Rakamin Academy

Virtual Internship Experience

KALBE Nutritionals

Supervised Learning - Regression

Disclaimer

Outline

1. Pengertian *Regression*
2. Cara Kerja
3. Studi Kasus

1 Regression

Regression merupakan algoritma machine learning yang tergolong ke dalam *supervised learning*, Regresi adalah metode statistik yang digunakan untuk menganalisis hubungan antara variabel dependen dan satu atau lebih variabel independen. Variabel dependen adalah variabel yang diprediksi, dan variabel independen adalah variabel yang digunakan untuk memprediksi variabel dependen.

❑ Algoritma Regresi

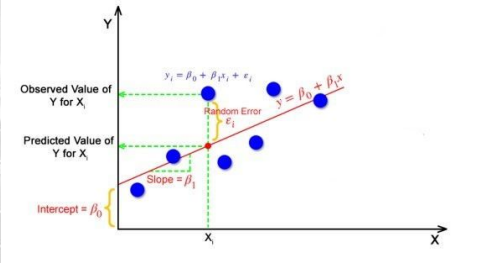
1. *Linear Regression*
2. *Logistic Regression*
3. *Support Vector Machines (SVM)*
4. Decision Trees
5. Random Forest

2 Cara Kerja

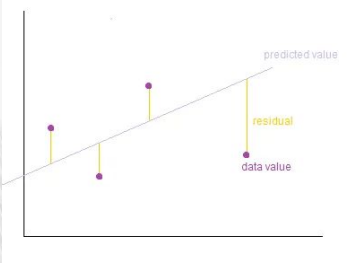
terdapat berbagai algoritma regresi, tetapi yang paling umum adalah regresi linier (Linear Regression). Regresi linier digunakan untuk memprediksi nilai kontinu dari sekumpulan variabel independen. Hal ini dilakukan dengan **menyesuaikan garis** pada data, **sehingga jumlah kesalahan kuadrat (*sum of squared errors*) antara nilai yang diprediksi dan nilai sebenarnya diminimalkan**. Persamaan untuk garis regresi linier adalah:

$$y = mx + b$$

Gambar 1. Persamaan Regresi



Gambar 2. Penjabaran notasi dari persamaan regresi linear



Gambar 3. penggambaran data aktual dengan data prediksi hasil regresi linear

y: nilai yang diprediksi,
m: kemiringan garis (gradien)
b: intersepsi y.

2 Cara Kerja

- **Menyesuaikan garis terhadap data**

Dalam Regresi Linier, *cost function Mean Squared Error* (MSE) digunakan untuk mencari garis yang paling mendekati dengan data. MSE merupakan persamaan rata-rata dari kesalahan kuadrat antara nilai yang diprediksi dan nilai sebenarnya

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

Gambar 4. Persamaan MSE menggunakan persamaan Regresi

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Gambar 5. Persamaan MSE

y_i : nilai aktual
$\hat{y}$: nilai prediksi
n : banyaknya data
Σ : penjumlahan dari seluruh data

Secara singkat, MSE merupakan perhitungan selisih dari nilai aktual data dengan nilai prediksi (yang diperoleh dengan kalkulasi regresi linear yakni y = mx+b) dan untuk dilakukan penjumlahan untuk setiap selisih antara data aktual dengan data prediksi, dan dibagi dengan banyaknya data. MSE yang rendah menunjukkan error yang sedikit, sehingga MSE yang rendah merupakan tujuan dari regresi linear yakni garis yang sesuai dengan data

3 Studi Kasus

Anda memperoleh data penjualan beberapa produk kalbe selama 1 bulan terakhir serta total penjualan keseluruhan harian selama 1 bulan. Tim bisnis ingin membuat suatu promo diskon terhadap salah satu produk, namun terkendala akan kekhawatiran produk tersebut pada hari berikutnya akan mengalami penurunan. Sebagai seorang Data Scientist, anda ingin membuat suatu model yang dapat memprediksi penjualan produk pada hari berikutnya dan memberikan hasil tersebut kepada tim bisnis

Petunjuk Pengerjaan:

1. Anda dapat mengerjakan menggunakan *jupyter notebook* atau *google colab*
2. Dapat melakukan pra pemrosesan terlebih dahulu, baik dari data cleaning hingga data transformation
3. Pastikan semua data sudah menggunakan tipe data yang tepat.
4. Selamat mencoba

Link Dataset

Solusi

- Memanggil Library yang diperlukan
- Load dataset
- mengecek informasi dataset
- membuat scatterplot untuk melihat relasi antara dua variabel

```
#Library untuk pengolahan data
import pandas as pd
import numpy as np

#Library untuk visualisasi data
import matplotlib.pyplot as plt
import seaborn as sns

#Library untuk model Linear Regression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# load data dan menampilkan data
df = pd.read_csv('Dataset CCV 5 - Supervised Learning Regression.csv')
df.head()
```

	Day	Diabetasol Coklat 1 kg	Fitbar Multigrain raisin 22 gram	Hydro Coco Original 500 ml	Slim & Fit Milk 312 gr	Daily Sales
0	1	7165158.0	115718.0	747417	7131950.0	20100380
1	2	11091106.0	302774.0	531021	4568946.0	19923983
2	3	3927138.0	150366.0	66812	2321753.0	10688946
3	4	1155500.0	74752.0	436916	NaN	4993264
4	5	6007919.0	370528.0	202243	3764485.0	14853006

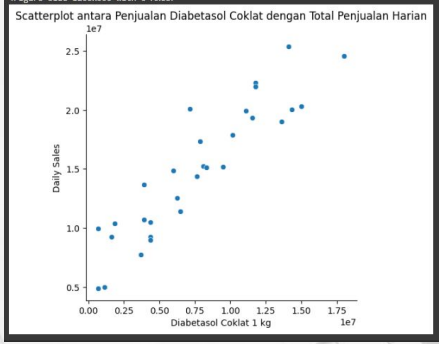
Gambar 1. Load Library dan dataset

```
# mendapatkan informasi data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 6 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Day                              31 non-null     int64
 1   Diabetasol Coklat 1 kg           30 non-null     float64
 2   Fitbar Multigrain raisin 22 gram 30 non-null     float64
 3   Hydro Coco Original 500 ml       31 non-null     int64
 4   Slim & Fit Milk 312 gr           28 non-null     float64
 5   Daily Sales                      31 non-null     int64
dtypes: float64(3), int64(3)
memory usage: 1.6 KB
```

Gambar 2. Mendapatkan informasi data

```
# membuat visualisasi sebaran data
# antara variabel dependen salary
# dengan variabel independen years
# menggunakan grafik scatterplot
plt.figure(figsize=(12,5))
sns.pairplot(df,x_vars=['Diabetasol Coklat 1 kg'],y_vars=['Daily Sales'],height=5,kind='scatter')
plt.xlabel('Diabetasol Coklat 1 kg')
plt.ylabel('Daily Sales')
plt.title('Scatterplot antara Penjualan Diabetasol Coklat dengan Total Penjualan Harian')
plt.show()
```



Gambar 3. Melakukan visualisasi dengan scatterplot

Solusi

- Melakukan drop data dengan *missing values* pada kolom Diabetasol Coklat 1 kg
- Menyimpan kolom total penjualan Diabetasol Coklat 1 Kg ke variabel x sebagai variabel independen
- Menyimpan kolom harga total penjualan harian ke variabel y sebagai variabel dependen
- Membagi data menjadi data latih dan data uji untuk pembangunan model dan melatih model
- Membangun model dan melakukan fit model ke data latih

```
df.dropna(subset = 'Diabetasol Coklat 1 kg', inplace = True)

X = df['Diabetasol Coklat 1 kg']
X.head()

0     7165158.0
1    11091106.0
2     3927138.0
3     1155500.0
4     6007919.0
Name: Diabetasol Coklat 1 kg, dtype: float64

y = df['Daily Sales']
y.head()

0    20100380
1    19923983
2    10688946
3     4993264
4    14853006
Name: Daily Sales, dtype: int64

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.7,random_state=100)

X_train = X_train.to_numpy()
X_test = X_test.to_numpy()

X_train = X_train.reshape(X_train.shape[0],1)
X_test = X_test.reshape(X_test.shape[0],1)
```

Gambar 5. Membagi data menjadi data latih dan data uji

```
# menyesuaikan garis regresi dengan sebaran data (fit)
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)

▸ LinearRegression

# model melakukan prediksi
y_pred = lr.predict(X_test)
```

Gambar 6. Membangun model

Solusi

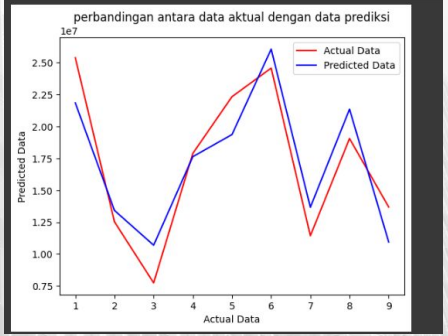
- Melakukan prediksi pada data uji
- Membuat visualisasi untuk melihat perbandingan antara data aktual dengan data prediksi
- Menghitung persamaan MSE untuk melihat performa model
- Menampilkan gradien dan intersepsi
- berhasil membuat model dan mendapatkan persamaan dalam membangun model

```
# menyesuaikan garis regresi dengan sebaran data (fit)
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
```

```
LinearRegression
```

```
# model melakukan prediksi
y_pred = lr.predict(X_test)
```

```
c = [i for i in range (1,len(y_test)+1,1)]
plt.plot(c,y_test,color='r',linestyle='-', label = 'Actual Data')
plt.plot(c,y_pred,color='b',linestyle='-', label = 'Predicted Data')
plt.xlabel('Actual Data')
plt.ylabel('Predicted Data')
plt.legend()
plt.title('perbandingan antara data aktual dengan data prediksi')
plt.show()
```



Gambar 7. Melakukan prediksi dan melihat perbandingan antara data

```
# metriks untuk evaluasi model
from sklearn.metrics import r2_score,mean_squared_error
```

```
[106] # perhitungan Mean square error
mse = mean_squared_error(y_test,y_pred)
print('mean squared error',mse)
```

```
mean squared error 5646943705294.958
```

```
[107] # Intecept and coeff of the line
print('Intercept of the model:',lr.intercept_)
print('Coefficient of the line:',lr.coef_)
```

```
Intercept of the model: 6728389.623898853
Coefficient of the line: [1.0712553]
```

kita berhasil membuat model regresi linear dengan persamaan:

$$y = 1.07x + 6728389.62$$

Gambar 8. Menghitung MSE dan mendapatkan intersepsi serta gradien model


Thank You!
KALBE
Nutritionals