)
    print("R-squared:", r2)

    return y_pred, result
```

Linear Regression



Y
Observed value
y_i
Random error ε_i
Y_{p_i}
Predicted value
Intercept θ_1
y_i
$Y = \theta_1 + \theta_2 X$
Slope = tanθ = θ_1
θ
xi
X

Regresi linearr adalah salah satu model statistik yang paling sederhana dan paling banyak digunakan. Hal ini mengasumsikan adanya hubungan linearr antara variabel independen dan dependen. Artinya perubahan variabel terikat sebanding dengan perubahan variabel bebas.

Persamaan yang menjelaskan bagaimana keterkaitan antara variabel X dengan variabel Y dan suatu model error disebut model regresi. Model regresi yang digunakan dalam regresi linear sederhana adalah:

$$Y= b0+b1*x$$

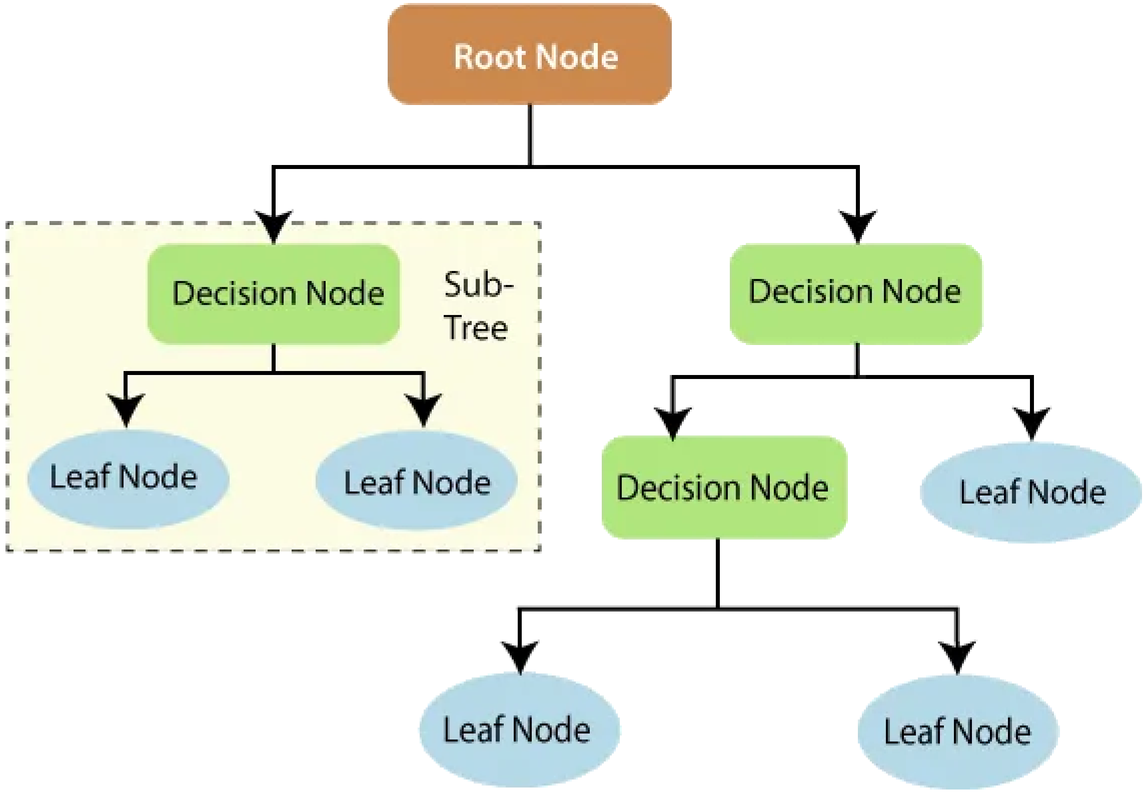
dimana b0 dan b1 menyatakan parameter model, X merupakan variabel independen.

In [34]:

```python
from sklearn.linear_model import LinearRegression
model = LinearRegression()
y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train, y_test)
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

Decision Tree



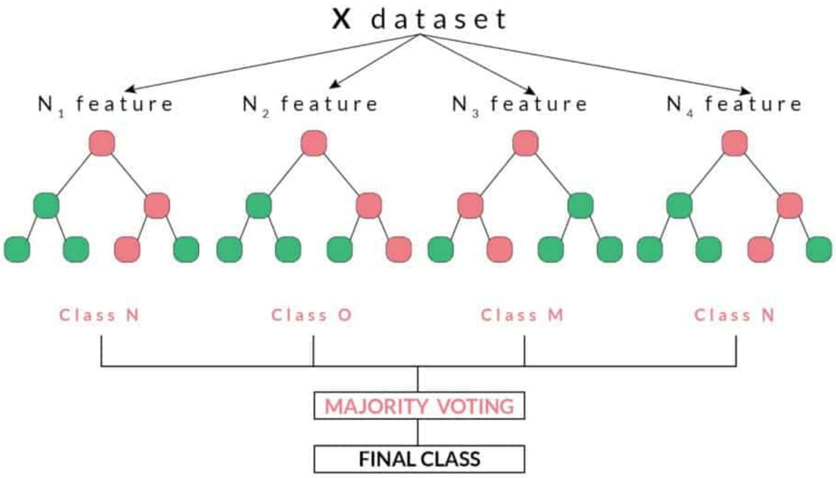
Regresi Decision Tree adalah jenis algoritma regresi yang membangun pohon keputusan untuk memprediksi nilai target. Decision Tree adalah struktur mirip pohon yang terdiri dari simpul dan cabang. Setiap node mewakili sebuah keputusan, dan setiap cabang mewakili hasil dari keputusan tersebut. Tujuan dari regresi pohon keputusan adalah untuk membangun pohon yang dapat secara akurat memprediksi nilai target untuk titik data baru.

In [35]:

```python
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor(random_state=12)
y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train, y_test)
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

Random Forest



Regresi random forest adalah metode ensemble yang menggabungkan beberapa pohon keputusan untuk memprediksi nilai target. Metode ensemble adalah jenis algoritme pembelajaran mesin yang menggabungkan beberapa model untuk meningkatkan performa model secara keseluruhan. Regresi random forest bekerja dengan membangun sejumlah besar pohon keputusan, yang masing-masing pohon dilatih pada subset data pelatihan yang berbeda. Prediksi akhir dibuat dengan merata-ratakan prediksi seluruh pohon.

In [36]:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=10)
y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train, y_test)
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

SVR

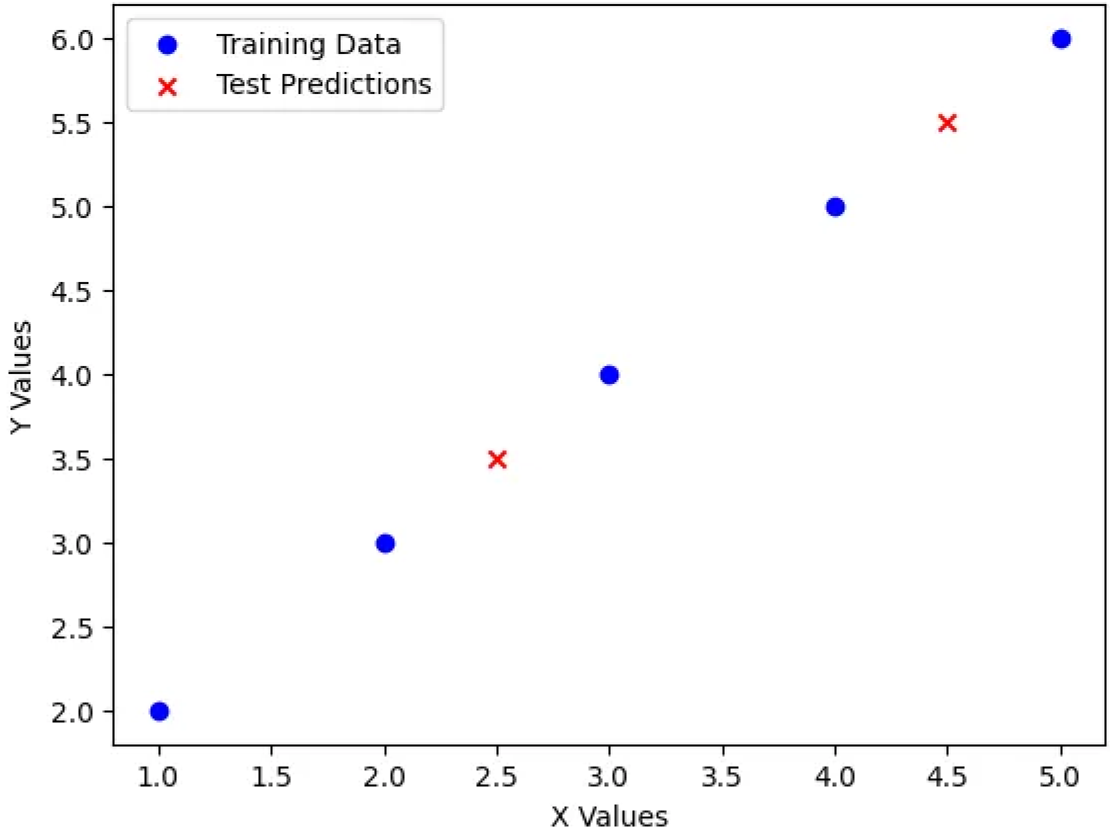
Support Vector Regression (SVR) adalah jenis algoritma regresi yang didasarkan pada algoritma support vector machine (SVM). SVM adalah jenis algoritma yang digunakan untuk tugas klasifikasi, tetapi juga dapat digunakan untuk tugas regresi. SVR bekerja dengan mencari hyperplane yang meminimalkan jumlah sisa kuadrat antara nilai prediksi dan nilai aktual.

In [37]:

```
from sklearn.svm import SVR
model = SVR()
y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train, y_test)
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

KNN



K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin non-parametrik yang dapat digunakan untuk tugas klasifikasi maupun regresi. Dalam konteks regresi, KNN sering disebut sebagai "Regresi KNN." Ini adalah algoritma yang sederhana dan intuitif yang membuat prediksi dengan mencari K titik data terdekat dari input yang diberikan dan melakukan rata-rata dari nilai target mereka.

In [38]:

```
from sklearn.neighbors import KNeighborsRegressor
model = KNeighborsRegressor(n_neighbors=3)
y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train, y_test)
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

All Model

In [39]:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor

def auto_model(X_train, y_train, X_test, y_test):
    # Definisi model-model yang akan digunakan
    models = [
        ('Linear Regression', LinearRegression()),
        ('Support Vector Machine (SVM) Regression', SVR()),
        ('Decision Tree Regression', DecisionTreeRegressor(random_state=12)),
        ('Random Forest Regression', RandomForestRegressor(n_estimators=100, random_stat
        ('K-Nearest Neighbors (KNN) Regression', KNeighborsRegressor(n_neighbors=3))
    ]
```

```python
    # Inisialisasi tabel untuk menyimpan hasil evaluasi
    table = {
        'Model': [],
        'Mean Absolute Error': [],
        'Mean Squared Error': [],
        'Root Mean Squared Error': [],
        'R-squared': []
    }

    # Latih dan evaluasi setiap model
    for name, model in models:
        y_pred, result = train_and_evaluate_regression(model, X_train, X_test, y_train,
        table['Model'].append(name)
        table['Mean Absolute Error'].append(result['Mean Absolute Error'])
        table['Mean Squared Error'].append(result['Mean Squared Error'])
        table['Root Mean Squared Error'].append(result['Root Mean Squared Error'])
        table['R-squared'].append(result['R-squared'])

    # Konversi ke DataFrame
    hasil = pd.DataFrame(table)

    return hasil
```

In [41]:

```python
# Panggil fungsi auto_model dengan X_train, X_test, y_train, y_test
hasil_evaluasi = auto_model(X_train, y_train, X_test, y_test);
```

```
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
C:\Users\tsigi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0
\LocalCache\local-packages\Python311\site-packages\sklearn\metrics\_regression.py:483: F
utureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calc
ulate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

In [42]:

```python
hasil_evaluasi
```

Out[42]:

	Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R-squared
0	Linear Regression	4181.194474	3.359692e+07	5796.284659	0.783593
1	Support Vector Machine (SVM) Regression	8598.964702	1.665022e+08	12903.571294	-0.072486
2	Decision Tree Regression	3345.766503	5.054786e+07	7109.701955	0.674407
3	Random Forest Regression	2530.195383	2.139321e+07	4625.279744	0.862200
4	K-Nearest Neighbors (KNN)	6285.787042	1.099411e+08	10485.282399	0.291839

Regression