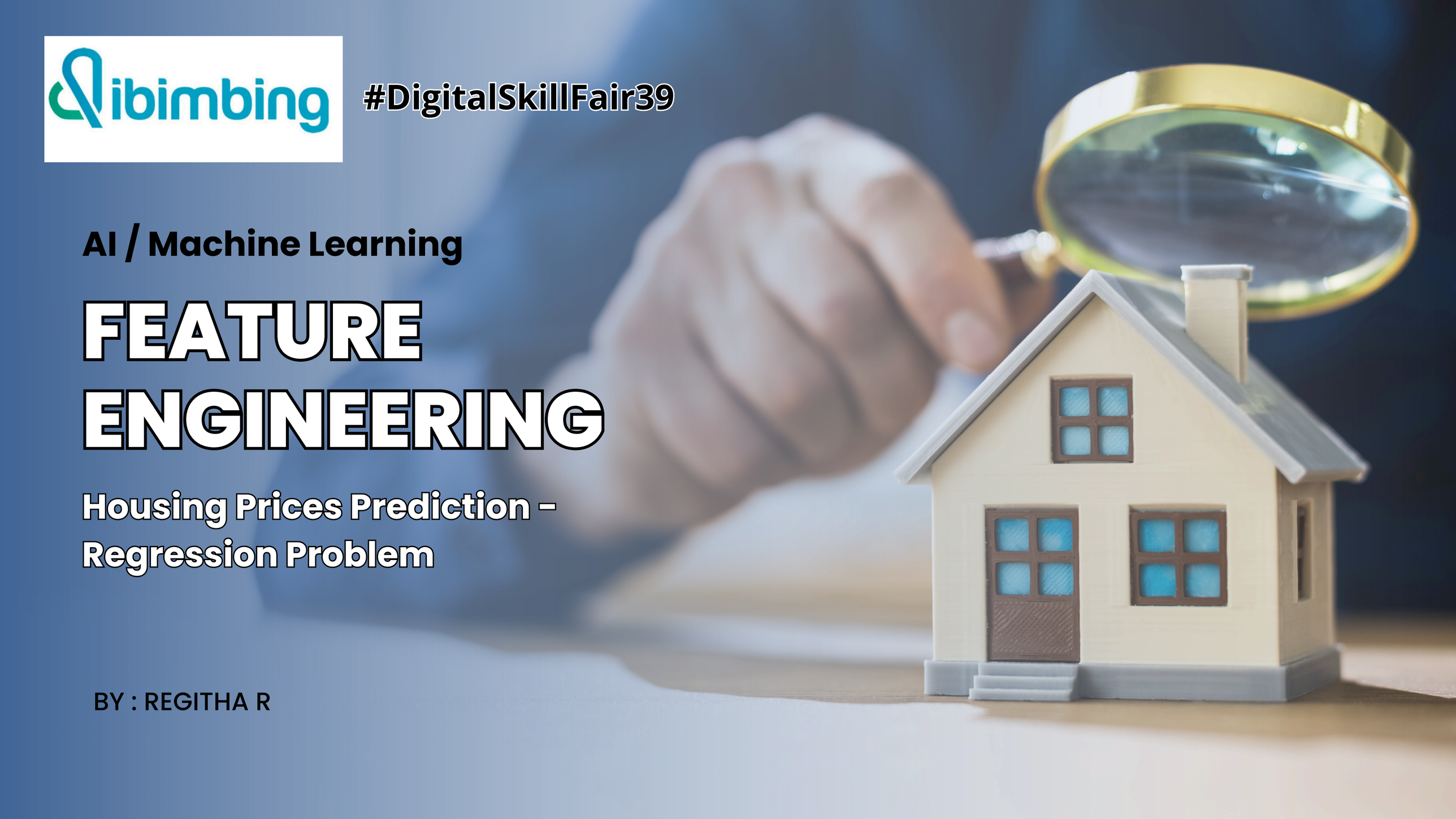
ibimbing

#DigitalSkillFair39

AI / Machine Learning

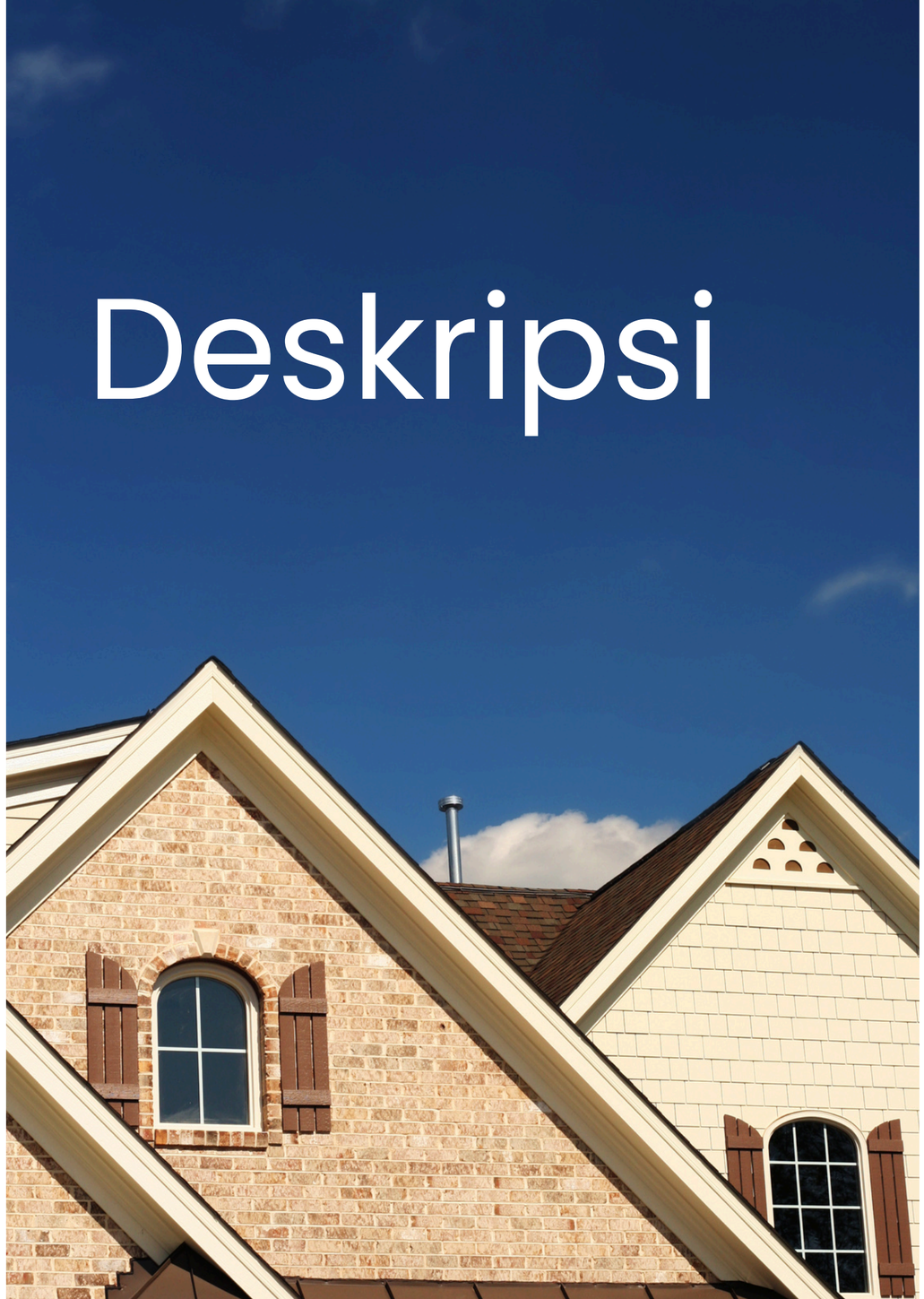
FEATURE ENGINEERING

Housing Prices Prediction - Regression Problem



BY : REGITHA R

Deskripsi



Tujuan

Projek kali ini di rancang untuk memprediksi harga rumah berdasarkan area dan fasilitas rumah yang tersedia.

Dataset

housing-prices-dataset Kaggle : https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

Metode

Metode algoritma yang digunakan adalah akgoritma Random Forest Regressor

IMPORT LIBRARY

```
import numpy as np # untuk perhitungan statistik
from numpy import mean # mencari nilai mean
import pandas as pd #membaca, pembersihan, analisis data
import matplotlib.pyplot as plt # visualisasi data
%matplotlib inline
import seaborn as sns # membuat grafik dan statistik
from sklearn.preprocessing import LabelEncoder, StandardScaler #pemrosesan data
from sklearn import preprocessing # pemrosesan data
from sklearn.model_selection import train_test_split # membagi data menjadi dua, yaitu data testing dan training
from sklearn import datasets
from sklearn.metrics import accuracy_score # mencari accurasi
from sklearn.metrics import confusion_matrix # membuat confusion maytris

import warnings
warnings.filterwarnings("ignore")

print('Import Library Berhasil')
```

```
Import Library Berhasil
```

LOADING DATA SET

[3]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no	unfurnished
541	1767150	2400	3	1	1	no	no	no	no	no	0	no	semi-furnished
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no	unfurnished
543	1750000	2910	3	1	1	no	no	no	no	no	0	no	furnished
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no	unfurnished

545 rows × 13 columns

Dari data yang telah terbaca tertera bahwa dataset tersebut ber isi 545 baris dan 13 kolom.

Type Data

```
[1]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

Deskriptif Statistik

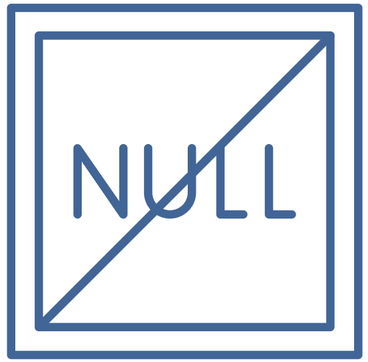
```
[9]: df.describe()
```

[9]:

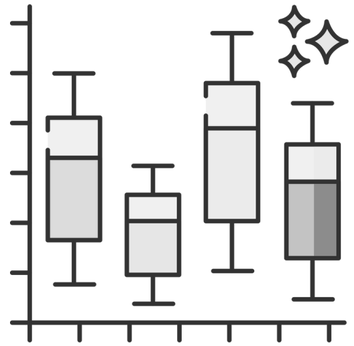
	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

FEATURE ENGINEERING

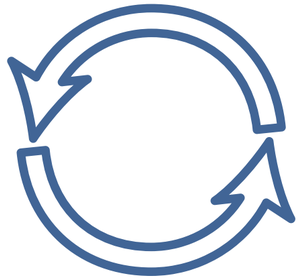
✔ Remove duplicates



✔ Handle missing values



✔ Handle outliers



✔ continuous values

✔ Handle categorical values

Remove Duplicates

Cek Duplikat

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

[13]: 0
```

Handle missing values

Pengecekan Missing Value

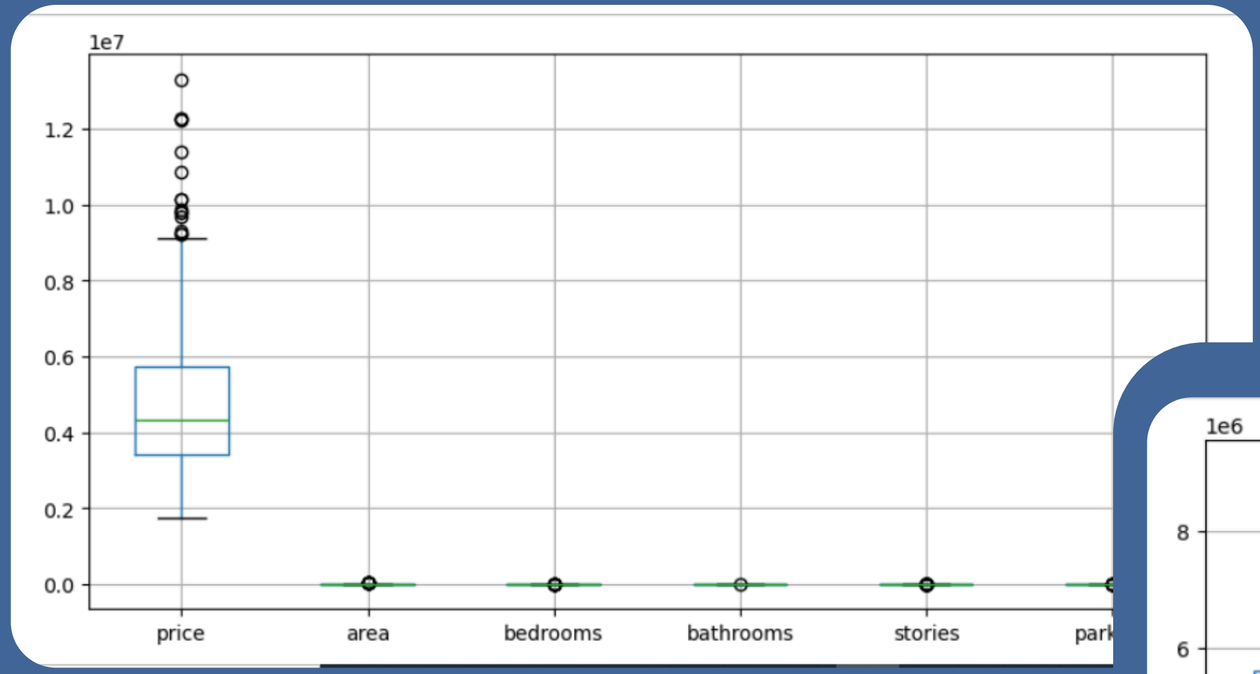
```
15]: df.isnull().sum()

15]: price             0
     area              0
     bedrooms          0
     bathrooms         0
     stories           0
     mainroad          0
     guestroom         0
     basement          0
     hotwaterheating   0
     airconditioning   0
     parking           0
     prefarea          0
     furnishingstatus  0
     dtype: int64
```

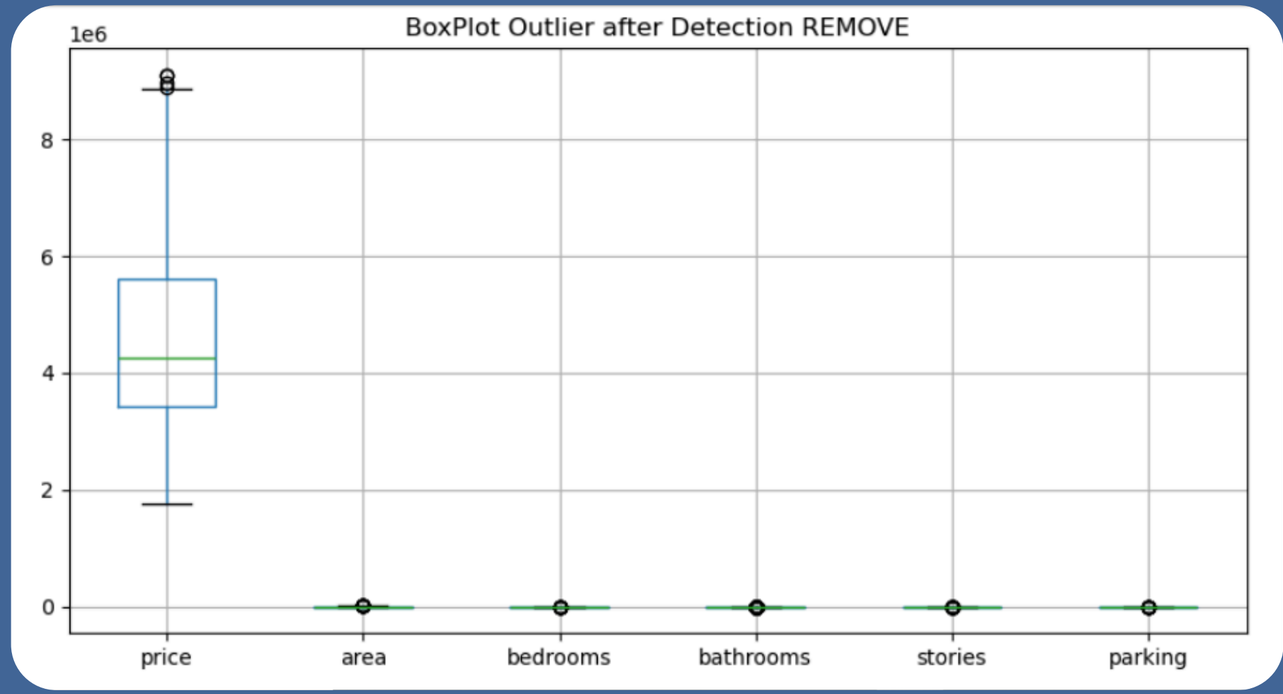
Pada data set tersebut tidak terdapat duplikat sata dan tidak terdapat missing value (Null)

Handle outliers

Sebelum Remove



Sesudah Remove



Pada data set terdapat outlier pada nilai kolom price.

Process categorical features (Label Encoder)

Sebelum Encoder

[3]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no	unfurnished
541	1767150	2400	3	1	1	no	no	no	no	no	0	no	semi-furnished
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no	unfurnished
543	1750000	2910	3	1	1	no	no	no	no	no	0	no	furnished
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no	unfurnished

545 rows × 13 columns

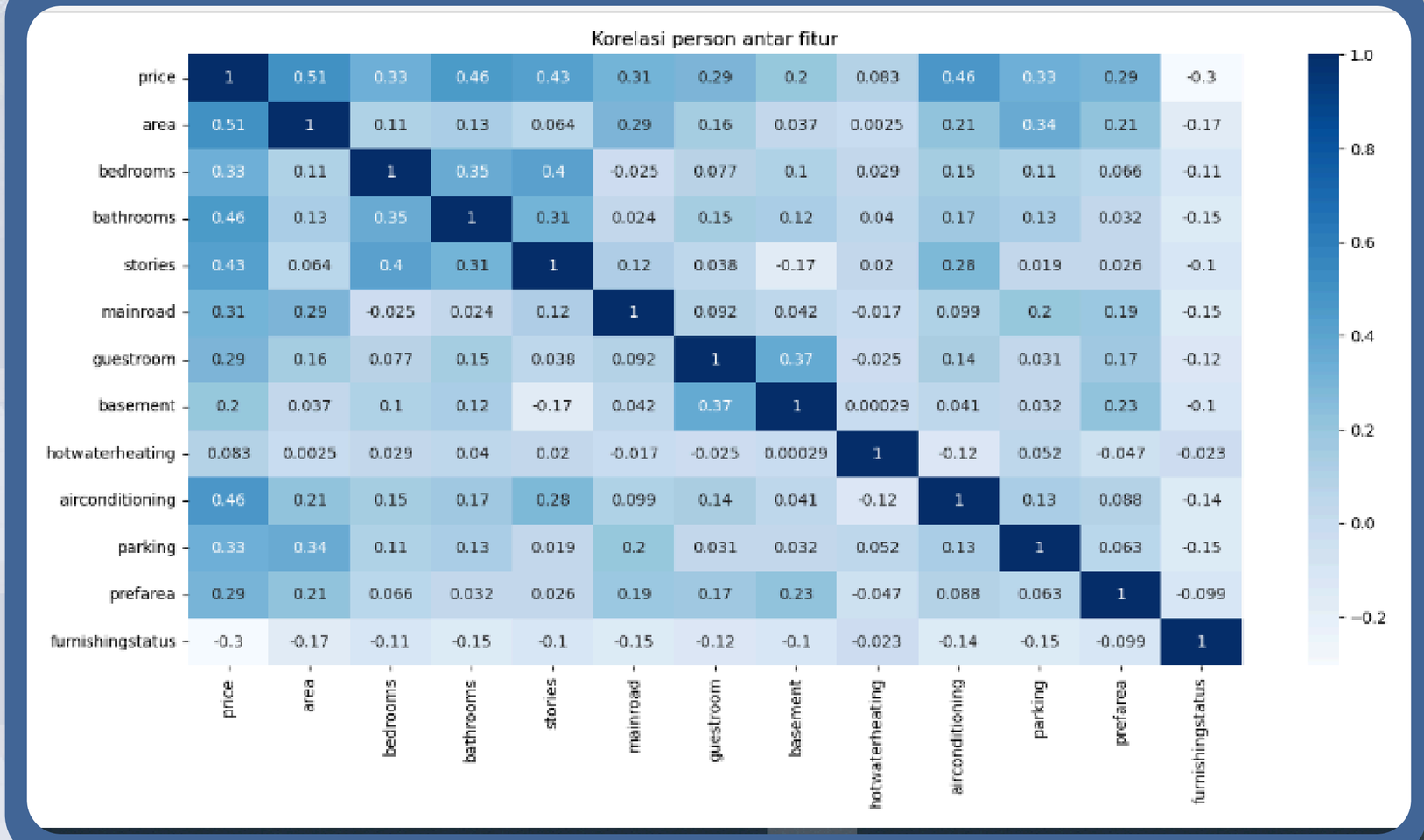
Sesudah Encoder

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
15	9100000	6000	4	1	2	1	0	1	0	0	2	0	1
16	9100000	6600	4	2	2	1	1	1	0	1	1	1	2
17	8960000	8500	3	2	4	1	0	0	0	1	2	0	0
18	8890000	4600	3	2	2	1	1	0	0	1	2	0	0
19	8855000	6420	3	2	2	1	0	0	0	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	1	0	1	0	0	2	0	2
541	1767150	2400	3	1	1	0	0	0	0	0	0	0	1
542	1750000	3620	2	1	1	1	0	0	0	0	0	0	2
543	1750000	2910	3	1	1	0	0	0	0	0	0	0	0
544	1750000	3850	3	1	2	1	0	0	0	0	0	0	2

530 rows × 13 columns

Pada data set terdapat 7 fitur yang dilakukan Label encoder yaitu fitur mainbord, guestroom, basement, hotwaterheating, airconditioning, prefea, dan furnising status.

Seleksi Fitur



Pada correlation map di atas setelah dilakukan encoding, terdapat hasil korelasi yang bagus dan agak bagus: Hasil korelasi yang memiliki korelasi positif adalah hubungan antara Price dengan area, dan airconditioning. Lalu hubungan bathrooms dan stories, hubungan antara area dengan parking, hubungan antara basement dengan guestroom.

Sehingga fitur yang akan digunakan adalah "price","area","airconditioning","bathrooms","stories", "parking","basement", dan"guestroom".

Continuous values (Standarisasi)

Sebelum Standarisasi

[3]:

	price	area	be
0	13300000	7420	
1	12250000	8960	
2	12250000	9960	
3	12215000	7500	
4	11410000	7420	
...	...	...	
540	1820000	3000	
541	1767150	2400	
542	1750000	3620	
543	1750000	2910	
544	1750000	3850	

545 rows × 13 columns

Sesudah Standarisasi

	price	area
15	2.821586	0.452609
16	2.821586	0.741976
17	2.733790	1.658306
18	2.689893	-0.222581
19	2.667944	0.655166
...	...	...
540	-1.743786	-0.994226
541	-1.776929	-1.283593
542	-1.787684	-0.695214
543	-1.787684	-1.037631
544	-1.787684	-0.584290

530 rows × 8 columns

Dilakukan standarisasi pada nilai fitur Price dan area untuk menyamakan skla fitur.

PENGUJIAN MODEL

Random Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
```

```
rasio = [0.2,0.3,0.4]
```

```
#model
model_rfc = RandomForestRegressor(n_estimators=100, random_state=42)
```

Pengujian menggunakan model Random Forest dengan percobaan 3 ratio yaitu 0.2, 0.3 dan 0.4 menghasilkan akurasi pelatihan tertinggi pada ratio 0.3 dengan hasil akurasi 93 %

```
for i in rasio:

    #setting rasio loop
    x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=i, random_state=42)

    #memuat model
    model = model_rfc

    #melatih model
    #model.fit(x_train, y_train)

    model.fit(x_train, y_train)


    #menghitung sisi dari pelatihan
    y_pred_train =  model.predict(x_train)

    #menghitung sisi dari pengujian
    y_pred_test =  model.predict(x_test)

    print(f"R2_score pelatihan Random Forest rasio {i} adalah : {r2_score(y_pred_train,y_train)}")
```

```
R2_score pelatihan Random Forest rasio 0.2 adalah : 0.91661875712394
R2_score pelatihan Random Forest rasio 0.3 adalah : 0.9227598778209787
R2_score pelatihan Random Forest rasio 0.4 adalah : 0.9176181495825938
```

PENGUJIAN MODEL

Linear Regresi

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
model_lr = LinearRegression()
```

Pengujian menggunakan model Linear Regresi dengan percobaan 3 ratio yaitu 0.2, 0.3 dan 0.4 tidak mengghasilkan akurasi pelatihan yang bagus semua nilai akurasi berada dibawah nilai 50 %.

```
]: for i in rasio:

    #setting rasio loop
    x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=i, random_state=42)

    #memuat model
    model = model_lr

    #melatih model
    #model.fit(x_train, y_train)

    model.fit(x_train, y_train)


    #menghitung sisi dari pelatihan
    y_pred_train =  model.predict(x_train)

    #menghitung sisi dari pengujian
    y_pred_test =  model.predict(x_test)

    print(f"R2_score pelatihan Linear Regresi {i} adalah : {r2_score(y_pred_train,y_train)}")
```

```
R2_score pelatihan Linear Regresi 0.2 adalah : 0.3456847550237455
R2_score pelatihan Linear Regresi 0.3 adalah : 0.3749782681240157
R2_score pelatihan Linear Regresi 0.4 adalah : 0.4046295165168923
```

PENGUJIAN MODEL

Gradient Boosting

```
[ ]: from sklearn.ensemble import GradientBoostingRegressor
```

```
[ ]: model_gbr = GradientBoostingRegressor(random_state=42)
```

Pengujian menggunakan model Gradient Boosting dengan percobaan 3 ratio yaitu 0.2, 0.3 dan 0.4 menghasilkan akurasi pelatihan yang tidak terlalu tinggi tertinggi. Nilai tertinggi pada ratio 0.4 dengan hasil akurasi 78 %

```
for i in rasio:

    #setting rasio loop
    x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=i, random_state=42)

    #memuat model
    model = model_gbr

    #melatih model
    #model.fit(x_train, y_train)

    model.fit(x_train, y_train)


    #menghitung sisi dari pelatihan
    y_pred_train =  model.predict(x_train)

    #menghitung sisi dari pengujian
    y_pred_test =  model.predict(x_test)

    print(f"R2_score pelatihan Gradient Boosting rasio {i} adalah : {r2_score(y_pred_train,y_train)}")
```

```
R2_score pelatihan Gradient Boosting rasio 0.2 adalah : 0.7010607433695308
R2_score pelatihan Gradient Boosting rasio 0.3 adalah : 0.7481834822758464
R2_score pelatihan Gradient Boosting rasio 0.4 adalah : 0.7899169713364891
```

KESIMPULAN

Model algoritma yang di uji :

1. **Random Forest**
2. **Linear Regresi**
3. **Gradient Booster**

01

Data yang digunakan sudah baik dan rapih, ini dibuktikan dengan tidak adanya duplikat, missing value dan minim outlier pada value data

02

Setelah melakukan seleksi fitur menggunkan korelasi, pada projek kali ini fitur yang akan digunakan adalah "price","area","airconditioning", "bathrooms","stories", "parking","basement", dan"guestroom".

03

Dari 3 model algoritma yang telah di uji, pada data set projek ini bagus untuk menggunkan algoritma Random Forest pada rasio 0.3 dengn nilai akurasi 93%.

Thank You!