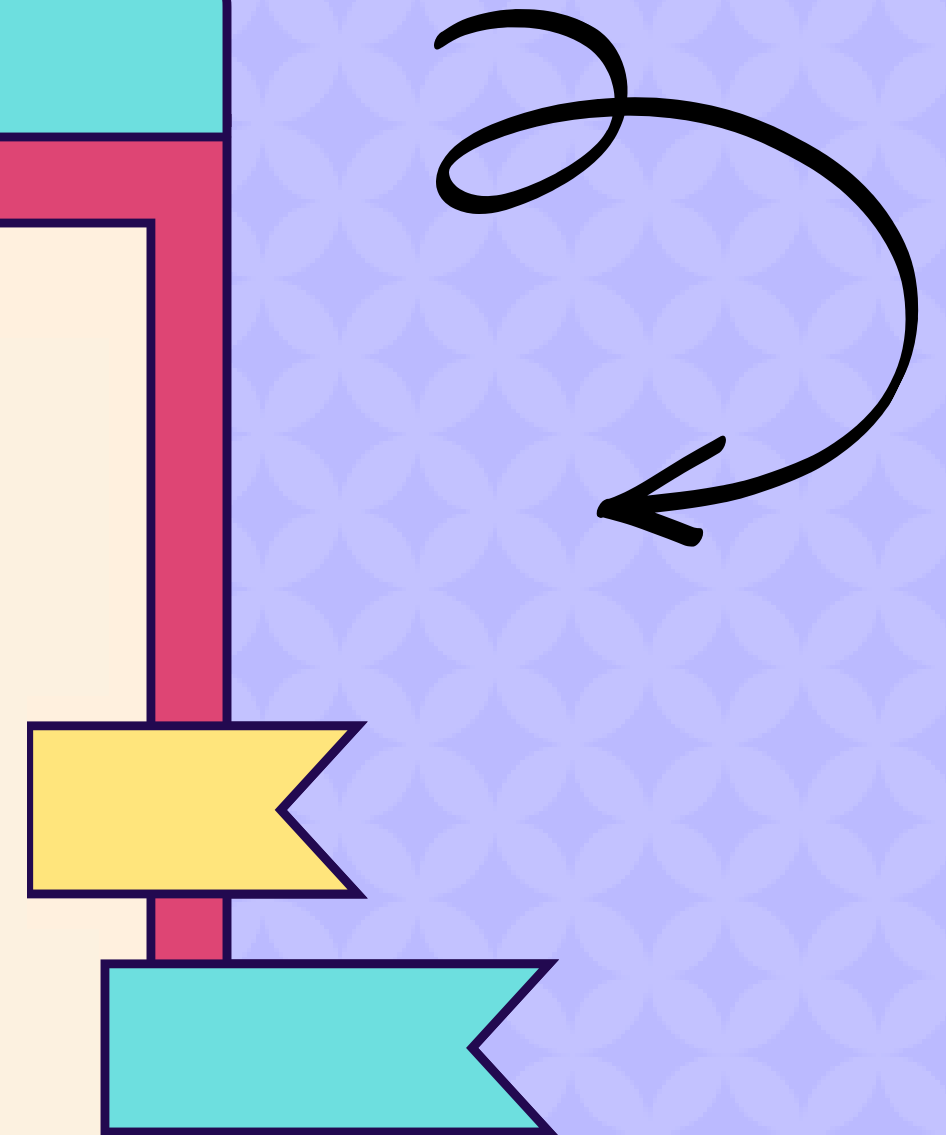
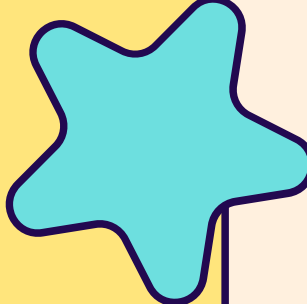




MACHINE LEARNING

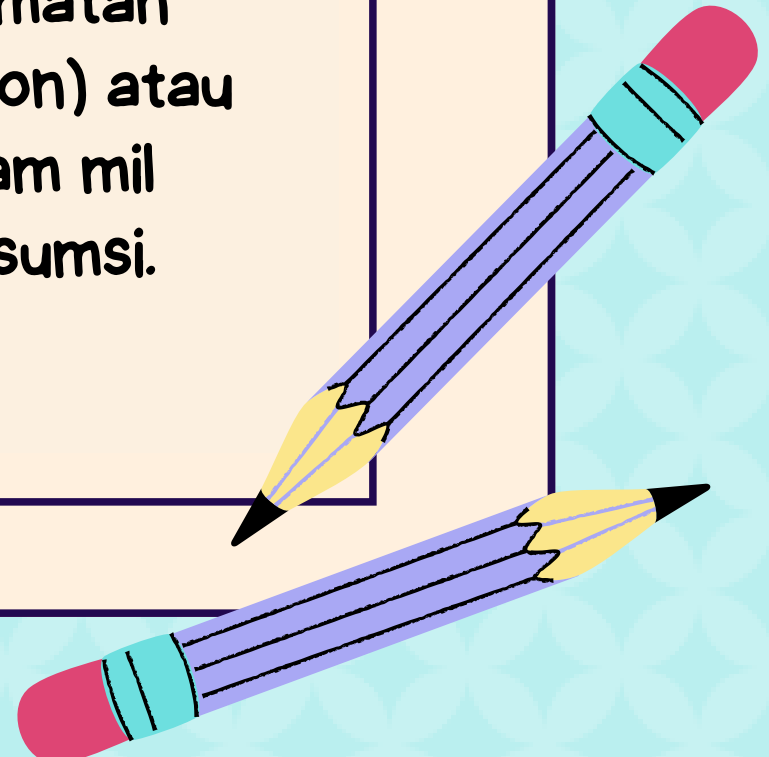
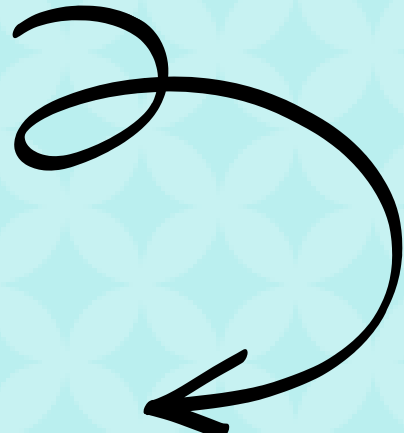
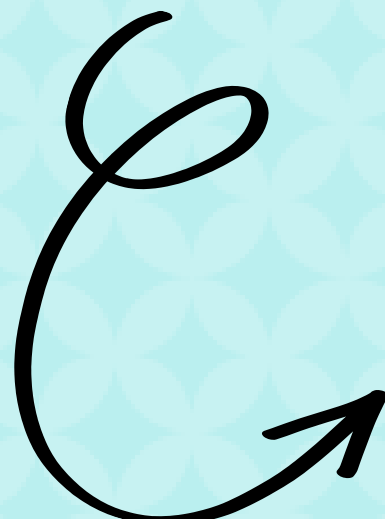
-NOVALDI RAMADHAN WALUYO (1301204423)
-THESSALONIKA ALEXANDRA MANURUNG (1301204372)
-MUHAMMAD RIZKI NURFIQRI (1301204009)
-SIGILIA ARABELLA MATULESSY (1301200128)





FORMULASI MASALAH

Saat ini banyak orang yang sudah memiliki kendaraan berbahan bakar bensin maupun solar. Tetapi bagaimana cara memprediksi tingkat kehematan bahan bakar tersebut?, maka dari itu laporan ini bertujuan untuk menyelesaikan permasalahan tersebut dengan memprediksi tingkat kehematan bahan bakar kendaraan/MPG (miles per gallon) atau bisa disebut rata-rata jarak tempuh mobil dalam mil untuk setiap galon bahan bakar yang dikonsumsi.



EKSPLORASI DAN PRA-PEMROSESAN DATA



LOAD DATA KE GOOGLE COLAB :

```
[15] from pydrive.auth import GoogleAuth
      from pydrive.drive import GoogleDrive
      from google.colab import auth
      from oauth2client.client import GoogleCredentials
```

```
[16] auth.authenticate_user()
      gauth = GoogleAuth()
      gauth.credentials = GoogleCredentials.get_application_default()
      drive = GoogleDrive(gauth)
```

```
▶ import matplotlib.pyplot as plt
   from sklearn.model_selection import train_test_split
   import torch
```

```
[18] downloaded = drive.CreateFile({'id': "1xgAI4INpK63bKbfK270ceEeXWEEJYsJf"}) # replace the id with id of file you want to access
      downloaded.GetContentFile('autos_mpg.csv') # replace the file name with your file
```

```
[19] import pandas as pd
```

```
[20] df = pd.read_csv("autos_mpg.csv")
```

EKSPLORASI DATA

01

```
df.head()
```

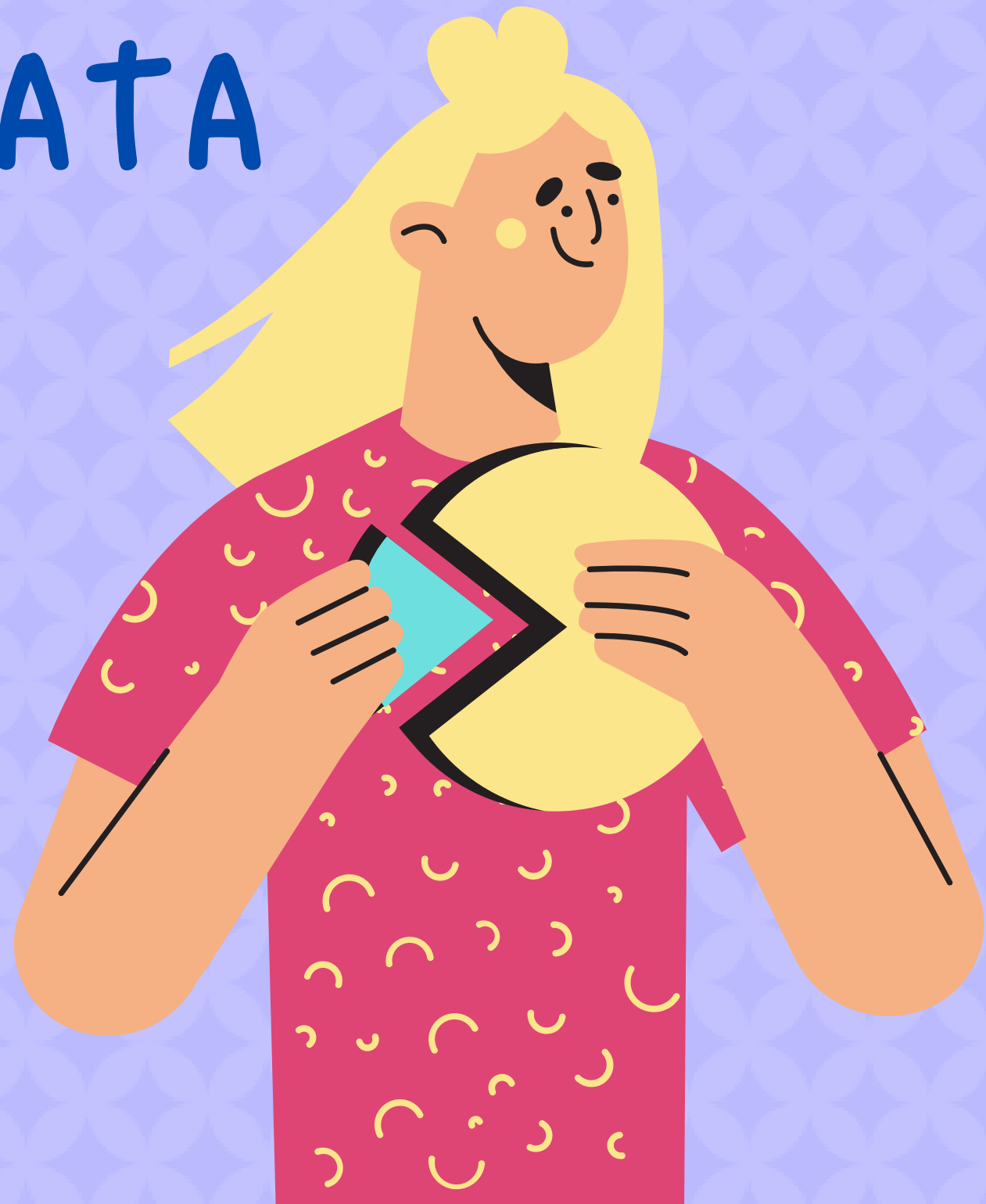
	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

02

```
df.describe()
```

	mpg	cylinders	displacement	weight	acceleration	model_year	origin
count	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	2970.424623	15.568090	76.010050	1.572864
std	7.815984	1.701004	104.269838	846.841774	2.757689	3.697627	0.802055
min	9.000000	3.000000	68.000000	1613.000000	8.000000	70.000000	1.000000
25%	17.500000	4.000000	104.250000	2223.750000	13.825000	73.000000	1.000000
50%	23.000000	4.000000	148.500000	2803.500000	15.500000	76.000000	1.000000
75%	29.000000	8.000000	262.000000	3608.000000	17.175000	79.000000	2.000000
max	46.600000	8.000000	455.000000	5140.000000	24.800000	82.000000	3.000000

PRA- PEMROSESAN DATA



MISSING VALUE

Untuk mengetahui apakah di dalam data ada nilai yang hilang maka kami akan mengecek terlebih dahulu, seperti berikut :

```
#Cek nilai missing value  
df.isnull().sum()
```

mpg	0
cylinders	0
displacement	0
horsepower	0
weight	0
acceleration	0
model_year	0
origin	0
car_name	0
dtype:	int64

SCALING

SKemudian mengikuti proses penskalaan, atau biasa dikenal dengan normalisasi. Data yang dimiliki mungkin jauh jangkauannya, maka perlu menormalkannya atau normalisasi, sebagai berikut :

```
import numpy as np  
  
df = df.replace("?", np.nan)
```

```
df = df.fillna(30)  
df
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows x 9 columns



PRA-PEMROSESAN DATA

SPLITTING DATA

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
x = df[["cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "origin"]]
y = df[["mpg"]]
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y)
```

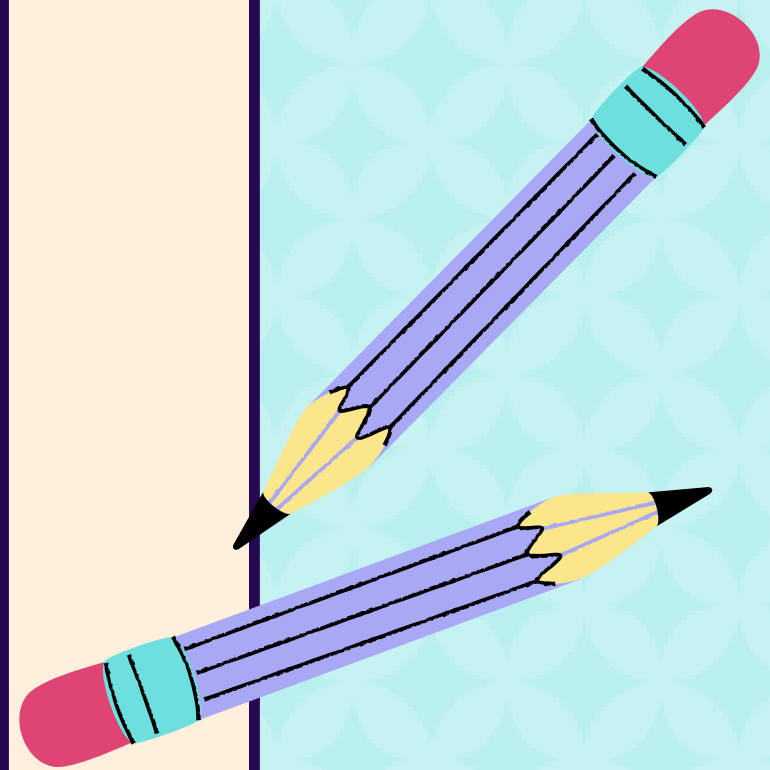
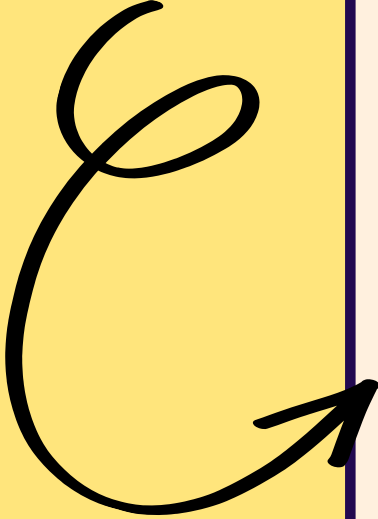
```
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.fit_transform(x_test)
```




PEMODELAN

Dalam Project Based Machine Learning ini, kelompok kami menggunakan satu library yang bernama BaggingRegressor. Dimana BaggingRegressor sendiri melatih setiap model regressor yang terdapat pada subset yang acak dari dataset yang asli dan juga menggabungkan hasil prediksi untuk membentuk prediksi akhir.

```
from sklearn.ensemble import BaggingRegressor  
  
model = BaggingRegressor()  
  
model = model.fit(x_train_scaled, y_train)
```



EVALUASI



Pada bagian evaluasi, kelompok kami menggunakan MAE, MSE, dan juga RMSE untuk memprediksi nilai skor akhir dengan menggunakan library sebelumnya. Sebelum mendapatkan nilai akhir, harus mengimport `mean_absolute_error` dan `mean_squared_error` terlebih dahulu. Kemudian dilanjut dengan menggunakan parameter `y_test` dan `y_pred` untuk mendapatkan nilai.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

```
print("MAE : ", mae)
print("MSE : ", mse)
print("RMSE: ", rmse)
```

```
MAE : 3.3049999999999993
MSE : 17.859777999999999
RMSE: 4.22608305644837
```

TEKNIK ANALISIS DATA

ARTI MAE

Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. Hasil Mean Absolute Error menunjukkan nilai rata-rata kesalahan mutlak atau absolut dari nilai sebenarnya dengan nilai peramalan.

ARTI MSE

Mean Squared Error (MSE) adalah Rata-rata Kesalahan kuadrat di antara nilai aktual dan nilai peramalan. Metode Mean Squared Error secara umum digunakan untuk mengecek estimasi berapa nilai kesalahan pada peramalan. Nilai Mean Squared Error yang rendah atau nilai mean squared error mendekati nol menunjukkan bahwa hasil peramalan sesuai dengan data aktual dan bisa dijadikan untuk perhitungan peramalan di periode mendatang. Metode Mean Squared Error biasanya digunakan untuk mengevaluasi metode pengukuran dengan model regresi atau model peramalan seperti Moving Average, Weighted Moving Average dan Analisis Trendline

ARTI RMSE

Root Mean Square Error (RMSE) merupakan besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) nilai RMSE maka hasil prediksi akan semakin akurat. Root Mean Squared Error (RMSE) merupakan salah satu cara untuk mengevaluasi model regresi linear dengan mengukur tingkat akurasi hasil perkiraan suatu model.

Berikut hasil dari MAE, MSE, DAN RMSE

MAE : 3.304999999999999999999993

MSE : 17.859777999999999999999999

RMSE: 4.22608305644837




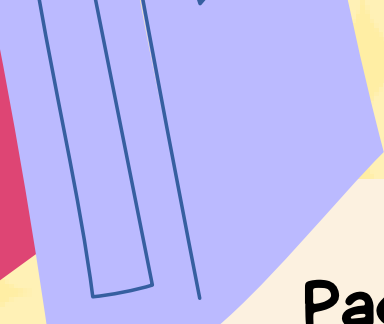
KESIMPULAN

etapi bagaimana cara memprediksi tingkat kehematan bahan bakar tersebut?, Maka dari itu, laporan ini bertujuan untuk menyelesaikan permasalahan tersebut dengan memprediksi tingkat kehematan bahan bakar kendaraan/MPG (miles per gallon) atau bisa disebut rata-rata jarak tempuh mobil dalam mil untuk setiap galon bahan bakar yang dikonsumsi.

Eksplorasi dan Pra-pemrosesan Data Sebelum melakukan eksplorasi, sudah dipastikan harus meload data ke google collab. Disini kami meload data dari google drive ke google collab, seperti berikut : Gambar 2.1 hasil dari eksplorasi data

Data pra-pemrosesan data sendiri memiliki maksud yakni sekumpulan dari berbagai metode yang diterapkan pada data untuk menghapus noise, missing value, dan data yang tidak konsisten. Terkadang data yang baru saja dikumpulkan pasti mengandung beberapa kesalahan seperti missing value, error, dan data-data yang tidak penting.

Untuk mengetahui apakah di dalam data ada nilai yang hilang maka kami akan mengecek terlebih dahulu. Disini kami mendapatkan 0 missing value, yang berarti di dalam data ini tidak ada nilai yang kosong atau hilang. Dimana BaggingRegressor sendiri melatih setiap model regressor yang terdapat pada subset yang acak dari dataset yang asli dan juga menggabungkan hasil prediksi untuk membentuk prediksi akhir.





Pada bagian evaluasi, kelompok kami menggunakan MAE, MSE, dan juga RMSE untuk memprediksi nilai skor akhir dengan menggunakan library sebelumnya. Sebelum mendapatkan nilai akhir, harus mengimport `mean_absolute_error` dan `mean_squared_error` terlebih dahulu. Kemudian dilanjutkan dengan menggunakan parameter `y_test` dan `y_pred` untuk mendapatkan nilai.

Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. Hasil Mean Absolute Error menunjukkan nilai rata-rata kesalahan mutlak atau absolut dari nilai sebenarnya dengan nilai peramalan. Mean Squared Error (MSE) adalah Rata-rata Kesalahan kuadrat antara nilai aktual dan nilai peramalan. Metode Mean Squared Error secara umum digunakan untuk mengecek estimasi berapa nilai kesalahan pada peramalan.

Nilai Mean Squared Error yang rendah atau nilai mean squared error mendekati nol menunjukkan bahwa hasil peramalan sesuai dengan data aktual dan bisa dijadikan untuk perhitungan peramalan di periode mendatang. Metode Mean Squared Error biasanya digunakan untuk mengevaluasi metode pengukuran dengan model regresi atau model peramalan seperti Moving Average, Weighted Moving Average dan Analisis Trendline.

Root Mean Square Error (RMSE) merupakan besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) nilai RMSE maka hasil prediksi akan semakin akurat. Root Mean Squared Error (RMSE) merupakan salah satu cara untuk mengevaluasi model regresi linear dengan mengukur tingkat akurasi hasil perkiraan suatu model.

Bisa disimpulkan bahwa RMSE dan MSE kurang lebih memiliki properti yang sama, karena sebenarnya RMSE merupakan akar kuadrat dari MSE. Dan juga nilai RMSE akan cenderung lebih naik diikuti ukuran sampel yang juga naik.



thank you!

