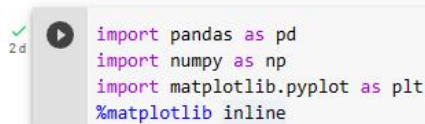


Tugas 1 (Regresi Linear)

Herlina Anwar - D082222026

Penjelasan Kode

▼ Import library



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Langkah pertama yang dilakukan yaitu meng-import library yang dibutuhkan seperti gambar diatas.

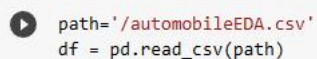
Import pandas as pd . Library Pandas merupakan library open source dalam bahasa pemrograman python yang berfungsi untuk memproses data, mulai dari pembersihan data, manipulasi data, hingga melakukan analisis data.

Import numpy as np. Library Numpy (numerical Python) merupakan library python yang menyediakan fungsi yang siap dipakai untuk memudahkan kita melakukan perhitungan sains seperti matrix, aljabar, statistik dan sebagainya.

Import matplotlib.pyplot as plt. Matplotlib digunakan untuk melakukan visualisasi data secara 2D ataupun 3D yang dapat disimpan dengan format gambar seperti JPEG, JPG, dan PNG.

%matplotlib inline digunakan untuk meng-embed gambar plot statis didalam progra.

▼ Membaca data .csv



```
path = '/automobileEDA.csv'
df = pd.read_csv(path)
```

Setelah itu, melakukan proses membaca file (*read*) berformat .csv yang berisi data. Yang disimpan pada direktori. Proses pembacaan file .csv dideklarasikan sebagai *df*.

Menampilkan contoh data yang paling atas

```
df.head()
```

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	...
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27	13
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27	16
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...	9.0	154.0	5000.0	19	26	16
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...	10.0	102.0	5500.0	24	30	13
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...	8.0	115.0	5500.0	18	22	17

5 rows x 29 columns

Kemudian menggunakan fungsi **head()** yang merupakan fungsi untuk mendapatkan data dari batas teratas seperti gambar diatas.

Menampilkan jumlah data

```
len(df)
```

201

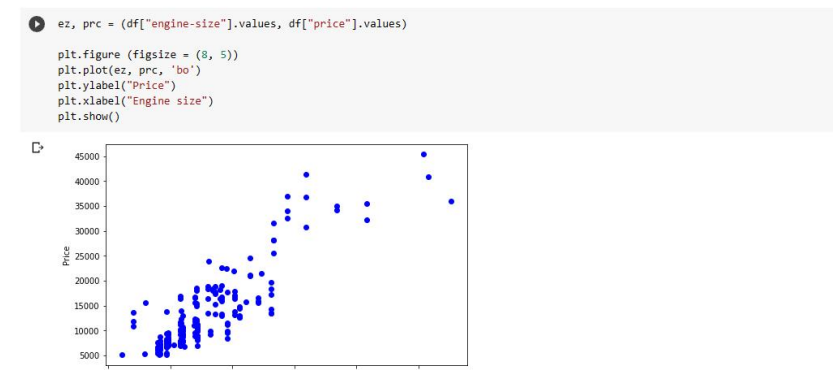
Menampilkan judul tabel (fitur) yang ada di file .csv

```
print(df.columns.values)
```

['symboling' 'normalized-losses' 'make' 'aspiration' 'num-of-doors' 'body-style' 'drive-wheels' 'engine-location' 'wheel-base' 'length' 'width' 'height' 'curb-weight' 'engine-type' 'num-of-cylinders' 'engine-size' 'fuel-system' 'bore' 'stroke' 'compression-ratio' 'horsepower' 'peak-rpm' 'city-mpg' 'highway-mpg' 'price' 'city-L/100km' 'horsepower-binned' 'diesel' 'gas']

Selanjutnya menampilkan jumlah data dengan menggunakan fungsi **len()**, selanjutnya akan keluar informasi jumlah data yaitu 201. kemudian menampilkan judul tabel pada file .csv yang diproses dengan menggunakan fungsi **print()**. setelah itu melakukan eksplorasi data dengan menampilkan plotting atribut engine-size dan price, seperti gambar dibawah.

Eksplorasi data dengan menampilkan plotting attribute 'engine-size' dan 'price'



▼ Import library Linear Regression

```
[ ] from sklearn.linear_model import LinearRegression

[ ] # Mendeklarasikan library Linear Regression sebagai 'lm'
    lm = LinearRegression()
```

Selanjutnya memasukkan *library* Linear Regression dengan menggunakan module **sklearn** yang berfungsi untuk membantu melakukan processing data ataupun melakukan training data untuk kebutuhan pemrograman python, yang kemudian fungsi linear regression dideklarasikan dengan **lm**.

▼ Menghitung Intercept (a) dan Slope (b), pada tabel 'Ukuran Mesin' (engine-size) dan 'Harga' (price)

```
# y = a + bx
# menghitung a (intercept) dan b (slope)

x_par = df[['engine-size']]
y_par = df['price']

lm.fit(x_par, y_par)
a = lm.intercept_
b = lm.coef_
print (a,b)
```

[-7963.338906281042 [166.86001569]

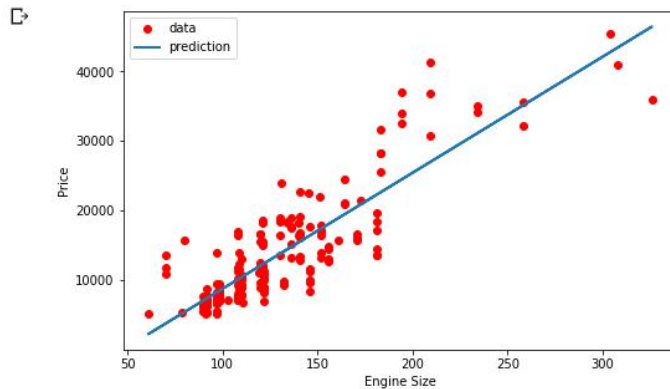
Activ
Print

Berikutnya melakukan perhitungan intercept(a) dan Slope (b) pada tabel dengan mendeklarasikan engine-size (**x_par**) dan price (**y_par**). kemudian mengimplementasikan model linear regression dengan persamaan **y_{model} = a+(b*x)**. Setelah itu menghitung nilai prediksi model_linear seperti pada gambar dibawah.

```
# Membuat model persamaan linear y = a + bx
def model_linear(x, a, b) :
    ymodel = a + (b * x)
    return ymodel
```

```
[ ] # Menghitung nilai y_prediksi
    y_prediksi = model_linear(x_par, a, b)
```

```
# Melakukan plotting data dan prediksi
plt.figure(figsize=(8,5))
plt.plot(ez, prc, 'ro', label='data')
plt.plot(ez, y_prediksi, linewidth=2.0, label='prediction')
plt.legend(loc='best')
plt.ylabel('Price')
plt.xlabel('Engine Size')
plt.show()
```



Langkah selanjutnya adalah melakukan plotting data dan prediksi, dengan **ylabel** (price) dan **xlabel**(engine size). kemudian menggunakan fungsi **show()** untuk menampilkan hasil regresi linear seperti gambar diatas, yang dimana titik berwarna merah adalah data dan garis biru adalah prediksi.

```
[ ] # Evaluasi akurasi model regresi
# Menentukan data latih dan data uji secara acak (Split Test)
i = np.random.rand(len(df)) < 0.8
x_latih = x_par[i]
y_latih = prc[i]
x_uji = ez[~i]
y_uji = prc[~i]
```

```
[ ] # Menghitung a (intercept) dan b (slope) dari data latih
lm.fit(x_latih, y_latih)
a_latih = lm.intercept_
b_latih = lm.coef_
print (a_latih, b_latih)
```

```
-7554.154161310411 [163.64676321]
```

Selanjutnya melakukan evaluasi terhadap model linear regression dengan menentukan data latih dan data uji secara acak dengan menggunakan fungsi **np.random.rand()**, kemudian menghitung intercept(a) dan slope(b) dari data latih, dan melakukan pemanggilan dengan fungsi **print()**

```
[ ] # Menghitung prediksi dari data uji
y_prediksi = model_linear(x_uji, a_latih, b_latih)

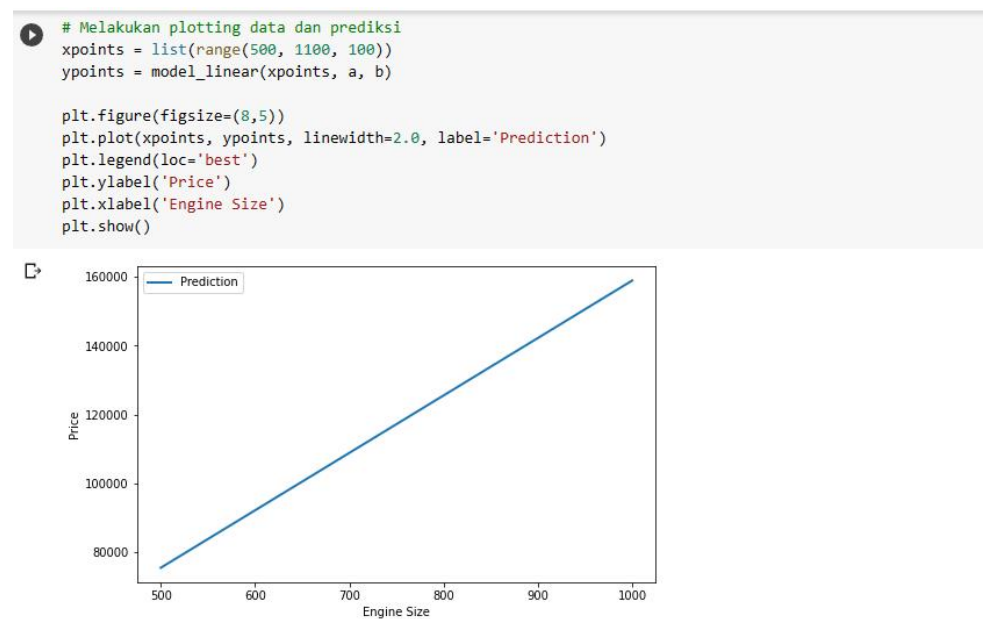
#menghitung MAE, MSE, dan RMSE
mae = np.mean(np.absolute(y_uji - y_prediksi))
mse = np.mean((y_uji - y_prediksi)**2)
rmse = np.sqrt(mse)

print("Mean Absolute Error = " + str(round(mae, 4)))
print("Mean Squared Error = " + str(round(mse, 4)))
print("Root Mean Squared Error = " + str(round(rmse, 4)))

Mean Absolute Error = 2546.3629
Mean Squared Error = 13139668.0872
Root Mean Squared Error = 3624.868
```

Selanjutnya menghitung nilai prediksi dari data uji kemudian mengimplementasikan model regresi linear dengan **x_uji** merupakan nilai dari variable *predictor* atau konstanta, **a_latih** merupakan nilai *intercept*, dan **b_latih** merupakan nilai dari *slope*. Sehingga mengubah nilai dari **y_prediksi**, yang dideklarasikan sebagai variabel *response*.

Setelah itu, menghitung nilai error menggunakan MAE, MSE, dan RMSE dengan menggunakan rumus masing masing seperti gambar diatas. Selanjutnya menggunakan fungsi **print()** untuk menampilkan hasil dari masing masing nilai.



Langkah terakhir adalah melakukan plotting data dan prediksi, dengan data yang dispesifikasikan sebagai *range(start, stop, step)*. Tiga buah parameter range tersebut, yaitu:

- *start* merupakan batas awal
- *stop* merupakan batas akhir
- *step* merupakan jumlah naik atau turun antar bilangan.

dijabarkan bawah variabel *predictor* dideklarasikan sebagai **xpoints**. Proses *slicing data* dilakukan dengan ketentuan **500** sebagai batas awal (*start*), **1100**

sebagai batas akhir (*stop*), dan **100** sebagai jumlah naik atau turun antar bilangan (*step*).

Selanjutnya mendefinisikan pada plot untuk engine size(*x_label*) dan price(*y_label*). kemudian menampilkan grafik dengan menggunakan fungsi **show()** dan menampilkan hasil prediksi seperti grafik diatas.