ACTIVITY PERTEMUAN 1

NAMA : MUHAMMAD TARMIDZI BARIQ

NPM : 51422161

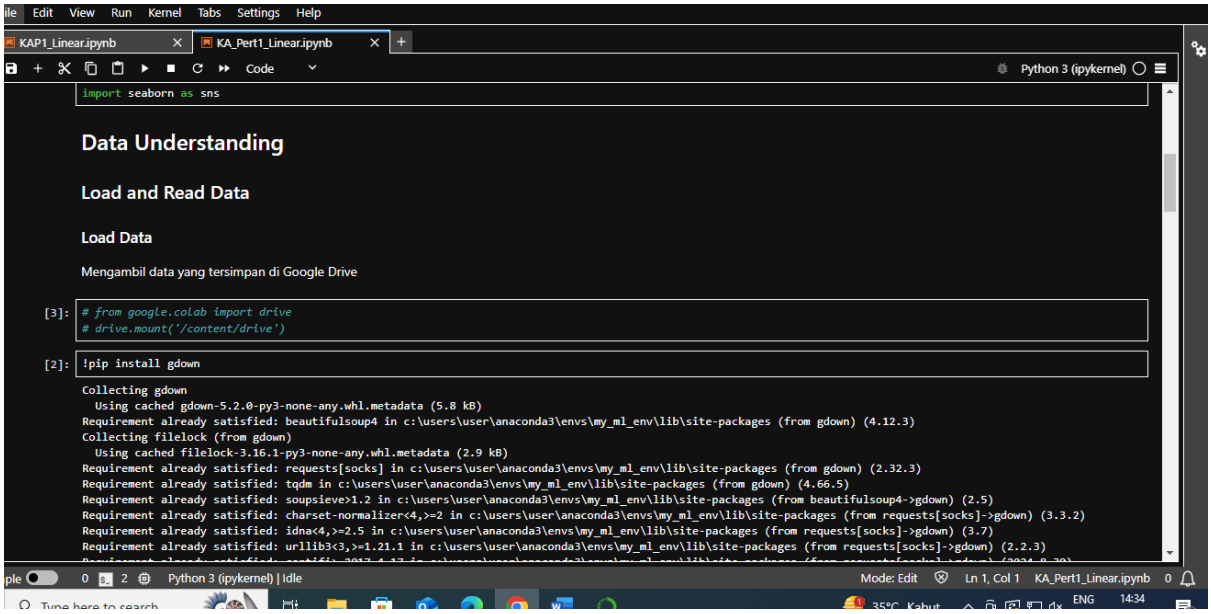
KELAS : 3IA11

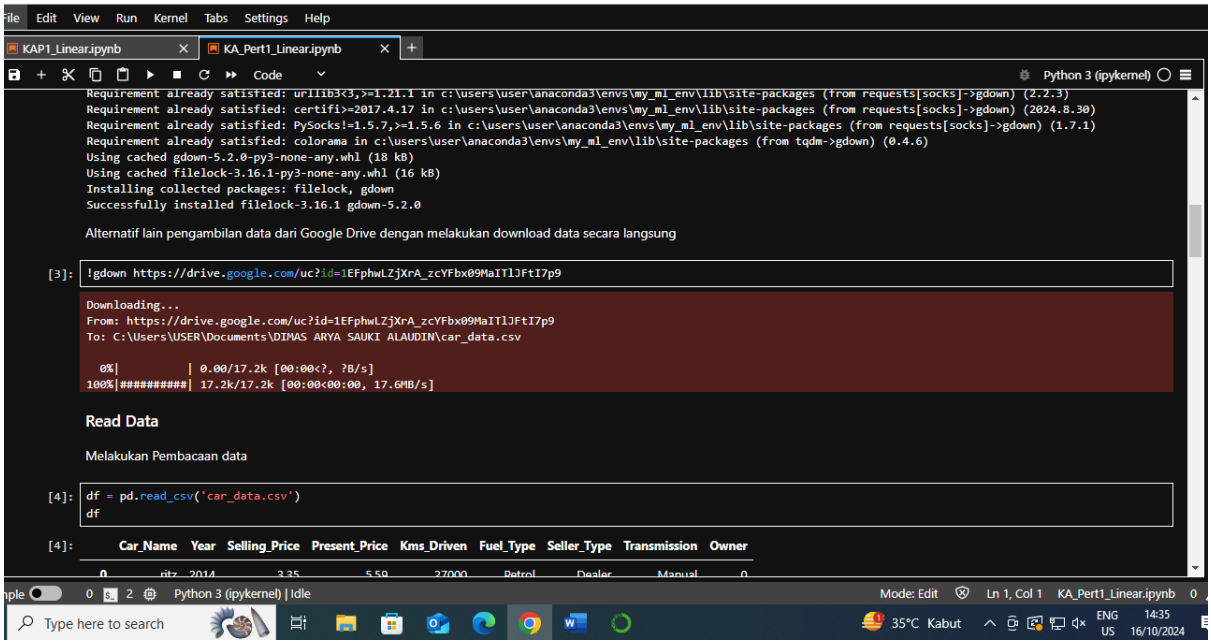
MATERI : PENGENALAN KECERDASAN ARTIFICIAL

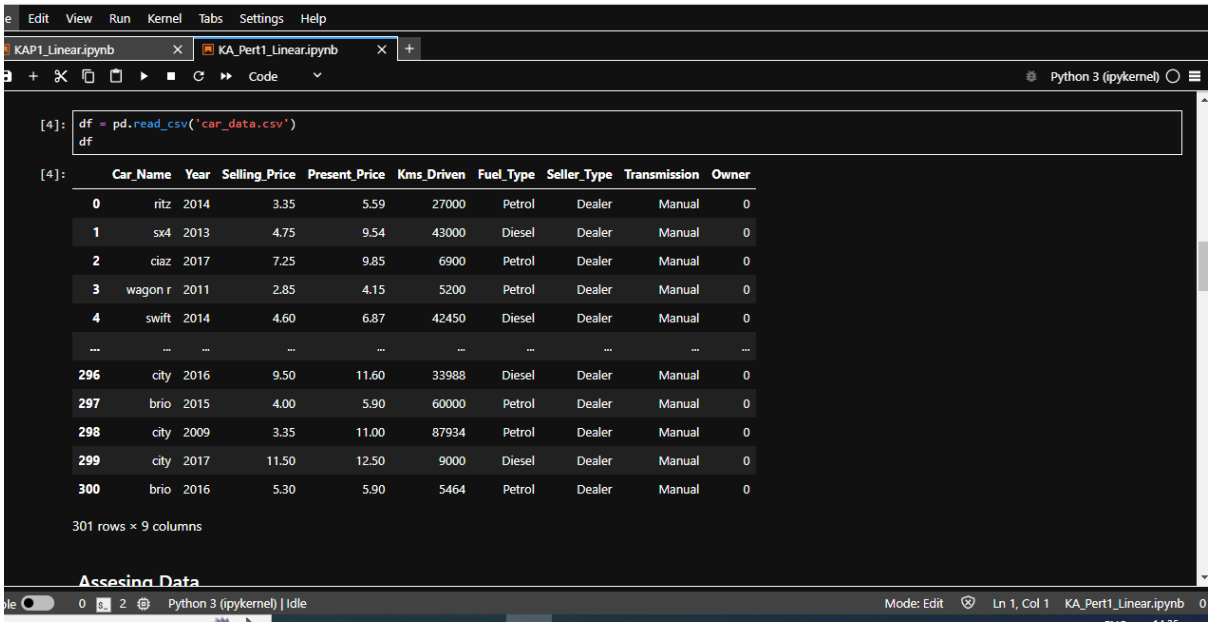
MATA PRAKTIKUM : KECERDASAN ARTIFICIAL

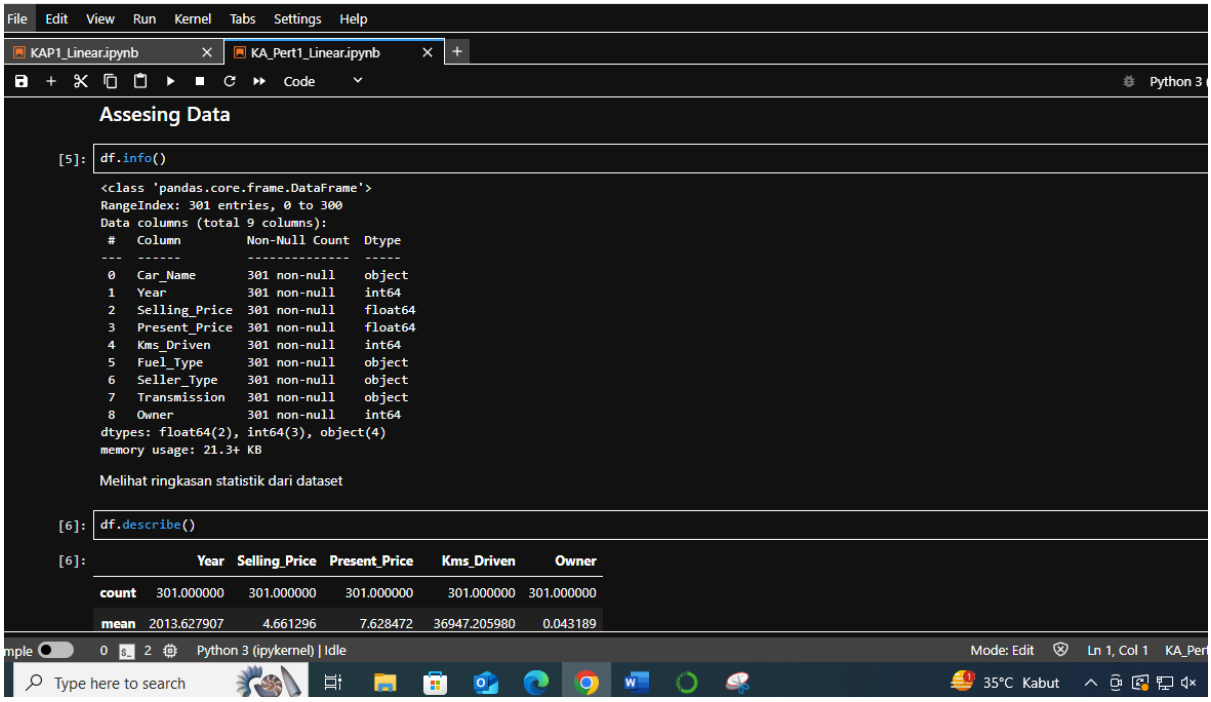
---

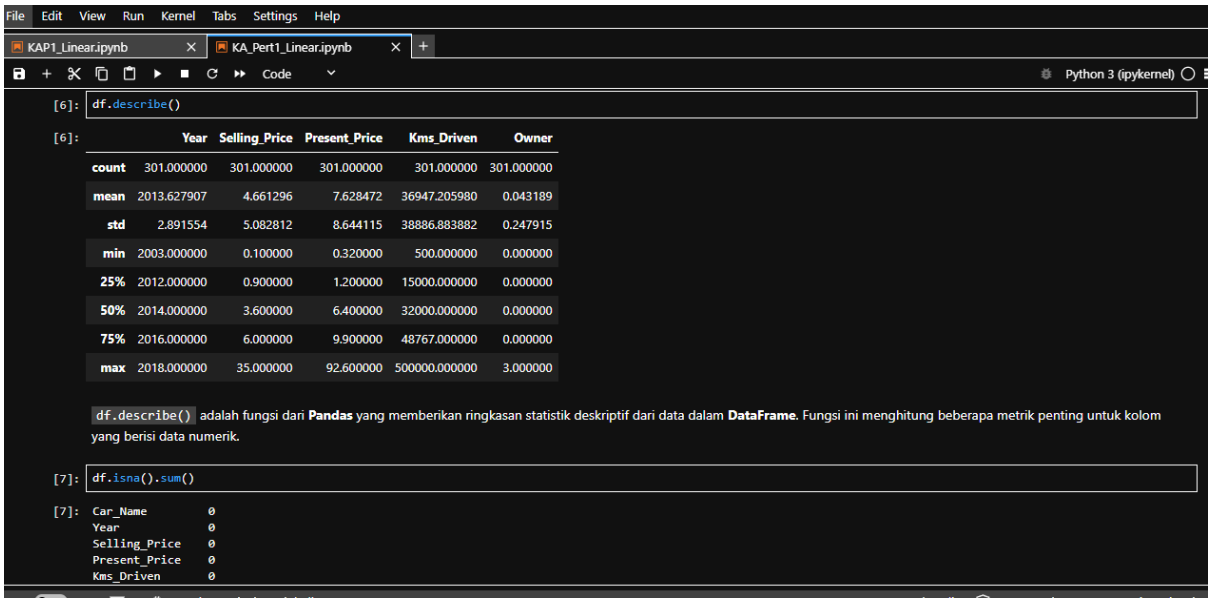
2. memprediksi harga mobil menggunakan data yang sudah tersedia di file excel Dikecerdasan artificial supervised learning




File Edit View Run Kernel Tabs Settings Help
KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3 (ipykernel)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\user\anaconda3\envs\my_ml_env\lib\site-packages (from requests[socks]->gdown) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\user\anaconda3\envs\my_ml_env\lib\site-packages (from requests[socks]->gdown) (2024.8.30)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in c:\users\user\anaconda3\envs\my_ml_env\lib\site-packages (from requests[socks]->gdown) (1.7.1)
Requirement already satisfied: colorama in c:\users\user\anaconda3\envs\my_ml_env\lib\site-packages (from tqdm->gdown) (0.4.6)
Using cached gdown-5.2.0-py3-none-any.whl (18 kB)
Using cached filelock-3.16.1-py3-none-any.whl (16 kB)
Installing collected packages: filelock, gdown
Successfully installed filelock-3.16.1 gdown-5.2.0
Alternatif lain pengambilan data dari Google Drive dengan melakukan download data secara langsung
[3]: !gdown https://drive.google.com/uc?id=1EFphwLZjXrA_zcYFbx09MaITlJFtI7p9
Downloading...
From: https://drive.google.com/uc?id=1EFphwLZjXrA_zcYFbx09MaITlJFtI7p9
To: C:\Users\USER\Documents\DIMAS ARYA SAUKI ALAUDIN\car_data.csv
0%| | 0.00/17.2k [00:00<?, ?B/s]
100%|##########| 17.2k/17.2k [00:00<00:00, 17.6MB/s]
Read Data
Melakukan Pembacaan data
[4]: df = pd.read_csv('car_data.csv')
df
[4]: Car_Name Year Selling_Price Present_Price Kms_Driven Fuel_Type Seller_Type Transmission Owner
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 1, Col 1
KA_Pert1_Linear.ipynb
Type here to search
35°C Kabut
ENG US
14:35
16/10/2024


Edit View Run Kernel Tabs Settings Help
KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3 (ipykernel)
[4]: df = pd.read_csv('car_data.csv')
df
[4]:
Car_Name Year Selling_Price Present_Price Kms_Driven Fuel_Type Seller_Type Transmission Owner
0 ritz 2014 3.35 5.59 27000 Petrol Dealer Manual 0
1 sx4 2013 4.75 9.54 43000 Diesel Dealer Manual 0
2 ciaz 2017 7.25 9.85 6900 Petrol Dealer Manual 0
3 wagon r 2011 2.85 4.15 5200 Petrol Dealer Manual 0
4 swift 2014 4.60 6.87 42450 Diesel Dealer Manual 0
... ...
296 city 2016 9.50 11.60 33988 Diesel Dealer Manual 0
297 brio 2015 4.00 5.90 60000 Petrol Dealer Manual 0
298 city 2009 3.35 11.00 87934 Petrol Dealer Manual 0
299 city 2017 11.50 12.50 9000 Diesel Dealer Manual 0
300 brio 2016 5.30 5.90 5464 Petrol Dealer Manual 0
301 rows × 9 columns
Assesing Data
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 1, Col 1
KA_Pert1_Linear.ipynb


File Edit View Run Kernel Tabs Settings Help
KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3
Assesing Data
[5]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Car_Name 301 non-null object
1 Year 301 non-null int64
2 Selling_Price 301 non-null float64
3 Present_Price 301 non-null float64
4 Kms_Driven 301 non-null int64
5 Fuel_Type 301 non-null object
6 Seller_Type 301 non-null object
7 Transmission 301 non-null object
8 Owner 301 non-null int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
Melihat ringkasan statistik dari dataset
[6]: df.describe()
[6]:
Year Selling_Price Present_Price Kms_Driven Owner
count 301.000000 301.000000 301.000000 301.000000 301.000000
mean 2013.627907 4.661296 7.628472 36947.205980 0.043189
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 1, Col 1
KA_Pert
Type here to search
35°C Kabut


File Edit View Run Kernel Tabs Settings Help
KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3 (ipykernel)
[6]: df.describe()
[6]:
Year Selling_Price Present_Price Kms_Driven Owner
count 301.000000 301.000000 301.000000 301.000000 301.000000
mean 2013.627907 4.661296 7.628472 36947.205980 0.043189
std 2.891554 5.082812 8.644115 38886.883882 0.247915
min 2003.000000 0.100000 0.320000 500.000000 0.000000
25% 2012.000000 0.900000 1.200000 15000.000000 0.000000
50% 2014.000000 3.600000 6.400000 32000.000000 0.000000
75% 2016.000000 6.000000 9.900000 48767.000000 0.000000
max 2018.000000 35.000000 92.600000 500000.000000 3.000000
df.describe() adalah fungsi dari Pandas yang memberikan ringkasan statistik deskriptif dari data dalam DataFrame. Fungsi ini menghitung beberapa metrik penting untuk kolom yang berisi data numerik.
[7]: df.isna().sum()
[7]: Car_Name 0
Year 0
Selling_Price 0
Present_Price 0
Kms_Driven 0

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code | Python 3 (ipykernel)

```
[7]: df.isna().sum()
```

```
[7]: Car_Name        0
     Year            0
     Selling_Price   0
     Present_Price   0
     Kms_Driven      0
     Fuel_Type       0
     Seller_Type     0
     Transmission    0
     Owner           0
     dtype: int64
```

`isna()` digunakan untuk melihat data yang bersifat Null. `.sum()` digunakan untuk menjumlah data yang ada.

Disini dapat dilihat bahwa data tidak memiliki data yang bernilai Null.

```
[8]: df.duplicated().sum()
```

```
[8]: 2
```

Kode ini digunakan untuk menjumlahkan duplikasi yang terdapat pada dataset. Disini dapat dilihat bahwa terdapat 2 data yang bersifat duplikat.

```
[9]: df.drop_duplicates(inplace=True)
```

Data Preparation

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code

[9]: df.drop_duplicates(inplace=True)

Data Preparation

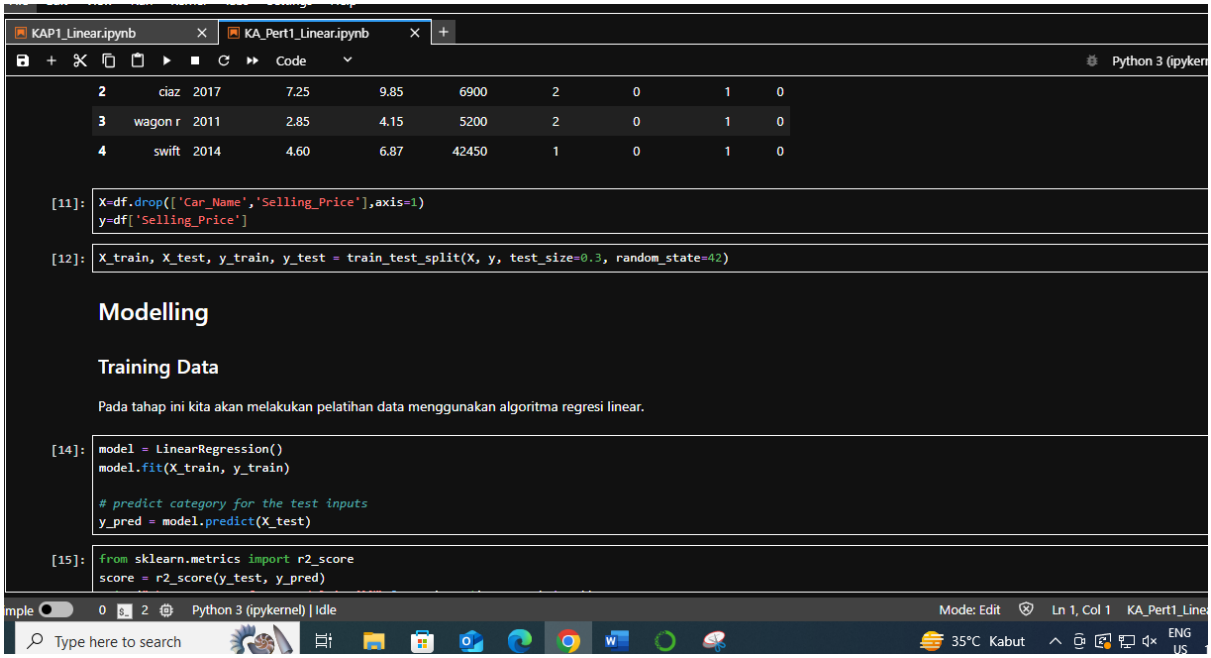
Data Pre-processing

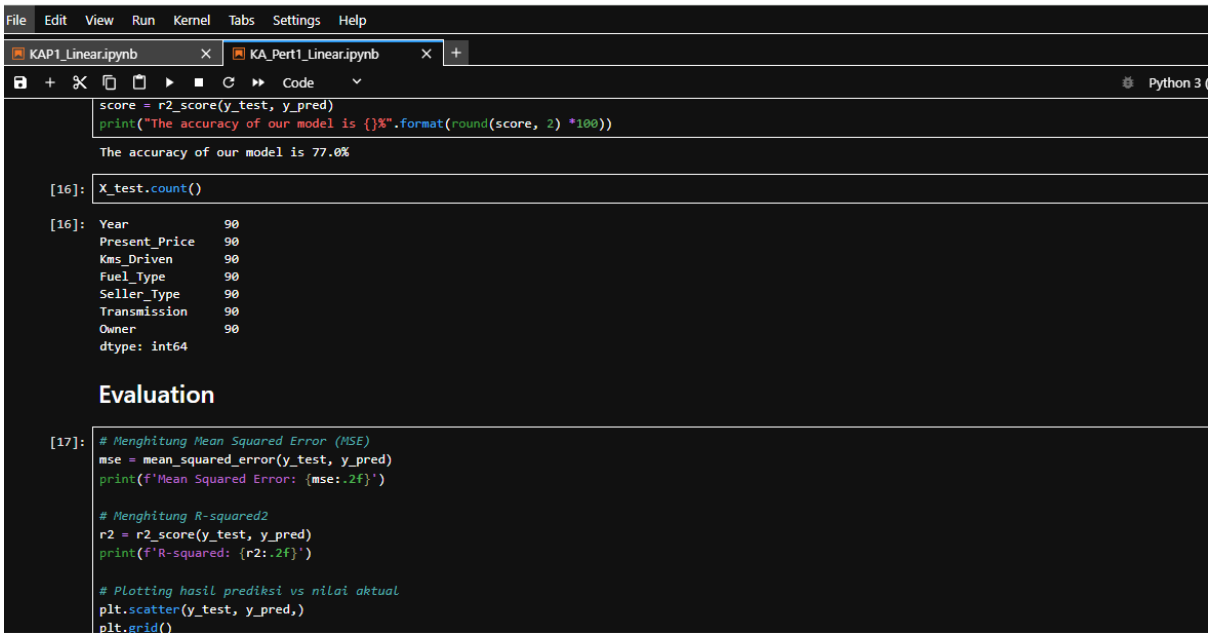
```
[10]: # Create a sample dataframe with categorical data
Fuel = pd.DataFrame({'Fuel_Type': ['Petrol', 'Diesel', 'CNG']})
Seller = pd.DataFrame({'Seller_Type': ['Dealer', 'Individual']})
Transmiss = pd.DataFrame({'Transmission': ['Manual', 'Automatic']})

# Create a LabelEncoder object
le = LabelEncoder()

# Fit and transform the categorical data
df['Fuel_Type'] = le.fit_transform(df['Fuel_Type'])
df['Seller_Type'] = le.fit_transform(df['Seller_Type'])
df['Transmission'] = le.fit_transform(df['Transmission'])
df.head()
```

[10]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	2	0	1	0
1	sx4	2013	4.75	9.54	43000	1	0	1	0
2	ciaz	2017	7.25	9.85	6900	2	0	1	0

Simple | 0 | 2 | Python 3 (ipykernel) | Idle | Mode:


KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3 (ipykern
2 ciaz 2017 7.25 9.85 6900 2 0 1 0
3 wagon r 2011 2.85 4.15 5200 2 0 1 0
4 swift 2014 4.60 6.87 42450 1 0 1 0
[11]: X=df.drop(['Car_Name','Selling_Price'],axis=1)
y=df['Selling_Price']
[12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
Modelling
Training Data
Pada tahap ini kita akan melakukan pelatihan data menggunakan algoritma regresi linear.
[14]: model = LinearRegression()
model.fit(X_train, y_train)
# predict category for the test inputs
y_pred = model.predict(X_test)
[15]: from sklearn.metrics import r2_score
score = r2_score(y_test, y_pred)
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 1, Col 1
KA_Pert1_Linea
Type here to search
35°C Kabut
ENG
US


File Edit View Run Kernel Tabs Settings Help
KAP1_Linear.ipynb
KA_Pert1_Linear.ipynb
Code
Python 3
score = r2_score(y_test, y_pred)
print("The accuracy of our model is {}%".format(round(score, 2) *100))
The accuracy of our model is 77.0%
[16]: X_test.count()
[16]: Year 90
Present_Price 90
Kms_Driven 90
Fuel_Type 90
Seller_Type 90
Transmission 90
Owner 90
dtype: int64
Evaluation
[17]: # Menghitung Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')
# Menghitung R-squared2
r2 = r2_score(y_test, y_pred)
print(f'R-squared: {r2:.2f}')
# Plotting hasil prediksi vs nilai aktual
plt.scatter(y_test, y_pred,)
plt.grid()

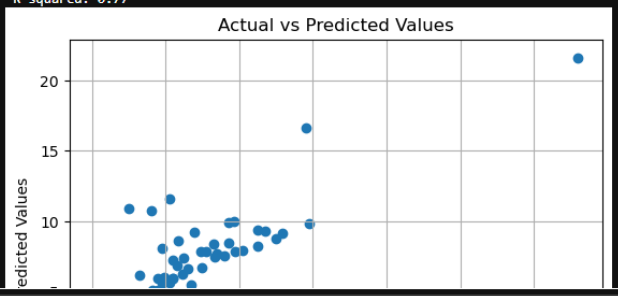
File Edit View Run Kernel Tabs Settings Help

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code

```
# Menghitung R-squared2
r2 = r2_score(y_test, y_pred)
print(f'R-squared: {r2:.2f}')

# Plotting hasil prediksi vs nilai aktual
plt.scatter(y_test, y_pred,)
plt.grid()
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values')
plt.show()
```

```
Mean Squared Error: 5.41
R-squared: 0.77
```

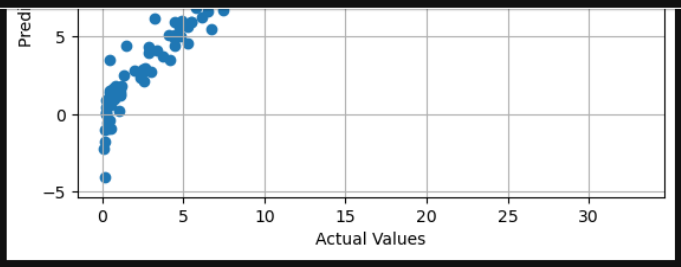


Simple 0 2 Python 3 (ipykernel) | Idle Mode: Edit Ln 1, Col

File Edit View Run Kernel Tabs Settings Help

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code



[18]:

```
# Menghitung Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')

# Menghitung R-squared
r2 = r2_score(y_test, y_pred)
print(f'R-squared: {r2:.2f}')

# Plotting hasil prediksi vs nilai aktual dengan scatter plot
plt.scatter(y_test, y_pred)
plt.grid()
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values')

# Menambahkan regplot dari Seaborn untuk garis regresi
```

Simple 0 2 Python 3 (ipykernel) | Idle Mode: Edit

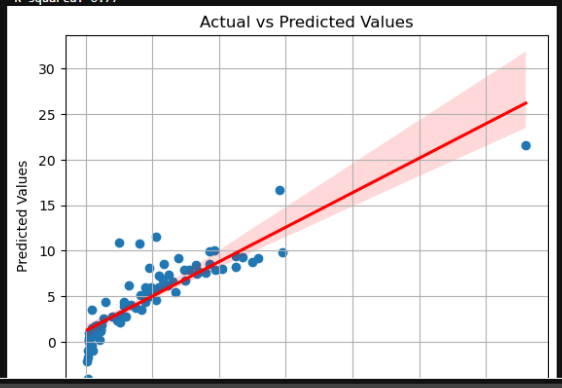
File Edit View Run Kernel Tabs Settings Help

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code | Python 3 (ipykerne

```
# Menambahkan regplot dari Seaborn untuk garis regresi
sns.regplot(x=y_test, y=y_pred, scatter=False, color='red')

plt.show()
```

```
Mean Squared Error: 5.41
R-squared: 0.77
```



Simple | 0 | 2 | Python 3 (ipykernel) | Idle | Mode: Edit | Ln 1, Col 1 | KA_Pert1_Linear

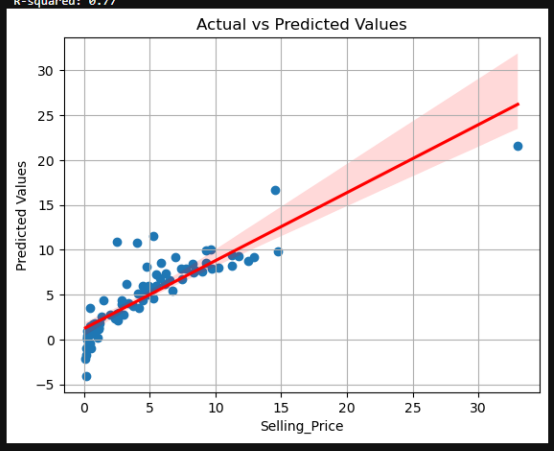
Type here to search | 35°C Kabut | ENG US

File Edit View Run Kernel Tabs Settings Help

KAP1_Linear.ipynb | KA_Pert1_Linear.ipynb

Code | Python 3 (ipykernel)

```
Mean Squared Error: 5.41
R-squared: 0.77
```



Snip & Sketch

Snip saved to clipboard

Select here to mark up and share the ima

Simple | 0 | 2 | Python 3 (ipykernel) | Idle | Mode: Edit

Type here to search | 35°C Kabut | ENG