```
import pandas as pd
import numpy as np
df_kategori=pd.read_csv('Dataset UTS Gasal 2425.csv')
df kategori.head(20)
                    numberofrooms hasyard haspool
                                                      floors
                                                                citycode \
    squaremeters
0
            75523
                                                           63
                                                                    9373
                                 3
                                         no
                                                 yes
                                58
1
            55712
                                                           19
                                                                   34457
                                         no
                                                 yes
2
            86929
                               100
                                                                   98155
                                        yes
                                                           11
                                                  no
3
            51522
                                 3
                                                           61
                                                                    9047
                                         no
                                                  no
4
            96470
                                74
                                                                   92029
                                                           21
                                        yes
                                                  no
5
            79770
                                 3
                                                           69
                                                                   54812
                                         no
                                                 yes
6
                                60
            75985
                                        yes
                                                  no
                                                           67
                                                                    6517
7
                                                                   61711
            64169
                                88
                                         no
                                                 yes
                                                            6
8
            92383
                                12
                                                                   71982
                                                           78
                                         no
                                                  no
9
            95121
                                46
                                                            3
                                                                    9382
                                         no
                                                 yes
10
            76485
                                47
                                                            9
                                                                   90254
                                        yes
                                                  no
11
            87060
                                27
                                                           91
                                                                   51803
                                         no
                                                 yes
12
            66683
                                19
                                                            6
                                                                   50801
                                        yes
                                                 yes
13
            84559
                                29
                                                           69
                                                                   53057
                                         no
                                                 yes
14
                                38
                                                           32
                                                                   59451
            76091
                                        yes
                                                  no
15
                                49
                                                           38
            92696
                                        yes
                                                  no
                                                                   74381
16
                                47
                                                           27
                                                                   44815
            59800
                                         no
                                                 yes
17
            54836
                                25
                                                           53
                                                                   64601
                                         no
                                                 yes
18
            70021
                                52
                                                           28
                                                                   95678
                                        yes
                                                  no
19
            54368
                                11
                                        yes
                                                 yes
                                                           20
                                                                   55761
    citypartrange
                                      made isnewbuilt hasstormprotector
                     numprevowners
basement \
                  3
                                      2005
                                  8
                                                   old
                                                                        yes
4313
                  6
                                  8
                                      2021
                                                   old
1
                                                                         no
2937
                  3
                                      2003
2
                                   4
                                                   new
                                                                         no
6326
                  8
                                   3
                                      2012
3
                                                   new
                                                                        yes
632
                  4
                                   2
                                      2011
                                                   new
                                                                        yes
5414
                 10
                                   5
                                      2018
                                                   old
                                                                        yes
8871
                  6
                                      2009
6
                                                   new
                                                                        yes
4878
                  3
                                   9
                                      2011
                                                   new
                                                                        yes
3054
                  3
                                   7
8
                                      2000
                                                   old
                                                                         no
7507
                  7
                                      1994
                                                   old
9
                                                                         no
```

10		2	9	2008	ne	ew	no
2866 11	)	8	10	2000	0]	ld	no
6629	)	O	10	2000	U	Lu	110
12		6	2	2001	0	Ld	no
7473	3						
13		7	7	2000	ne	ew	no
3573 14	3	5	8	2016	n.	N. /	no
8150	)	J	0	2010	116	ew	no
15		9	2	2021	o <sup>1</sup>	ld	no
1559	)						
16		6	9	2021	0	ld	no
5075	5	10	-	2020			
17 5278	)	10	5	2020	ne	ew	no
18	,	4	6	1992	o]	ld	yes
4486	)	•	O .	1332		c u	yes
19		3	7	2021	0	ld	no
231							
	2++16	a2 r2 a0	hacetoragoroom	bacqu	octroom	price	catagony
0	attic 9005	956	hasstorageroom no	nasgu	estroom 7	7559081.5	category Luxury
1	8852	135	yes		9	5574642.1	
2	4748	654	no		10	8696869.3	
3	5792	807	yes		5	5154055.2	
4	1172	716	yes		9	9652258.1	
5	7117	240	no		7	7986665.8	-
6	281	384	yes		5	7607322.9	
7	129	726	no		9	6420823.1	
8	9056	892	yes		1	9244344.0	Luxury
9	1221	328	no		10	9515440.4	
10	3129	982	no		1	7653300.8	
11	435	512	no		7	8711426.0	
12	796	237	yes		3	6677649.1	
13	9556	918	yes		8	8460604.0	
14	6037	930	no		7	7614076.6	•
15	5111	957	yes		2	9272740.1	
16	3104	864	no		4	5984462.1	
17	1059	313	yes		6	5492532.0	
18 19	6919 1939	680 223	yes		1 8	7005572.2 5446398.1	
19	1939	223	no		O	3440396.1	. Milate
		i2=df_ka i2.head	ategori.drop('pı ( <mark>50</mark> )	rice',	axis= <mark>1</mark> )		
	square	meters	numberofrooms h	nasvard	haspool	floors	citycode \
0	3 4 5 6 7	75523	3	no	yes	63	9373
1		55712	58	no	yes	19	34457
2		86929	100	yes	no	11	98155

3	51522	3	no	no	61	9047
4	96470	74	yes	no	21	92029
5 6	79770	3	no	yes	69	54812
6	75985	60	yes	no	67	6517
7	64169	88	no	yes	6	61711
8	92383	12	no	no	78	71982
9	95121	46	no	yes	3	9382
10	76485	47	yes	no	9	90254
11	87060	27	no	yes	91	51803
12	66683	19	yes	yes	6	50801
13	84559	29	no	yes	69	53057
14	76091	38	yes	no	32	59451
15	92696	49	yes	no	38	74381
16	59800	47	no	yes	27	44815
17	54836	25	no	yes	53	64601
18	70021	52	yes	no	28	95678
19	54368	11	yes	yes	20	55761
20	63053	6	yes	yes	28	45312
21	64393	8	no	no	51	95335
22	93876	60	no	yes	70	5484
23	84016	15	yes	no	55	63595
24	89768	48	yes	yes	17	71000
25	58478	5	no	yes	35	5898
26	66621	48	no	no	89	52165
27	73314	43	no	yes	38	49895
28	59972	28	no	yes	18	32083
29	71591	20	yes	no	58	46834
30	67311	67	no	no	10	45626
31	61534	73	yes	no	97	22943
32	84091	50	no	yes	72	22718
33	51434	64	no	no	23	79754
34	78960	55	no		76	23408
35	81870	60	no	yes	100	58048
36	91559	36		yes yes	21	82521
30 37	72098	9	no no	-	67	91168
3 <i>1</i>	55232	23	no	yes	8	849
30 39		49	no	no	92	2423
	53735 71307		no	yes		
40	71397 65151	71 91	no	no	93	68199
41	65151 61484	81	yes	yes	3	66191
42	61484	91	yes	no	94	87015
43	57160	15	yes	yes	43	40786
44 45	55933	15	no	no	97	5800
45	81936	68	no	no	92	6143
46	99683	12	yes	no	77	18300
47	62887	45	no	yes	7	91125
48	73062	34	yes	yes	38	44770
49	84284	/6	no	no	39	55/23
49	84284	76	no mada is	no	39	55723

citypartrange numprevowners made isnewbuilt hasstormprotector

basement	\				
0	3	8	2005	old	yes
4313 1	6	8	2021	old	no
2937	Ü	· ·	2021	o ca	110
2	3	4	2003	new	no
6326	0	2	2012	<b></b>	
3 632	8	3	2012	new	yes
4	4	2	2011	new	yes
5414				-	,
5	10	5	2018	old	yes
8871		0	2000		
6	6	9	2009	new	yes
4878 7	3	9	2011	new	yes
3054	5	9	2011	Hew	yes
8	3	7	2000	old	no
7507					
9	7	9	1994	old	no
615	2	0	2000		
10 2860	2	9	2008	new	no
11	8	10	2000	old	no
6629	O .	10	2000	otu	110
12	6	2	2001	old	no
7473					
13	7	7	2000	new	no
3573	-	0	2016		
14 8150	5	8	2016	new	no
15	9	2	2021	old	no
1559	J	_	2021	o cu	110
16	6	9	2021	old	no
5075					
17	10	5	2020	new	no
5278	1	6	1002	old	\ <b>40</b> 6
18 4480	4	6	1992	old	yes
19	3	7	2021	old	no
231	J	ŕ		0 00	0
20	3	1	1997	old	yes
8414					-
21	4	1	1990	new	no
3835	2	1	1000	no.	V00
22 4086	2	1	1999	new	yes
23	1	7	2016	new	no
3284	_	•	_ J <b>_ J</b>		

24	6	9	1993	old	yes
2485 25	6	10	2016	old	no
8366				o tu	
26	10	1	1995	new	yes
5024 27	10	1	2018	old	yes
3281		_	2010	014	, 00
28	9	8	2021	new	yes
8384 29	7	4	1998	old	no
6486	,	7	1330	otu	110
30	3	3	1990	new	yes
6928 31	9	5	2001	old	no
9265	9	J	2001	otu	no
32	7	5	1993	old	no
2668	1.0	_	2012		
33 2080	10	2	2012	new	yes
34	8	4	2015	new	yes
7126	_	_			-
35 3632	3	8	2020	old	no
36	6	2	2007	old	yes
788		_			,
37	2	3	2014	new	no
9080 38	8	3	1991	old	no
1492	O .	5	1331	otu	110
39	4	7	2006	old	no
8654 40	3	10	1005	201	V05
3477	3	10	1995	new	yes
41	7	9	1991	old	no
3218	2	0	2012		
42 5486	8	8	2013	new	no
43	8	8	2002	old	no
4018					
44 7369	9	8	2001	new	yes
45	3	1	2011	new	no
6393					
46	5	8	2002	old	yes
4034 47	4	3	1993	new	no
292	7	3	1333	IICW	110
48	4	8	2016	old	no

982	3					
49		5	1	1998	old	no
450	0					
	attic		hasstorageroom	hasguestroom		
0	9005	956	no	7	Luxury	
1	8852	135	yes	9	Middle	
2	4748	654	no	10	Luxury	
3	5792	807	yes	5	Middle	
4	1172	716	yes	9	Luxury	
5	7117	240	no	7	Luxury	
6	281	384	yes	5	Luxury	
7	129	726	no	9	Middle	
8	9056	892	yes	1	Luxury	
9	1221	328	no	10	Luxury	
10	3129	982	no	1	Luxury	
11	435	512	no	7	Luxury	
12	796	237	yes	3	Middle	
13	9556	918	yes	8	Luxury	
14	6037	930	no	7	Luxury	
15	5111	957	yes	2	Luxury	
16	3104	864	no	4	Middle	
17	1059	313	yes	6	Middle	
18	6919	680	yes	1	Luxury	
19	1939	223	no	8	Middle	
20	6270	939	yes	8	Middle	
21	2403	559	no	6	Middle	
22	5991	494	yes	8	Luxury	
23	9879	641	no	2	Luxury	
24	108	864	no	7	Luxury	
25	4799	979	yes	7	Middle	
26	8103	388	yes	4	Middle	
27	5020	968	no	8	Luxury	
28	7226	226	yes	4	Middle	
29	3310	366	no	0	Luxury	
30	7808	774	yes	5	Middle	
31	8974	755	yes	6	Middle	
32	4669	766	no	8	Luxury	
33	9575	753	no	7	Middle	
34	5012	974	yes	0	Luxury	
35	5960	723	yes	3	Luxury	
36	4788	132	yes	8	Luxury	
37	9356	740	yes	9	Luxury	
38	5697	625	no	6	Middle	
39	9588	290	yes	8	Middle	
40	5530	342	no	2	Luxury	
41	9119	849	no	4	Middle	
42	3641	766	no	3	Middle	
43	4871	836	yes	2	Middle	

```
44
     6739
               686
                                                6
                                                    Middle
                               ves
45
     9082
               734
                                                0
                                                    Luxury
                                no
46
     2877
               787
                               yes
                                                6
                                                    Luxury
47
               675
                                                4
     744
                                                    Middle
                                no
48
     7174
               728
                                                0
                                                    Luxury
                               yes
49
     4877
               480
                                no
                                                    Luxury
df_kategori2['category'].value_counts()
category
Basic
          4344
          3065
Luxury
          2591
Middle
Name: count, dtype: int64
print("data null \n" ,df_kategori2.isnull().sum())
print("\ndata kosong \n",df_kategori2.empty)
print("\ndata nan \n" ,df_kategori2.isna().sum())
data null
                       0
 squaremeters
                      0
numberofrooms
                      0
hasyard
                      0
haspool
floors
                      0
                      0
citycode
citypartrange
                      0
numprevowners
                      0
                      0
made
                      0
isnewbuilt
                      0
hasstormprotector
basement
                      0
                      0
attic
                      0
garage
                      0
hasstorageroom
                      0
hasguestroom
                      0
category
dtype: int64
data kosong
False
data nan
                       0
 squaremeters
numberofrooms
                      0
                      0
hasyard
                      0
haspool
                      0
floors
                      0
citycode
citypartrange
                      0
```

```
0
numprevowners
                      0
made
isnewbuilt
                      0
hasstormprotector
                      0
basement
                      0
                      0
attic
                      0
garage
hasstorageroom
                      0
                      0
hasguestroom
category
                      0
dtype: int64
median_chole = df_kategori2['category'].median()
print(median chole)
df kategori2['Cholesterol'] =
df_kategori2['Cholesterol'].fillna(median_chole)
222.0
print(df kategori2['category'].value counts())
print("data null \n" ,df_kategori2.isnull().sum())
print("\ndata kosong \n",df_kategori2.empty)
print("\ndata nan \n" ,df_kategori2.isna().sum())
category
Basic
          4344
          3065
Luxury
Middle
          2591
Name: count, dtype: int64
data null
 squaremeters
                       0
                      0
numberofrooms
                      0
hasyard
                      0
haspool
                      0
floors
citycode
                      0
                      0
citypartrange
                      0
numprevowners
                      0
made
isnewbuilt
                      0
                      0
hasstormprotector
basement
                      0
attic
                      0
                      0
garage
                      0
hasstorageroom
                      0
hasguestroom
                      0
category
```

```
dtype: int64
data kosong
False
data nan
                      0
squaremeters
numberofrooms
                     0
hasyard
                     0
                     0
haspool
                     0
floors
                     0
citycode
                     0
citypartrange
numprevowners
                     0
made
                     0
isnewbuilt
                     0
                     0
hasstormprotector
                     0
basement
                     0
attic
                     0
garage
                     0
hasstorageroom
                     0
hasquestroom
                     0
category
dtype: int64
print("Sebelum drop missing value", df_kategori2.shape)
df_kategori2 = df_kategori2.dropna(how="any", inplace=False)
print("Sesudah drop missing value" , df kategori2.shape)
Sebelum drop missing value (10000, 17)
Sesudah drop missing value (10000, 17)
df kategori2['category'].value counts()
category
Basic
          4344
Luxury
          3065
Middle
          2591
Name: count, dtype: int64
print("Sebelum drop data dengan gender Bi", df kategori2.shape)
df kategori2=df kategori2[df kategori2['category']!='Bi']
print("Sesudah drop data dengan gender Bi" , df_kategori2.shape)
Sebelum drop data dengan gender Bi (10000, 17)
Sesudah drop data dengan gender Bi (10000, 17)
df kategori2.head(20)
    squaremeters numberofrooms hasyard haspool floors
                                                          citycode \
0
           75523
                                                      63
                                                               9373
                                      no
                                             yes
```

1	55712	58	no	yes	19	34457
2 3	86929	100	yes	no	11	98155
	51522	3	no	no	61	9047
4 5 6	96470 79770	74 3	yes no	no	21 69	92029 54812
6	75985	60	yes	yes no	67	6517
7	64169	88	no	yes	6	61711
8	92383	12	no	no	78	71982
9	95121	46	no	yes	3	9382
10	76485	47	yes	no	9	90254
11	87060	27	no	yes	91	51803
12	66683	19	yes	yes	6	50801
13	84559	29	no	yes	69	53057
14	76091	38	yes	no	32	59451
15	92696	49	yes	no	38	74381
16	59800	47	no	yes	27	44815
17	54836	25	no	yes	53	64601
18	70021	52	yes	no	28	95678
19	54368	11	yes	yes	20	55761
citv	partrange	numprevowners	made i	isnewbuilt	hasstorn	nprotector
basement	\					
0	3	8	2005	old		yes
4313						
1	6	8	2021	old		no
2937						
2	3	4	2003	new		no
6326	0	2	2012			
3	8	3	2012	new		yes
632 4	4	2	2011	now		VOC
5414	7	۷	2011	new		yes
5	10	5	2018	old		yes
8871	10	3	2010	Oca		yes
6	6	9	2009	new		yes
4878	-			-		,
7	3	9	2011	new		yes
3054						
8	3	7	2000	old		no
7507						
9	7	9	1994	old		no
615		•	2000			
10	2	9	2008	new		no
2860	0	10	2000	-1 d		
11 6629	8	10	2000	old		no
12	6	2	2001	old		no
7473	Ü	Z	2001	otu		110
13	7	7	2000	new		no
_5	,	7	2000	TICW		110

```
3573
14
                 5
                                    2016
                                                 new
                                                                     no
8150
                 9
                                 2
                                                 old
15
                                    2021
                                                                     no
1559
                 6
16
                                 9
                                    2021
                                                 old
                                                                     no
5075
17
                10
                                 5
                                    2020
                                                 new
                                                                     no
5278
18
                                 6
                                    1992
                                                 old
                                                                    yes
4480
                 3
19
                                 7
                                    2021
                                                 old
                                                                     no
231
           garage hasstorageroom
                                    hasquestroom category
    attic
0
     9005
               956
                                                7
                                                    Luxury
                                no
                                                9
1
     8852
               135
                                                    Middle
                               ves
2
     4748
               654
                                               10
                                                    Luxury
                                no
3
     5792
               807
                                                5
                                                    Middle
                               yes
4
     1172
               716
                                                9
                               yes
                                                    Luxury
5
                                                7
     7117
               240
                                no
                                                    Luxury
6
                                                5
      281
               384
                               yes
                                                    Luxury
7
                                                9
      129
               726
                                                    Middle
                                no
8
     9056
                                                1
               892
                               yes
                                                    Luxury
9
     1221
               328
                                no
                                               10
                                                    Luxury
10
     3129
               982
                                                1
                                                    Luxury
                                no
                                                7
11
      435
               512
                                no
                                                    Luxury
12
      796
               237
                                                3
                                                    Middle
                               yes
13
                                                8
     9556
               918
                                                    Luxury
                               yes
                                                7
14
     6037
               930
                                                    Luxury
                                no
15
                                                2
     5111
               957
                               yes
                                                    Luxury
16
     3104
               864
                                                4
                                                    Middle
                                no
17
     1059
               313
                                                6
                                                    Middle
                               yes
18
     6919
               680
                               yes
                                                1
                                                    Luxury
19
     1939
               223
                                                    Middle
                                no
print("Sebelum Pengecekan data Duplikat", df kategori2.shape)
df kategori3 = df kategori2.drop duplicates(keep='last')
print("Sesudah Pengecekan data Duplikat" , df kategori3.shape)
Sebelum Pengecekan data Duplikat (10000, 17)
Sesudah Pengecekan data Duplikat (10000, 17)
from sklearn.model selection import train test split
x regress = df kategori3.drop(columns=['category'], axis = 1)
y regress = df kategori3['category']
x_train_category, x_test_category, y_train_category, y_test_category =
train test split(x regress, y regress, test size= 0.25, random state =
```

```
78)
print(x train category.shape)
print(x test category.shape)
(7500, 16)
(2500, 16)
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make column transformer
kolom kategori=['hasyard','haspool','isnewbuilt','hasstormprotector','
hasstorageroom']
transform=make column transformer(
    (OneHotEncoder(), kolom kategori), remainder='passthrough'
)
x train category enc = transform.fit transform(x train category)
x test category enc = transform.fit transform(x test category)
df train enc = pd.DataFrame(x train category enc,
columns=transform.get_feature_names_out())
df test enc = pd.DataFrame(x test category enc,
columns=transform.get feature names out())
df train enc.head(20)
df test enc.head(20)
    onehotencoder__hasyard_no
                                onehotencoder__hasyard_yes \
0
                           1.0
                                                         0.0
                                                         1.0
1
                           0.0
2
                           1.0
                                                         0.0
3
                           1.0
                                                         0.0
4
                           0.0
                                                         1.0
5
                                                         1.0
                           0.0
6
                           0.0
                                                         1.0
7
                           0.0
                                                         1.0
8
                                                         1.0
                           0.0
9
                           1.0
                                                         0.0
10
                           0.0
                                                         1.0
11
                           0.0
                                                         1.0
12
                           1.0
                                                         0.0
13
                           0.0
                                                         1.0
14
                           1.0
                                                         0.0
15
                           0.0
                                                         1.0
                           1.0
16
                                                         0.0
17
                           0.0
                                                         1.0
18
                           0.0
                                                         1.0
19
                           1.0
                                                         0.0
```

```
onehotencoder__haspool_no
                                  onehotencoder__haspool_yes \
0
                             \overline{0}.0
                                                             1.0
1
                             0.0
                                                             1.0
2
                             1.0
                                                             0.0
                             0.0
                                                             1.0
4
                                                             0.0
                             1.0
5
                             0.0
                                                             1.0
6
                             0.0
                                                             1.0
7
                             0.0
                                                             1.0
8
                             1.0
                                                             0.0
9
                             1.0
                                                             0.0
10
                             0.0
                                                             1.0
11
                             1.0
                                                             0.0
12
                             1.0
                                                             0.0
13
                             1.0
                                                             0.0
14
                             1.0
                                                             0.0
15
                             1.0
                                                             0.0
16
                             1.0
                                                             0.0
                                                             1.0
17
                             0.0
18
                             1.0
                                                             0.0
19
                                                             0.0
                             1.0
    onehotencoder isnewbuilt new
                                       onehotencoder isnewbuilt old
0
                                                                     1.0
1
                                  0.0
                                                                     1.0
2
                                  0.0
                                                                     1.0
3
                                  1.0
                                                                     0.0
4
                                  1.0
                                                                     0.0
5
                                  0.0
                                                                     1.0
6
                                  0.0
                                                                     1.0
7
                                  0.0
                                                                     1.0
8
                                  1.0
                                                                     0.0
9
                                 1.0
                                                                     0.0
10
                                  0.0
                                                                     1.0
11
                                  0.0
                                                                     1.0
12
                                  1.0
                                                                     0.0
13
                                  1.0
                                                                     0.0
14
                                  0.0
                                                                     1.0
                                  1.0
15
                                                                     0.0
16
                                 1.0
                                                                     0.0
17
                                  1.0
                                                                     0.0
18
                                  1.0
                                                                     0.0
19
                                  1.0
                                                                     0.0
    onehotencoder hasstormprotector no
onehotencoder hasstormprotector yes \
                                        1.0
0.0
                                        0.0
```

1.0	
2	1.0
0.0	1.0
3 0.0	1.0
4	1.0
0.0	
5 1.0	0.0
6	0.0
6 1.0	
7	0.0
1.0	0.0
8 1.0	0.0
9	1.0
0.0	
10	1.0
0.0 11	0.0
1.0	
12	0.0
1.0	1.0
13 0.0	1.0
14	1.0
0.0	
15	1.0
0.0 16	1.0
0.0	110
17	1.0
0.0	1.0
18 0.0	1.0
19	1.0
0.0	
onehotencoderhasstorageroom_	no
onehotencoderhasstorageroom_yes	\
0 1	.0
0.0	
1 0	. 0
2 0	. 0
1.0	
3	. 0
1.0 4 1	. 0
7	. 0

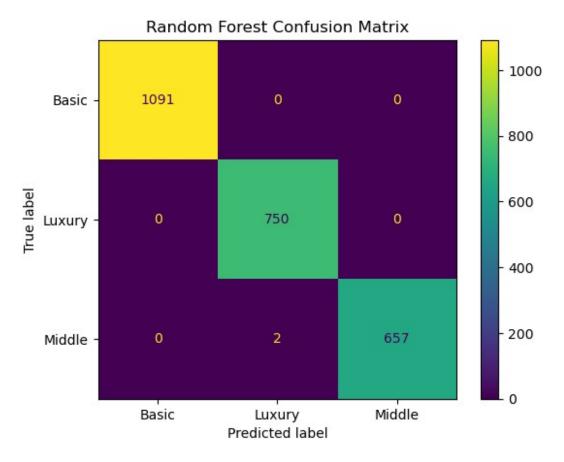
0.0 5	• • •		0.0			
1.0			0.0			
1.0			0.0			
1.0			0.0			
1.0						
9			0.0			
10 1.0			0.0			
11 0.0			1.0			
12 1.0			0.0			
13 1.0			0.0			
14			1.0			
15 1.0	• • •		0.0			
16	• • •		0.0			
1.0	• • •		1.0			
0.0 18			0.0			
1.0 19			0.0			
1.0						
rema	<pre>remainder_ indercit</pre>	_numberofrooms ycode \	remainder_	_floors		
Θ		73.0		13.0	42855.0	
1		95.0		3.0	75381.0	
2		39.0		8.0	91674.0	
3		47.0		63.0	58471.0	
4		64.0		83.0	30779.0	
5		91.0		67.0	65183.0	
6		48.0		70.0	21012.0	
7		93.0		35.0	12062.0	

8	74.0	14.0	76662.0
9	72.0	76.0	87732.0
10	18.0	66.0	38920.0
11	37.0	26.0	96016.0
12	76.0	95.0	76985.0
13	26.0	99.0	38185.0
14	69.0	45.0	88591.0
15	9.0	25.0	33740.0
16	29.0	58.0	13202.0
17	91.0	43.0	93072.0
18	68.0	62.0	97608.0
19	48.0	53.0	34588.0
<pre>remaindercitypart remaindermade \</pre>	range remainder <u></u> r	numprevowners	
0 2015.0	9.0	6.0	
1	5.0	6.0	
2003.0	2.0	2.0	
2009.0	10.0	1.0	
1990.0 4	6.0	4.0	
1992.0 5	1.0	10.0	
2019.0 6	5.0	3.0	
2007.0	10.0	9.0	
1998.0 8	9.0	6.0	
2004.0 9	2.0	8.0	
2017.0			
10 2007.0	7.0	2.0	
11	3.0	6.0	

2000.0 12		4.0	7.0	
2009.0 13		7.0	7.0	
1994.0 14		1.0	8.0	
1995.0 15		2.0	1.0	
2009.0 16		6.0	5.0	
1993.0 17		1.0	7.0	
2012.0 18		1.0	9.0	
2001.0 19		9.0	6.0	
2003.0				
rema 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	inderbasement 2560.0 6810.0 1477.0 1730.0 594.0 6824.0 5861.0 2064.0 9501.0 6414.0 4949.0 1261.0 3586.0 7053.0 3429.0 7818.0 6740.0 1528.0 131.0 907.0	remainderattic 6823.0 1391.0 3153.0 7967.0 8310.0 7141.0 2750.0 2720.0 9579.0 6111.0 6205.0 6017.0 1109.0 8363.0 1494.0 5829.0 4961.0 336.0 3418.0	remaindergarage 239.0 556.0 952.0 722.0 898.0 956.0 652.0 315.0 768.0 613.0 185.0 850.0 472.0 827.0 827.0 876.0 567.0 484.0 893.0 656.0 162.0	
rema 0 1 2 3 4 5 6	( ) ( )	oom 0.0 3.0 2.0 0.0 5.0 6.0 9.0		

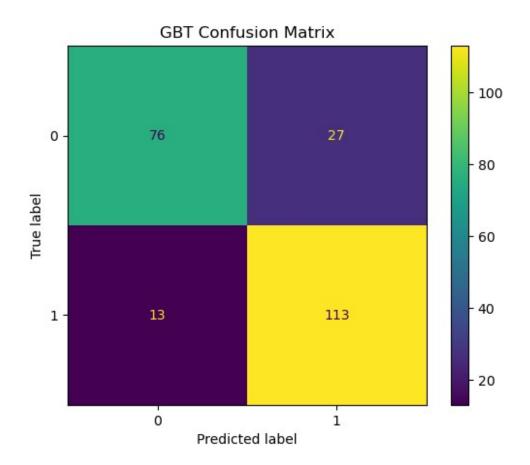
```
8
                        1.0
9
                        7.0
10
                        6.0
11
                       10.0
12
                        7.0
13
                        3.0
14
                        3.0
15
                        0.0
16
                        5.0
17
                        8.0
18
                       10.0
19
                        5.0
[20 rows x 21 columns]
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.feature_selection import SelectKBest, SelectPercentile
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.model selection import GridSearchCV, StratifiedKFold
from sklearn.metrics import confusion matrix, classification report,
ConfusionMatrixDisplay
import numpy as np
pipe RF = [
    ('data scaling', StandardScaler()),
    ('feature select', SelectKBest()),
    ('clf', RandomForestClassifier(random state=78,
class weight='balanced'))
params grid RF = [
    {
        'data scaling': [StandardScaler()],
        'feature select k': np.arange(2, 6),
        'clf max depth': np.arange(4, 5),
        'clf n estimators': [100, 150]
    },
{
        'data scaling': [StandardScaler()],
        'feature select': [SelectPercentile()],
        'feature select__percentile': np.arange(20, 50),
        'clf max_depth': np.arange(4, 5),
        'clf n estimators': [100, 150]
    },
        'data scaling': [MinMaxScaler()],
        'feature select k': np.arange(2, 6),
        'clf max depth': np.arange(4, 5),
```

```
'clf n estimators': [100, 150]
    },
        'data scaling': [MinMaxScaler()],
        'feature select': [SelectPercentile()],
        'feature select__percentile': np.arange(20, 50),
        'clf max depth': np.arange(4, 5),
        'clf n estimators': [100, 150]
    }
]
estimator RF = Pipeline(pipe RF)
GSCV_RF = GridSearchCV(estimator_RF, params grid RF,
cv=StratifiedKFold(n splits=5))
GSCV RF.fit(x train category enc, y train category)
print("GSCV training finished")
GSCV training finished
print("CV Score : {}".format(GSCV RF.best score ))
print("Test Score:
{}".format(GSCV RF.best estimator .score(x test category enc,
y test category)))
print("Best model:", GSCV RF.best estimator )
mask = GSCV RF.best estimator .named steps['feature
select'].get support()
print("Best features:", df train enc.columns[mask])
RF pred = GSCV RF.predict(x test category enc)
import matplotlib.pyplot as plt
cm = confusion_matrix(y_test_category, RF_pred,
labels=GSCV_RF.classes_)
disp = ConfusionMatrixDisplay(confusion matrix=cm,
display_labels=GSCV_RF.classes_)
disp.plot()
plt.title("Random Forest Confusion Matrix")
plt.show()
print("Classification report RF: \n",
classification report(y test category, RF pred))
```



Classification			_	
	precision	recall	f1-score	support
Basic	1.00	1.00	1.00	1091
Luxury	1.00	1.00	1.00	750
Middle	1.00	1.00	1.00	659
				0.500
accuracy			1.00	2500

```
1.00
                                       1.00
                                                 2500
                   1.00
   macro avq
                             1.00
weighted avg
                   1.00
                                       1.00
                                                 2500
print("CV Score : {}".format(GSCV GBT.best score ))
print("Test Score:
{}".format(GSCV GBT.best estimator .score(x test enc, y test)))
print("Best model:", GSCV GBT.best estimator )
mask =
GSCV GBT.best estimator .named steps['feat select'].get support()
print("Best features:", df train enc.columns[mask])
RF pred = GSCV GBT.predict(x test enc)
import matplotlib.pyplot as plt
cm = confusion matrix(y test, RF pred, labels=GSCV GBT.classes )
disp = ConfusionMatrixDisplay(confusion matrix=cm,
display labels=GSCV_GBT.classes_)
disp.plot()
plt.title("GBT Confusion Matrix")
plt.show()
print("Classification report GBT: \n", classification report(y test,
RF_pred))
CV Score: 0.854437744631334
Test Score: 0.8253275109170306
Best model: Pipeline(steps=[('feat select',
SelectPercentile(percentile=48)),
                ('clf',
                 GradientBoostingClassifier(learning rate=0.01,
max depth=4,
                                            n estimators=150,
                                             random state=47))])
Best features: Index(['onehotencoder__Sex_M',
'onehotencoder ChestPainType ASY',
       'onehotencoder__ChestPainType_ATA',
'onehotencoder ExerciseAngina N',
       'onehotencoder ExerciseAngina Y',
'onehotencoder ST Slope Flat',
       'onehotencoder ST Slope Up', 'remainder Age',
'remainder MaxHR',
       'remainder Oldpeak'],
      dtype='object')
```



Classification	•				
	precision	recall	f1-score	support	
0 1	0.85 0.81	0.74 0.90	0.79 0.85	103 126	
1	0.01	0.90	0.05	120	
accuracy macro avg weighted avg	0.83 0.83	0.82 0.83	0.83 0.82 0.82	229 229 229	
#import Library evaluasi from sklearn.fe from sklearn.fe from sklearn.me from sklearn.me from sklearn.me ConfusionMatrix import numpy as #buat rancangan	reprocessing eature_select inear_model i odel_selectio ipeline impore etrics import kDisplay s np	import M ion impo mport Lo n import t Pipeli classif	inMaxScaler rt SelectPe gisticRegre GridSearch ne ication_rep	r, Standardercentile, ession aCV, Strat	dScaler SelectKBest ifiedKFold usion_matrix,

```
pipe logreg = Pipeline(steps=[
    ('scale', MinMaxScaler()),
    ('feat_select', SelectKBest()),
    ('clf', LogisticRegression(class weight='balanced',
max iter=1000)
1)
#buat parameter grid untuk step feature selection dan classifier
params grid logreg = [
    'scale': [MinMaxScaler()],
    'feat select k':np.arange(2,6),
    'clf penalty': ['l2', 'none'],
    'clf C':[0.1, 1, 10],
    'clf__solver': ['lbfgs', 'saga']
    },
    'scale': [MinMaxScaler()],
    'feat select': [SelectPercentile()],
    'feat select percentile':np.arange(20,50),
    'clf__penalty': ['l2', 'none'],
    'clf__C':[0.1, 1, 10],
    'clf_solver': ['lbfgs', 'saga']
    },
    'scale': [StandardScaler()],
    'feat select k':np.arange(2,6),
    'clf__penalty': ['l2', 'none'],
    'clf C':[0.1, 1, 10],
    'clf solver': ['lbfgs', 'saga']
    },
    'scale': [StandardScaler()],
    'feat_select':[SelectPercentile()],
    'feat select percentile': np.arange(20,50),
    'clf penalty': ['l2', 'none'],
    'clf__C':[0.1, 1, 10],
    'clf solver': ['lbfgs', 'saga']
    }
1
#muat pipeline dan parameter grid ke dalam objek GridSearchCV dengan
Stratified 5-fold CV
SKF = StratifiedKFold(n splits=5, shuffle=True, random state=78)
GSCV LogReg = GridSearchCV(pipe logreg, params grid logreg, cv=SKF)
#jalankan objek GSCV untuk melatih model dengan train set menggunakan
fungsi fit
GSCV LogReg.fit(x train category enc, y train category)
print("GSCV training finished")
```

```
GSCV training finished
c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
model selection\ validation.py:547: FitFailedWarning:
2040 fits failed out of a total of 4080.
The score on these train-test partitions for these parameters will be
set to nan.
If these failures are not expected, you can try to debug them by
setting error score='raise'.
Below are more details about the failures:
2040 fits failed with the following error:
Traceback (most recent call last):
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
model_selection\_validation.py", line 895, in _fit_and_score
   estimator.fit(X train, y train, **fit params)
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
base.py", line 1474, in wrapper
    return fit method(estimator, *args, **kwargs)
           ^^^<del>^</del>^^^^^^^
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
pipeline.py", line 475, in fit
    self. final estimator.fit(Xt, y, **last step params["fit"])
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
base.py", line 1467, in wrapper
   estimator. validate params()
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
base.py", line 666, in validate params
    validate parameter constraints(
  File "c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
utils\_param_validation.py", line 95, in
validate parameter constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'penalty'
parameter of LogisticRegression must be a str among {'l2',
'elasticnet', 'll'} or None. Got 'none' instead.
 warnings.warn(some fits failed message, FitFailedWarning)
c:\Users\YAWE MANOLO\anaconda3\Lib\site-packages\sklearn\
model selection\ search.py:1051: UserWarning: One or more of the test
0.8604
          0.85933333
 0.84426667 0.8444
                             nan
                                        nan
                                                  nan
                                        nan 0.86973333 0.87013333
       nan
                  nan
                             nan
 0.86706667 0.8672
                      0.87013333 0.87026667 0.8672
                                                      0.8672
       nan
                  nan
                             nan
                                        nan
                                                  nan
                                                             nan
                  nan 0.8704
                                 0.87013333  0.86946667  0.86933333
       nan
 0.87026667 0.87026667 0.8692
                                 0.8692
                                                  nan
```

```
nan
                           nan
                                     nan
      nan
                                               nan
                                                         nan
0.8444
          0.84453333  0.84453333  0.84453333  0.84453333  0.84453333
0.86013333 \ 0.86013333 \ 0.86013333 \ 0.87213333 \ 0.87213333 \ 0.87226667
0.87226667 0.87226667 0.87226667 0.87226667 0.87146667 0.87146667
0.87146667 \ 0.87146667 \ 0.87146667 \ 0.87106667 \ 0.87106667 \ 0.87133333
0.87133333 \ 0.87133333 \ 0.87106667 \ 0.87106667 \ 0.87106667 \ 0.87106667
                    0.8444
          0.8444
                              0.8444
                                        0.84453333   0.84453333
0.8444
0.8604
          0.86013333 0.86
                              0.87226667 0.87213333 0.87226667
0.87226667 0.87226667 0.87226667 0.87226667 0.87106667 0.87133333
          0.87133333  0.87133333  0.8708
                                        0.87066667 0.87106667
0.87093333  0.87106667  0.87133333  0.87093333  0.8708
                                                   0.87106667
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
                nan
                                     nan
                                               nan
                                                         nan
      nan
                           nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
      nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
0.86693333 0.8672
                    0.8672
                              0.8672
                                        0.8672
                                                   0.8672
0.87706667 0.87706667 0.87706667 0.8836
                                        0.8836
                                                   0.88333333
0.88346667 0.88346667 0.88346667 0.8832
                                        0.8832
                                                   0.8832
0.8832
          0.8832
                    0.8832
                              0.8832
                                        0.8832
                                                   0.8832
0.8672
          0.8672
                    0.8672
                              0.8672
                                        0.8672
                                                   0.8672
0.87706667 0.87706667 0.87706667 0.8836
                                        0.8836
                                                   0.88306667
0.88306667 0.88306667 0.88306667 0.88306667 0.88346667 0.88346667
nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                          nan
                                     nan
                                               nan
                                                         nan
      nan
                nan
                           nan
                                     nan
                                               nan
                                                         nan
0.8852
0.8784
          0.8784
                    0.8784
                                        0.8852
                                                   0.88506667
0.88506667 0.88506667 0.88506667 0.88506667 0.88466667 0.88466667
0.88466667 0.88466667 0.88466667 0.8844
                                        0.8844
                                                   0.88453333
0.8692
          0.8692
                    0.8692
                              0.8692
                                        0.8692
                                                   0.8692
0.8784
          0.8784
                    0.8784
                              0.8852
                                        0.8852
                                                   0.8852
0.8852
          0.8852
                    0.8852
                              0.8852
                                        0.8848
                                                   0.8848
```

```
0.8848 0.8848
                              0.8844
                                         0.8844
0.8848
                                                   0.88453333
                              0.8844
0.88453333  0.88453333  0.8844
                                         0.8844
                                                   0.8844
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
                 nan
                           nan
                                     nan
                                                          nan
      nan
                                                nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
                                     nan
                                                nan
      nan
                 nan
                           nan
                                                          nan
                                                nan
      nan
                 nan
                           nan
                                     nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
0.8692
          0.87013333  0.86733333  0.86746667  0.8692
                                                   0.87013333
          0.8676
0.8672
                           nan
                                     nan
                                                nan
                                     nan 0.87013333 0.8704
                           nan
      nan
                nan
0.8696
          0.86946667 0.87
                               0.87026667 0.86946667 0.86946667
                 nan
                                     nan
                                                nan
      nan
                           nan
                                                          nan
                                                   0.87
                 nan 0.87026667 0.87026667 0.87
      nan
0.87026667 0.87026667 0.87
                               0.87
                                                nan
                                                          nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
0.86733333  0.86746667  0.86746667  0.86746667  0.86746667
0.86746667 0.86746667 0.86893333 0.88386667 0.88386667 0.88373333
0.88386667 0.88386667 0.88386667 0.8836
                                         0.8836
                                                   0.8836
                    0.88266667 0.88266667 0.88266667 0.88266667
0.8836
          0.8836
                    0.86733333  0.86746667  0.86746667  0.8676
0.86746667 0.8672
0.86746667 0.86746667 0.86893333 0.884
                                         0.88386667 0.88373333
0.88386667 0.88386667 0.88386667 0.88346667 0.88346667
0.88346667 0.8836
                    0.88266667 0.88266667 0.88253333 0.8828
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
                 nan
                                     nan
                                                nan
      nan
                           nan
                                                          nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
      nan
                 nan
                           nan
                                     nan
                                                          nan
                                                nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
                                                          nan
      nan
                 nan
                           nan
                                     nan
                                                nan
          0.86946667 0.86946667 0.86946667 0.86946667 0.86946667
0.8696
0.86946667 0.86946667 0.8712
                               0.8852
                                         0.8852
                                                   0.88533333
0.88533333 0.88533333 0.88533333 0.88506667 0.88506667
0.88506667  0.88506667  0.88506667  0.88453333  0.88453333  0.88466667
0.88466667 0.88466667 0.88453333 0.88453333 0.88453333 0.88453333
0.86946667 0.86946667 0.86946667 0.86946667 0.86946667
0.86946667 0.86946667 0.8712
                           0.88533333 0.88533333 0.88533333
0.88506667 0.88506667 0.88506667 0.8844 0.88453333 0.88466667
```

```
0.88466667 0.88453333 0.88466667 0.88466667 0.88466667 0.88466667
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
                     nan
                                nan
                                            nan
                                                         nan
        nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
                     nan
                                 nan
                                                                     nan
        nan
                                            nan
                                                         nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
 0.87
             0.87
                         0.87
                                     0.87
                                                 0.87
                                                             0.87
 0.87
             0.87
                         0.8716
                                     0.886
                                                 0.886
                                                             0.88613333
 0.88613333  0.88613333  0.88613333  0.88613333  0.886
                                                             0.886
 0.886
             0.886
                         0.886
                                     0.8852
                                                 0.8852
                                                             0.88506667
 0.88506667 \ 0.88506667 \ 0.88493333 \ 0.88493333 \ 0.88493333 \ 0.88493333
 0.87
             0.87
                         0.87
                                     0.87
                                                 0.87
                                                             0.87
 0.87
                         0.8716
             0.87
                                     0.886
                                                 0.886
                                                             0.886
 0.886
             0.886
                         0.886
                                                 0.886
                                                             0.886
                                     0.886
 0.886
             0.886
                         0.886
                                     0.8852
                                                 0.8852
                                                             0.88506667
 0.88506667 0.88506667 0.8848
                                     0.8848
                                                 0.8848
                                                             0.8848
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                nan
                                                         nan
                                            nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                nan
                                            nan
                                                         nan
                                                                     nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
                     nan
                                            nan
                                                                     nan
        nan
                                 nan
                                                         nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan
                     nan
                                            nan
                                                         nan
                                                                     nan
        nan
                                 nan
                     nan
                                            nan
                                                         nan
                                                                     nan
        nan
                                 nan
        nan
                     nan
                                 nan
                                            nan
                                                         nan
                                                                     nan]
  warnings.warn(
#tampilkan skor cross-validation
print("CV Score : {}".format(GSCV LogReg.best score ))
#tampilkan skor model terbaik GSCV pada test set
print("Test Score:
{}".format(GSCV LogReg.best_estimator_.score(x_test_category_enc,
y test category)))
#tampilkan best model dan best features
print("Best model:", GSCV LogReg.best estimator )
mask =
GSCV LogReg.best estimator .named steps['feat select'].get support()
print("Best features:", df_train_enc.columns[mask])
#buat prediksi dari test set
LogReg pred = GSCV LogReg.predict(x test category enc)
import matplotlib.pyplot as plt
#buat confusion matrix
```

```
cm = confusion matrix(y test category, LogReg pred,
labels=GSCV LogReg.classes )
#buat confusion matrix display
disp = ConfusionMatrixDisplay(confusion matrix=cm,
display labels=GSCV LogReg.classes )
disp.plot()
plt.title("Logistic Regression Confusion Matrix")
plt.show()
#tampilkan classification report
print("Classification report Logistic Regression:\n",
classification report(y test category, LogReg pred))
NameError
                                          Traceback (most recent call
last)
Cell In[1], line 2
      1 #tampilkan skor cross-validation
----> 2 print("CV Score : {}".format(GSCV_LogReg.best_score_))
      3 #tampilkan skor model terbaik GSCV pada test set
      4 print("Test Score:
{}".format(GSCV LogReg.best estimator .score(x test category enc,
y test category)))
NameError: name 'GSCV LogReg' is not defined
import pickle
with open('RF_heartDisease model.pkl','wb') as r:
    pickle.dump((GSCV RF), r)
print("Model RF berhasil disimpan")
Model RF berhasil disimpan
```

```
import pandas as pd
import numpy as np
#Baca dataset dengan menggunakan function read csv dari pandas
df kategori=pd.read csv('Dataset UTS Gasal 2425.csv')
df kategori.head(10000)
                      numberofrooms hasyard haspool
                                                        floors
                                                                  citycode \
      squaremeters
0
              75523
                                   3
                                           no
                                                   yes
                                                             63
                                                                      9373
1
              55712
                                  58
                                                             19
                                                                     34457
                                           no
                                                   yes
2
              86929
                                 100
                                                             11
                                                                     98155
                                          yes
                                                    no
3
              51522
                                   3
                                                             61
                                                                      9047
                                           no
                                                    no
4
              96470
                                  74
                                                             21
                                                                     92029
                                          yes
                                                    no
                 . . .
                                  . . .
                                           . . .
                                                   . . .
                                                            . . .
                                                                       . . .
9995
                341
                                                                      1960
                                  83
                                           no
                                                    no
                                                              8
              21514
9996
                                   5
                                                             11
                                                                     91373
                                           no
                                                   yes
9997
               1726
                                  89
                                                              5
                                                                     73133
                                           no
                                                   yes
9998
              44403
                                  29
                                                             12
                                                                     34606
                                          yes
                                                   yes
               1440
9999
                                  84
                                           no
                                                    no
                                                             49
                                                                     18412
                                       made isnewbuilt hasstormprotector
      citypartrange
                       numprevowners
0
                    3
                                    8
                                        2005
                                                     old
                                                                         yes
1
                    6
                                        2021
                                                     old
                                                                           no
2
                    3
                                        2003
                                                     new
                                                                           no
3
                                     3
                                        2012
                                                     new
                                                                         yes
                                        2011
                                                     new
                                                                          yes
                                                                          . . .
9995
                                        1993
                                                     new
                                                                         yes
                                                     old
9996
                                        1999
                                                                           no
9997
                                                     old
                                        2009
                                                                         yes
9998
                    9
                                        1990
                                                     old
                                                                          yes
9999
                    6
                                    10
                                        1994
                                                     new
                                                                           no
      basement
                 attic garage hasstorageroom
                                                   hasquestroom
                                                                       price
category
           4313
                   9005
                             956
                                               no
                                                                  7559081.5
```

Luxury						
1	2937	8852	135	yes	9	5574642.1
Middle						
2	6326	4748	654	no	10	8696869.3
Luxury	622	F700	007		_	E1E40EE 2
3 Middle	632	5792	807	yes	5	5154055.2
4	5414	1172	716	yes	9	9652258.1
Luxury	7414	11/2	710	y <del>c</del> 3	9	9032230.1
9995	2366	4016	229	yes	5	35371.3
Basic						
9996	2584	5266	787	no	3	2153602.9
Basic	0211	1000	210		4	176425 0
9997 Basic	9311	1698	218	no	4	176425.9
9998	9061	1742	230	no	0	4448474.0
Basic	3001	1,72	250	110	U	777077710
9999	8485	2024	278	yes	6	146708.4
Basic				•		

## [10000 rows x 18 columns]

df\_kategori2=df\_kategori.drop('price',axis=1)
df\_kategori2.head(50)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	squaremeters 75523 55712 86929 51522 96470 79770 75985 64169 92383 95121 76485 87060 66683 84559 76091 92696 59800 54836 70021	numberofrooms 3 58 100 3 74 3 60 88 12 46 47 27 19 29 38 49 47 25 52	no no yes no yes no no yes no yes no yes yes no yes	yes yes no no no yes no yes no yes yes yes yes yes yes no no yes yes	floors 63 19 11 61 21 69 67 6 78 3 9 91 6 69 32 38 27 53 28 20	citycode 9373 34457 98155 9047 92029 54812 6517 61711 71982 9382 90254 51803 50801 53057 59451 74381 44815 64601 95678 55761	
			-				
20	63053	6	yes	yes	28	45312	
21	64393	8	no	no	51	95335	

22	93876	60	no	yes	70	5484
23	84016	15	yes		55	63595
24	89768	48	yes		17	71000
25	58478	5	no	=	35	5898
26	66621	48	no		89	52165
27	73314	43	no	_	38	49895
28 29	59972 71591	28 20	no	_	18 58	32083 46834
30	67311	67	yes no		10	45626
31	61534	73	yes		97	22943
32	84091	50	no		72	22718
33	51434	64	no		23	79754
34	78960	55	no		76	23408
35	81870	60	no	_	100	58048
36	91559	36	no	_	21	82521
37	72098	9	no	_	67	91168
38	55232	23	no	no	8	849
39	53735	49	no	yes	92	2423
40	71397	71	no	no	93	68199
41	65151	81	yes	yes	3	66191
42	61484	91	yes		94	87015
43	57160	15	yes	_	43	40786
44	55933	15	no		97	5800
45	81936	68	no		92	6143
46	99683	12	yes		77	18300
47 48	62887	45 34	no		7 38	91125
49	73062 84284	76	yes no	_	30 39	44770 55723
49	04204	70	110	110	29	33723
citypa	rtrange	numprevowners	made	isnewbuilt	hasstori	mprotector
	\	•				•
0	3	8	2005	old		yes
4313						
1	6	8	2021	old		no
2937						
2	3	4	2003	new		no
6326	0	2	2012			
3	8	3	2012	new		yes
632	4	า	2011	no. /		\\0.C
4 5414	4	2	2011	new		yes
5	10	5	2018	old		yes
8871	10	J	2010	otu		yes
6	6	9	2009	new		yes
4878	U	9	2003	TICW		ycs
7	3	9	2011	new		yes
3054	J					, 55
8	3	7	2000	old		no
7507						

615 10	9	7	9	1994	old	no
2860 11		2	9	2008	new	no
6629 12 6 2 2001 old no 7473 13 7 7 2000 new no 3573 14 5 8 2016 new no 8150 15 9 2 2021 old no 1559 16 6 9 2021 old no 55075 17 10 5 2020 new no 5278 18 4 6 1992 old yes 4480 19 3 7 2021 old no 231 20 3 1 1997 old yes 8414 21 4 1 1990 new no 3835 22 2 1 1999 new yes 4086 23 1 7 2016 new no 3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 8384 29 7 4 1998 old no 6486 30 3 1990 new yes 8384 29 7 4 1998 old no 66928 31 9 5 2001 old no 9265 32 1 9 5 2001 old no	2860					
12 6 2 2001 old no 7473 13 7 7 2000 new no 3573 14 5 8 2016 new no 8150		8	10	2000	old	no
7473 13		6	2	2001	o] d	no
13         7         7         2000         new         no           3573         14         5         8         2016         new         no           8150         15         9         2         2021         old         no           155         9         2         2021         old         no           1559         16         6         9         2021         old         no           5075         10         5         2020         new         no           5278         1         10         5         2020         new         no           5278         18         4         6         1992         old         yes           4480         19         3         7         2021         old         no           231         3         1         1997         old         yes           8414         21         4         1         1990         new         no           3835         22         2         1         1999         new         yes           24         6         9         1993         old         yes           25 <t< td=""><td></td><td>O .</td><td>2</td><td>2001</td><td>otu</td><td>110</td></t<>		O .	2	2001	otu	110
14       5       8       2016       new       no         8150       9       2       2021       old       no         1559       16       6       9       2021       old       no         5075       17       10       5       2020       new       no         5278       18       4       6       1992       old       yes         4480       19       3       7       2021       old       no         231       20       3       1       1997       old       yes         8414       21       4       1       1990       new       no         3835       22       2       1       1999       new       yes         4086       23       1       7       2016       new       no         3284       24       6       9       1993       old       yes         2485       25       6       10       2016       old       no         3264       27       10       1       2018       old       yes         5024       7       1       1998       old       no	13	7	7	2000	new	no
8150         15       9       2 2021       old       no         1559       6       9 2021       old       no         5075       10       5 2020       new       no         5278       18       4       6 1992       old       yes         4480       19       3       7 2021       old       no         231       20       3       1 1997       old       yes         8414       21       4       1 1990       new       no         3835       22       2       1 1999       new       yes         4086       23       1       7 2016       new       no         3284       24       6       9 1993       old       yes         2485       25       6       10       2016       old       no         8366       26       10       2016       old       yes         5024       27       10       1 2018       old       yes         5024       27       10       1 2018       old       no         28       9       8 2021       new       yes         8384       9		_				
15 9 2 2021 old no 1559 16 6 9 2021 old no 5075 17 10 5 2020 new no 5278 18 4 6 1992 old yes 4480 19 3 7 2021 old no 231 20 3 1 1997 old yes 8414 21 4 1 1990 new no 3835 22 2 1 1999 new yes 4086 23 1 7 2016 new no 3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 3281 29 7 4 1998 old no 6486 30 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		5	8	2016	new	no
1559 16 6 9 2021 old no 5075 17 10 5 2020 new no 5278 18 4 6 1992 old yes 4480 19 3 7 2021 old no 231 20 3 1 1997 old yes 8414 21 4 1 1990 new no 3835 22 2 1 1999 new yes 4086 23 1 7 2016 new no 3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 1995 new yes 2281 28 9 8 2021 new yes 3281 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		0	2	2021	014	no
16       6       9       2021       old       no         5075       10       5       2020       new       no         5278       18       4       6       1992       old       yes         4480       19       3       7       2021       old       no         231       20       3       1       1997       old       yes         8414       21       4       1       1990       new       no         3835       22       2       1       1999       new       yes         4086       23       1       7       2016       new       no         3284       24       6       9       1993       old       yes         2485       6       10       2016       old       no         8366       26       10       2016       old       no         8366       26       10       1       1995       new       yes         5024       27       10       1       2018       old       yes         28       9       8       2021       new       yes         38384       9 <td></td> <td>9</td> <td>2</td> <td>2021</td> <td>otu</td> <td>110</td>		9	2	2021	otu	110
5075         17         10         5         2020         new         no           5278         18         4         6         1992         old         yes           4480         199         old         no         old         no           231         20         3         1         1997         old         yes           8414         21         4         1         1990         new         no           3835         22         2         1         1999         new         yes           4086         23         1         7         2016         new         no           3284         24         6         9         1993         old         yes           2485         25         6         10         2016         old         no           8366         26         10         1         1995         new         yes           5024         27         10         1         2018         old         yes           8384         29         7         4         1998         old         no           29         7         4         1998         old		6	9	2021	old	no
5278         18       4       6       1992       old       yes         4480       3       7       2021       old       no         29       3       1       1997       old       yes         8414       21       4       1       1990       new       no         3835       22       2       1       1999       new       yes         4086       22       2       1       1999       new       yes         23       1       7       2016       new       no         3284       24       6       9       1993       old       yes         24       6       9       1993       old       no         2485       25       6       10       2016       old       no         25       6       10       2016       old       yes         5024       27       10       1       2018       old       yes         3281       28       9       8       2021       new       yes         8384       29       7       4       1998       old       no         29		•	_		0 00	
18       4       6       1992       old       yes         4480       3       7       2021       old       no         231       20       3       1       1997       old       yes         8414       21       4       1       1990       new       no         3835       22       2       1       1999       new       yes         4086       22       1       1999       new       no         3284       24       6       9       1993       old       yes         2445       6       9       1993       old       no         8366       25       6       10       2016       old       no         8366       26       10       1       1995       new       yes         5024       7       10       1       2018       old       yes         3281       29       7       4       1998       old       no         6486       30       3       3       1990       new       yes         6928       31       9       5       2001       old       no         9265 <td></td> <td>10</td> <td>5</td> <td>2020</td> <td>new</td> <td>no</td>		10	5	2020	new	no
4480 19						
19 3 7 2021 old no 231 20 3 1 1997 old yes 8414 21 4 1 1990 new no 3835 22 2 2 1 1999 new yes 4086 23 1 7 2016 new no 3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 3281 28 9 8 2021 new yes 3384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 2668		4	6	1992	old	yes
231 20		_	_			
20 3 1 1997 old yes 8414 21 4 1 1990 new no 3835 22 2 1 1999 new yes 4086 23 1 7 2016 new no 3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 3281 28 9 8 2021 new yes 3281 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		3	7	2021	old	no
8414 21		2	1	1007	م 1 ما	
21       4       1 1990       new       no         3835       22       2       1 1999       new       yes         4086       23       1       7 2016       new       no         3284       24       6       9 1993       old       yes         2485       25       6       10 2016       old       no         8366       26       10       1 1995       new       yes         5024       27       10       1 2018       old       yes         3281       28       9       8 2021       new       yes         8384       29       7       4 1998       old       no         6486       30       3 1990       new       yes         6928       31       9       5 2001       old       no         2668       7       5 1993       old       no		3	7	1997	ota	yes
3835 22		4	1	1990	new	no
22       2       1 1999       new       yes         4086       23       1       7 2016       new       no         3284       24       6       9 1993       old       yes         2485       25       6       10 2016       old       no         8366       26       10       1 1995       new       yes         5024       27       10       1 2018       old       yes         3281       28       9       8 2021       new       yes         8384       29       7       4 1998       old       no         6486       30       3 1990       new       yes         6928       31       9       5 2001       old       no         32       7       5 1993       old       no		<del>-</del> T	_	1330	TICW	110
4086 23		2	1	1999	new	ves
3284 24 6 9 1993 old yes 2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 8384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no	4086					•
24       6       9       1993       old       yes         2485       25       6       10       2016       old       no         8366       26       10       1       1995       new       yes         5024       27       10       1       2018       old       yes         3281       28       9       8       2021       new       yes         8384       29       7       4       1998       old       no         6486       30       3       3       1990       new       yes         6928       31       9       5       2001       old       no         9265       32       7       5       1993       old       no         2668		1	7	2016	new	no
2485 25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 8384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		_	_			
25 6 10 2016 old no 8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 8384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		6	9	1993	old	yes
8366 26 10 1 1995 new yes 5024 27 10 1 2018 old yes 3281 28 9 8 2021 new yes 8384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no		6	10	2016	o] d	no
26       10       1 1995       new       yes         5024       10       1 2018       old       yes         3281       9       8 2021       new       yes         8384       29       7       4 1998       old       no         6486       30       3 1990       new       yes         6928       31       9       5 2001       old       no         9265       32       7       5 1993       old       no         2668		U	10	2010	otu	110
5024 27		10	1	1995	new	ves
3281 28 9 8 2021 new yes 8384 29 7 4 1998 old no 6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no			_			,
28       9       8       2021       new       yes         8384       29       7       4       1998       old       no         6486       30       3       1990       new       yes         6928       31       9       5       2001       old       no         9265       32       7       5       1993       old       no         2668	27	10	1	2018	old	yes
8384 29						
29       7       4       1998       old       no         6486       30       3       1990       new       yes         6928       31       9       5       2001       old       no         9265       32       7       5       1993       old       no         2668       7       2668       1993       old       no		9	8	2021	new	yes
6486 30 3 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no 2668		-	4	1000	7.1	
30 3 1990 new yes 6928 31 9 5 2001 old no 9265 32 7 5 1993 old no 2668		1	4	1998	ola	no
6928 31 9 5 2001 old no 9265 32 7 5 1993 old no 2668		2	3	1000	now	VAC
31 9 5 2001 old no 9265 32 7 5 1993 old no 2668		J	5	1990	IICW	yes
9265 32 7 5 1993 old no 2668		9	5	2001	old	no
32 7 5 1993 old no 2668					-	
	32	7	5	1993	old	no
33 10 2 2012 new yes						
	33	10	2	2012	new	yes

2080 34		8	4	2015	no	V05
7126		0	4	2013	new	yes
35		3	8	2020	old	no
3632						
36		6	2	2007	old	yes
788		2	2	2014		
37 9080		2	3	2014	new	no
38		8	3	1991	old	no
1492		U	3	1331	oca	110
39		4	7	2006	old	no
8654						
40		3	10	1995	new	yes
3477		_	0	1001	-1.1	
41 3218		7	9	1991	old	no
42		8	8	2013	new	no
5486	1	U	0	2015	TICW	110
43		8	8	2002	old	no
4018	}					
44		9	8	2001	new	yes
7369		2	2	2011		
45 6393		3	1	2011	new	no
46		5	8	2002	old	yes
4034		3	0	2002	oca	yes
47		4	3	1993	new	no
292						
48		4	8	2016	old	no
9823		5	1	1000	o 1 d	20
49 4500		5	T	1998	old	no
4300						
	attic	garage	hasstorageroom	hasgu	estroom category	
0	9005	956	no		7 Luxury	
1	8852	135	yes		9 Middle	
1 2 3	4748	654	no		10 Luxury	
3	5792	807	yes		5 Middle	
4 5	1172 7117	716 240	yes		9 Luxury 7 Luxury	
6	281	384	no yes		7 Luxury 5 Luxury	
6 7	129	726	no		9 Middle	
8	9056	892	yes		1 Luxury	
9	1221	328	no		10 Luxury	
10	3129	982	no		1 Luxury	
11	435	512	no		7 Luxury	
12	796	237	yes		3 Middle	
13	9556	918	yes		8 Luxury	

14       6037       930       no       7       Luxury         15       5111       957       yes       2       Luxury         16       3104       864       no       4       Middle         17       1059       313       yes       6       Middle         18       6919       680       yes       1       Luxury         19       1939       223       no       8       Middle         20       6270       939       yes       8       Middle         21       2403       559       no       6       Middle         21       2403       559       no       6       Middle         22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28							
16       3104       864       no       4       Middle         17       1059       313       yes       6       Middle         18       6919       680       yes       1       Luxury         19       1939       223       no       8       Middle         20       6270       939       yes       8       Middle         21       2403       559       no       6       Middle         22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31		6037	930	no		Luxury	
17       1059       313       yes       6       Middle         18       6919       680       yes       1       Luxury         19       1939       223       no       8       Middle         20       6270       939       yes       8       Middle         21       2403       559       no       6       Middle         22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32	15	5111	957	yes	2	Luxury	
18       6919       680       yes       1       Luxury         19       1939       223       no       8       Middle         20       6270       939       yes       8       Middle         21       2403       559       no       6       Middle         22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32	16	3104	864	no	4	Middle	
18       6919       680       yes       1       Luxury         19       1939       223       no       8       Middle         20       6270       939       yes       8       Middle         21       2403       559       no       6       Middle         22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         26       8103       388       yes       4       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       6       Middle         31	17	1059	313	yes	6	Middle	
19 1939 223	18	6919	680		1	Luxury	
20 6270 939 yes 8 Middle 21 2403 559 no 6 Middle 22 5991 494 yes 8 Luxury 23 9879 641 no 2 Luxury 24 108 864 no 7 Luxury 25 4799 979 yes 7 Middle 26 8103 388 yes 4 Middle 27 5020 968 no 8 Luxury 28 7226 226 yes 4 Middle 29 3310 366 no 0 Luxury 30 7808 774 yes 5 Middle 31 8974 755 yes 6 Middle 32 4669 766 no 8 Luxury 33 9575 753 no 7 Middle 34 5012 974 yes 0 Luxury 35 5960 723 yes 3 Luxury 36 4788 132 yes 3 Luxury 37 9356 740 yes 9 Luxury 38 5697 625 no 6 Middle 39 9588 290 yes 8 Middle 40 5530 342 no 2 Luxury 41 9119 849 no 4 Middle 42 3641 766 no 3 Middle 43 4871 836 yes 2 Middle 44 6739 686 yes 6 Middle 45 9082 734 no 0 Luxury 46 2877 787 yes 6 Luxury 47 744 675 no 4 Middle 48 7174 728 yes 0 Luxury	19	1939	223		8		
21			939			Middle	
22       5991       494       yes       8       Luxury         23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       7       Middle         34	21	2403	559		6	Middle	
23       9879       641       no       2       Luxury         24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       9       Luxury         38       5697       625       no       6       Middle         40	22	5991					
24       108       864       no       7       Luxury         25       4799       979       yes       7       Middle         26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         40	23		641			_	
25							
26       8103       388       yes       4       Middle         27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42							
27       5020       968       no       8       Luxury         28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       9       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42							
28       7226       226       yes       4       Middle         29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43							
29       3310       366       no       0       Luxury         30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44							
30       7808       774       yes       5       Middle         31       8974       755       yes       6       Middle         32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45							
31 8974 755 yes 6 Middle 32 4669 766 no 8 Luxury 33 9575 753 no 7 Middle 34 5012 974 yes 0 Luxury 35 5960 723 yes 3 Luxury 36 4788 132 yes 8 Luxury 37 9356 740 yes 9 Luxury 38 5697 625 no 6 Middle 39 9588 290 yes 8 Middle 40 5530 342 no 2 Luxury 41 9119 849 no 4 Middle 42 3641 766 no 3 Middle 43 4871 836 yes 2 Middle 44 6739 686 yes 6 Middle 45 9082 734 no 0 Luxury 46 2877 787 yes 6 Luxury 47 744 675 no 4 Middle 48 7174 728 yes 0 Luxury							
32       4669       766       no       8       Luxury         33       9575       753       no       7       Middle         34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47				——————————————————————————————————————			
33       9575       753       no       7 Middle         34       5012       974       yes       0 Luxury         35       5960       723       yes       3 Luxury         36       4788       132       yes       8 Luxury         37       9356       740       yes       9 Luxury         38       5697       625       no       6 Middle         39       9588       290       yes       8 Middle         40       5530       342       no       2 Luxury         41       9119       849       no       4 Middle         42       3641       766       no       3 Middle         43       4871       836       yes       2 Middle         44       6739       686       yes       6 Middle         45       9082       734       no       0 Luxury         46       2877       787       yes       6 Luxury         47       744       675       no       4 Middle         48       7174       728       yes       0 Luxury							
34       5012       974       yes       0       Luxury         35       5960       723       yes       3       Luxury         36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury							
35 5960 723 yes 3 Luxury 36 4788 132 yes 8 Luxury 37 9356 740 yes 9 Luxury 38 5697 625 no 6 Middle 39 9588 290 yes 8 Middle 40 5530 342 no 2 Luxury 41 9119 849 no 4 Middle 42 3641 766 no 3 Middle 43 4871 836 yes 2 Middle 44 6739 686 yes 6 Middle 45 9082 734 no 0 Luxury 46 2877 787 yes 6 Luxury 47 744 675 no 4 Middle 48 7174 728 yes 0 Luxury							
36       4788       132       yes       8       Luxury         37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury						_	
37       9356       740       yes       9       Luxury         38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury					8	_	
38       5697       625       no       6       Middle         39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury				——————————————————————————————————————			
39       9588       290       yes       8       Middle         40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury						_	
40       5530       342       no       2       Luxury         41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury							
41       9119       849       no       4       Middle         42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury				<u>-</u>			
42       3641       766       no       3       Middle         43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury							
43       4871       836       yes       2       Middle         44       6739       686       yes       6       Middle         45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury							
44       6739       686       yes       6 Middle         45       9082       734       no       0 Luxury         46       2877       787       yes       6 Luxury         47       744       675       no       4 Middle         48       7174       728       yes       0 Luxury					2		
45       9082       734       no       0       Luxury         46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury				_			
46       2877       787       yes       6       Luxury         47       744       675       no       4       Middle         48       7174       728       yes       0       Luxury							
47 744 675 no 4 Middle 48 7174 728 yes 0 Luxury							
48 7174 728 yes 0 Luxury				=		_	
, and the second se							
15 10// 100 IIO 0 Luxui y							
	7.5	1077	<del>1</del> 00	110	J	Luxury	

## df\_kategori2.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 17 columns):

	(	,	
#	Column	Non-Null Count	Dtype
0	squaremeters	10000 non-null	int64
1	numberofrooms	10000 non-null	int64
2	hasyard	10000 non-null	object
3	haspool	10000 non-null	object
4	floors	10000 non-null	int64
5	citycode	10000 non-null	int64

6 7 8	citypartrange numprevowners made	10000	non-null non-null non-null	int64 int64 int64		
9	isnewbuilt	10000	non-null	object		
10	hasstormprotector	10000	non-null	object		
11	basement	10000	non-null	int64		
12	attic	10000	non-null	int64		
13	garage	10000	non-null	int64		
14	hasstorageroom	10000	non-null	object		
15	hasguestroom	10000	non-null	int64		
16	category		non-null	object		
<pre>dtypes: int64(11), object(6)</pre>						

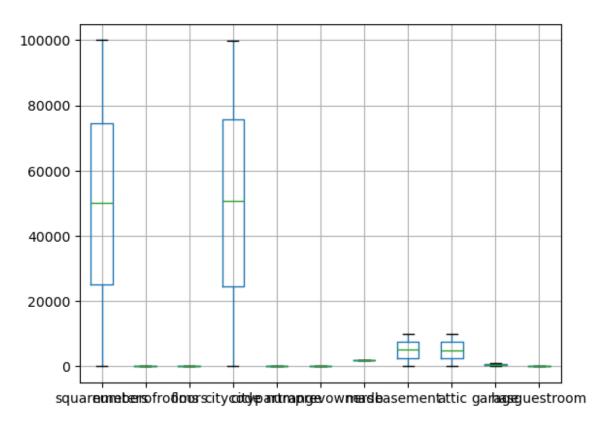
memory usage: 1.3+ MB

## #cek deskripsi data df\_kategori2.describe()

_				
•	uaremeters	numberofrooms	floors	citycode
citypartr		10000 000000	10000 000000	10000 000000
count 1 10000.000	0000.00000	10000.000000	10000.000000	10000.000000
	9870.13120	50.358400	50.276300	50225.486100
5.510100				
	8774.37535	28.816696	28.889171	29006.675799
2.872024	00 00000	1 000000	1 000000	2 000000
min 1.000000	89.00000	1.000000	1.000000	3.000000
	5098.50000	25.000000	25.000000	24693.750000
3.000000				
	0105.50000	50.000000	50.000000	50693.000000
5.000000 75% 7	4609.75000	75.000000	76.000000	75683.250000
8.000000	4009.75000	73.000000	70.00000	73003.230000
	9999.00000	100.000000	100.000000	99953.000000
10.000000				
nu	mprovovnorc	made	basement	attic
garage \	mprevowners	illaue	Dasement	attic
J ,	0000.000000	10000.00000	10000.000000	10000.00000
10000.000				
mean	5.521700	2005.48850	5033.103900	5028.01060
553.12120 std	2.856667	9.30809	2876.729545	2894.33221
262.05017		9.50009	2070.729545	2094.33221
min	1.000000	1990.00000	0.000000	1.00000
100.00000				
25%	3.000000	1997.00000	2559.750000	2512.00000
327.75000 50%	5.000000	2005.50000	5092.500000	5045.00000
554.00000		2003130000	3032130000	30 13 100000

```
75%
             8.000000
                         2014.00000
                                        7511.250000
                                                       7540.50000
777.25000
max
            10.000000
                         2021.00000 10000.000000 10000.00000
1000.00000
        hasguestroom
        10000.00000
count
mean
             4.99460
             3.17641
std
             0.00000
min
25%
             2.00000
50%
             5.00000
75%
             8.00000
            10.00000
max
df_kategori2['category'].value_counts()
category
Basic
           4344
Luxury
           3065
Middle
           2591
Name: count, dtype: int64
#gunakan fungsi isnull, empty, dan isna untuk mengecek data kosong
print("data null \n", df_kategori2.isnull().sum())
print("\ndata kosong \n", df_kategori2.empty)
print("\ndata nan \n", df_kategori2.isna().sum())
data null
                        0
 squaremeters
numberofrooms
                       0
hasyard
                       0
                       0
haspool
                       0
floors
                       0
citycode
                       0
citypartrange
                       0
numprevowners
made
                       0
                       0
isnewbuilt
hasstormprotector
                       0
                       0
basement
attic
                       0
                       0
garage
                       0
hasstorageroom
                       0
hasguestroom
category
dtype: int64
data kosong
 False
```

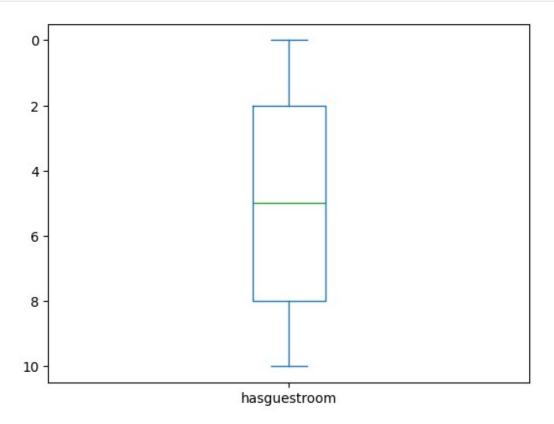
```
data nan
squaremeters
                      0
numberofrooms
                     0
                     0
hasyard
                     0
haspool
                     0
floors
                     0
citycode
                     0
citypartrange
                     0
numprevowners
                     0
made
isnewbuilt
                     0
hasstormprotector
                     0
                     0
basement
                     0
attic
garage
                     0
hasstorageroom
                     0
                     0
hasguestroom
category
                     0
dtype: int64
#cek data outlier
import matplotlib.pyplot as plt
df_kategori2.boxplot()
plt.show()
```



```
#menggunakan fungsi remove outliner untuk menghilangkan outlier
from pandas.api.types import is numeric dtype
def remove outlier(df in):
    for col name in list(df in.columns):
        if is_numeric_dtype(df_in[col_name]):
            q1 = df in[col name].quantile(0.25)
            q3 = df in[col name].quantile(0.75)
            iqr = q3-q1
            batas atas = q3 + (1.5*iqr)
            batas bawah = q1 - (1.5*iqr)
            df out = df in.loc[(df in[col name] >= batas bawah) &
(df in[col name] <= batas atas)]</pre>
    return df out
df kategori clean = remove outlier(df kategori2)
print("jumlah baris DataFrame sebelum dibuang outlier",
df kategori2.shape[0])
print("jumlah baris DataFrame sebelum dibuang outlier",
df kategori clean.shape[0])
df kategori clean.hasguestroom.plot(kind='box', vert=True)
#untuk membalik sumbu v
```

```
plt.gca().invert_yaxis()
plt.show()

jumlah baris DataFrame sebelum dibuang outlier 10000
jumlah baris DataFrame sebelum dibuang outlier 10000
```



print(df\_kategori\_clean.dtypes) squaremeters int64 numberofrooms int64 hasyard object haspool object floors int64 citycode int64 citypartrange int64 numprevowners int64 made int64 isnewbuilt object hasstormprotector object basement int64 attic int64 garage int64 hasstorageroom object hasguestroom int64

```
category object
dtype: object
```

Drop data yang kosong (Missing value)

```
#drop data yang memiliki missing value
print("Sebelum drop missing value",df_kategori2.shape)
df_kategori2 = df_kategori2.dropna(how="any",inplace=False)
print("sesudah drop missing value", df_kategori2.shape)

Sebelum drop missing value (10000, 17)
sesudah drop missing value (10000, 17)
```

pengecekan data duplikat

```
#pengecekan data duplikat
print("Sebelum Penegcekan data duplikat, ", df_kategori2.shape)
df kategori3=df kategori2.drop duplicates(keep='last')
print("Setelah Pengecekan data duplikat, ", df_kategori3.shape)
Sebelum Penegcekan data duplikat,
                                   (10000.17)
Setelah Pengecekan data duplikat, (10000, 17)
from sklearn.model selection import train test split
x=df kategori3.drop(columns=['category'],axis=1)
y= df kategori3['category']
x_train, x_test, y_train, y_test =
train test split(x,y,test size=0.25, random state=78)
print(x train.shape)
print(x test.shape)
(7500, 16)
(2500, 16)
```

data encoding

```
#buat varibel baru untk menampung hadil transformasi kolom
x train enc=transform. fit transform(x train)
#khusus untuk train set gunakan fungsi fit_transform, untuk test set
qunakan transform saja
x test enc=transform. fit transform(x test)
#jika ingin melihat hasil dari transformasi, muat dalam dataframe
df train enc=pd.DataFrame(x train enc,
columns=transform.get_feature_names_out())
df test enc=pd.DataFrame(x test enc,
columns=transform.get feature names out())
df train enc.head(10)
df test enc.head(10)
   onehotencoder hasyard no onehotencoder hasyard yes \
0
                          1.0
                                                        0.0
1
                          0.0
                                                        1.0
2
                          1.0
                                                        0.0
3
                          1.0
                                                        0.0
4
                          0.0
                                                        1.0
5
                          0.0
                                                        1.0
6
                          0.0
                                                        1.0
7
                          0.0
                                                        1.0
8
                          0.0
                                                        1.0
9
                          1.0
                                                        0.0
   onehotencoder haspool no
                               onehotencoder haspool yes \
0
                          0.0
                                                        1.0
1
                          0.0
                                                        1.0
2
                          1.0
                                                        0.0
3
                          0.0
                                                        1.0
4
                          1.0
                                                        0.0
5
                          0.0
                                                        1.0
6
                          0.0
                                                        1.0
7
                          0.0
                                                        1.0
8
                          1.0
                                                        0.0
9
                                                        0.0
                          1.0
                                   onehotencoder__isnewbuilt_old \
   onehotencoder__isnewbuilt_new
0
                              0.0
                                                               1.0
1
                              0.0
                                                               1.0
2
                              0.0
                                                               1.0
3
                              1.0
                                                               0.0
4
                              1.0
                                                               0.0
5
                              0.0
                                                               1.0
6
                              0.0
                                                               1.0
7
                              0.0
                                                               1.0
8
                              1.0
                                                               0.0
9
                              1.0
                                                               0.0
```

```
onehotencoder__hasstormprotector_no
onehotencoder_hasstormprotector_yes \
                                       1.0
0.0
                                       0.0
1
1.0
                                       1.0
2
0.0
3
                                       1.0
0.0
                                       1.0
4
0.0
5
                                       0.0
1.0
                                       0.0
1.0
7
                                       0.0
1.0
                                       0.0
8
1.0
9
                                       1.0
0.0
   onehotencoder__hasstorageroom_no onehotencoder__hasstorageroom_yes
. . .
                                                                          0.0
0
                                   1.0
. . .
1
                                   0.0
                                                                          1.0
. . .
                                   0.0
                                                                          1.0
2
                                   0.0
3
                                                                          1.0
. . .
                                   1.0
                                                                          0.0
4
. . .
                                   0.0
                                                                          1.0
5
. . .
                                   0.0
                                                                          1.0
6
. . .
7
                                   0.0
                                                                          1.0
. . .
                                   0.0
8
                                                                          1.0
                                   0.0
                                                                          1.0
9
. . .
                                remainder floors
   remainder numberofrooms
                                                     remainder citycode \
                         73.0
                                                                  42855.0
0
                                               13.0
1
                         95.0
                                               3.0
                                                                  75381.0
```

2 3 4 5 6 7 8 9	39.0 47.0 64.0 91.0 48.0 93.0 74.0 72.0		8.0 63.0 83.0 67.0 70.0 35.0 14.0 76.0	91674.0 58471.0 30779.0 65183.0 21012.0 12062.0 76662.0 87732.0
	remaindercitypartrange	remaindern	umprevowners	remaindermade
0	9.0		6.0	2015.0
1	5.0		6.0	2003.0
2	2.0		2.0	2009.0
3	10.0		1.0	1990.0
4	6.0		4.0	1992.0
5	1.0		10.0	2019.0
6	5.0		3.0	2007.0
7	10.0		9.0	1998.0
8	9.0		6.0	2004.0
9	2.0		8.0	2017.0
0 1 2 3 4 5 6 7 8 9	remainderbasement rema 2560.0 6810.0 1477.0 1730.0 594.0 6824.0 5861.0 2064.0 9501.0 6414.0	inderattic 6823.0 1391.0 3153.0 7967.0 8310.0 7141.0 2750.0 2720.0 9579.0 6111.0		arage \ 239.0 556.0 952.0 722.0 898.0 956.0 652.0 315.0 768.0 613.0
0 1 2 3 4	remainderhasguestroom 10.0 3.0 2.0 0.0 5.0			

```
5 6.0
6 10.0
7 5.0
8 1.0
9 7.0
[10 rows x 21 columns]
```

## Pipeline

```
#import Library yang dibutuhkan untuk pipeline, GSCV, dan metrik
evaluasi
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.feature_selection import SelectPercentile, SelectKBest
from sklearn.svm import SVC
from sklearn.model selection import GridSearchCV, StratifiedKFold
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification report, confusion matrix,
ConfusionMatrixDisplay
#buat rancangan pipeline mulai dari data scaling hingga classifier
pipe svm = Pipeline(steps=[
    ('scale', MinMaxScaler()),
    ('feat select', SelectKBest()),
    ('clf', SVC(class weight='balanced'))
])
#buat parameter grid untuk step feature selection dan classifier
params_grid_svm = [
    'scale': [MinMaxScaler()],
    'feat select k':np.arange(2,6),
    'clf kernel': ['poly', 'rbf'],
    'clf C':[0.1,1],
    'clf gamma':[0.1, 1]
    },
    'scale': [MinMaxScaler()],
    'feat select': [SelectPercentile()],
    'feat_select__percentile':np.arange(20,50),
    'clf kernel': ['poly','rbf'],
    'clf C':[ 0.1, 1],
    'clf gamma': [0.1, 1]
    },
    'scale': [StandardScaler() ],
    'feat select k':np.arange(2,6),
    'clf__kernel': ['poly','rbf'],
```

```
'clf C':[0.1, 1],
    'clf__gamma':[0.1, 1]
    },
    'scale': [StandardScaler() ],
    'feat_select':[SelectPercentile()],
    'feat select percentile': np.arange(20,50),
    'clf_kernel':['poly','rbf'],
    'clf C':[0.1, 1],
    'clf gamma':[0.1, 1]
    }
1
#muat rancangan pipeline ke dalam objek pipeline
estimator svm = Pipeline(pipe_svm)
#muat pipeline dan parameter gri ke dalam objek GSCV dengan Stratified
5-fold CV
SKF = StratifiedKFold(n splits=5, shuffle=True, random state=78)
GSCV SVM = GridSearchCV(pipe svm, params grid svm, cv=SKF)
#jalankan objek GSC untuk melatih model dengan train set menggunakan
fungsi fit
GSCV SVM.fit(x train enc, y train)
print("GSCV training finished")
#64min
GSCV training finished
#tampilkan skor cross-validation
print("CV Score : {}".format(GSCV SVM.best score ))
#tampilkan skor model terbaik GSCV pada test set
print("Test Score:
{}".format(GSCV SVM.best estimator .score(x test enc, y test)))
#tampilkan best model dan best features
print("Best model:", GSCV SVM.best estimator )
mask =
GSCV_SVM.best_estimator_.named_steps['feat_select' ].get_support()
print("Best features:", df_train_enc.columns[mask])
#buat prediksi dari test set
SVM_pred = GSCV_SVM.predict(x_test_enc)
import matplotlib.pyplot as plt
#buat confusion matrix
cm = confusion_matrix(y_test, SVM_pred, labels=GSCV SVM.classes )
#buat confusion matrix display
disp = ConfusionMatrixDisplay(confusion matrix=cm,
display labels=GSCV SVM.classes )
```

```
disp.plot()
plt.title("SVM Confusion Matrix")
plt.show()
#tampilkan classification report
print("Classification report SVM:\n", classification_report(y_test,
SVM_pred))
Test Score: 0.9952
Best model: Pipeline(steps=[('scale', StandardScaler()),
               ('feat select', SelectPercentile(percentile=31)),
               ('clf',
                SVC(C=1, class weight='balanced', gamma=1,
kernel='poly'))])
Best features: Index(['onehotencoder hasyard no',
'onehotencoder__hasyard_yes',
       'onehotencoder haspool no', 'onehotencoder haspool yes',
       'onehotencoder isnewbuilt new',
'onehotencoder__isnewbuilt_old',
       'remainder squaremeters'],
     dtype='object')
```



```
Classification report SVM:
                             recall f1-score
               precision
                                                support
       Basic
                   1.00
                             0.99
                                        1.00
                                                  1091
                   1.00
                             1.00
                                        1.00
      Luxury
                                                   750
      Middle
                   0.98
                             1.00
                                        0.99
                                                   659
                                        1.00
                                                  2500
    accuracy
   macro avq
                   0.99
                             1.00
                                        1.00
                                                  2500
weighted avg
                   1.00
                             1.00
                                        1.00
                                                  2500
from sklearn.ensemble import GradientBoostingClassifier
from sklearn. feature selection import SelectFromModel
from sklearn. tree import DecisionTreeClassifier
pipe GBT = Pipeline(steps=[
    ('feat select', SelectKBest()),
    ('clf', GradientBoostingClassifier(random state=78))])##random
state isi dengan 2
params grid GBT = [
                    {
                        'feat select k': np.arange(2,6),
                        'clf max depth': [*np.arange(4,5)],
                        'clf n estimators': [100,150],
                        'clf learning rate': [0.01,0.1,1]
                    },
{
                        'feat select': [SelectPercentile()],
                        'feat select percentile':np.arange(20,50),
                        'clf max depth': [*np.arange(4,5)],
                        'clf__n_estimators': [100,150],
                        'clf learning rate': [0.01,0.1,1]
                    },
{
                        'feat select k': np.arange(2,6),
                        'clf max depth': [*np.arange(4,5)],
                        'clf n estimators': [100,150],
                        'clf learning rate': [0.01,0.1,1]
                    },
                    {
                        'feat select':[SelectPercentile()],
                        'feat select percentile':np.arange(20,50),
                        'clf max depth': [*np.arange(4,5)],
                         'clf__n_estimators': [100,150],
```

```
'clf learning rate':[0.01,0.1,1]
                    }
]
GSCV GBT =
GridSearchCV(pipe GBT,params grid GBT,cv=StratifiedKFold(n splits=5))
GSCV GBT.fit(x train enc,y train)
print("GSCV Finished")
NameError
                                          Traceback (most recent call
last)
Cell In[4], line 5
      2 from sklearn. feature selection import SelectFromModel
      3 from sklearn. tree import DecisionTreeClassifier
----> 5 pipe GBT = Pipeline(steps=[
            ('feat select', SelectKBest()),
      7
            ('clf',
GradientBoostingClassifier(random state=78))])##random state isi
dengan 2
      9 params grid GBT = [
     10
                            {
     11
                                 'feat select k': np.arange(2,6),
   (\ldots)
     41
     42 ]
     44 GSCV GBT =
GridSearchCV(pipe GBT,params grid GBT,cv=StratifiedKFold(n splits=5))
NameError: name 'Pipeline' is not defined
#tampilkan skor cross-validation
print("CV Score: {}".format(GSCV GBT.best score ))
#tampilkan skor model terbaik GSCV pada test set
print("Test Score:
{}".format(GSCV GBT.best estimator .score(x test enc, y test)))
#tampilkan best model dan best features
print("Best model:", GSCV GBT.best estimator )
mask =
GSCV GBT.best estimator .named steps['feat select' ].get support()
print("Best features:", df_train enc.columns[mask])
#buat prediksi dari test set
GBT pred = GSCV GBT.predict(x test enc)
```

```
import matplotlib.pyplot as plt
#buat confusion matrix
cm = confusion_matrix(y_test, GBT_pred, labels=GSCV_GBT.classes_)
#buat confusion matrix display
disp = ConfusionMatrixDisplay(confusion matrix=cm,
display_labels=GSCV_GBT.classes )
disp.plot()
plt.title("GBT Confusion Matrix")
plt.show()
#tampilkan Classification report
print("Classification report GBT: \n", classification report(y test,
GBT pred))
NameError
                                          Traceback (most recent call
last)
Cell In[3], line 2
      1 #tampilkan skor cross-validation
----> 2 print("CV Score: {}".format(GSCV GBT.best score ))
      3 #tampilkan skor model terbaik GSCV pada test set
      4 print("Test Score:
{}".format(GSCV GBT.best estimator .score(x test enc, y test)))
NameError: name 'GSCV GBT' is not defined
import pickle
#simpan model menggunakan library Pickle
with open('BestModel CLF GBT pytorch.pkl','wb') as r:
    pickle.dump((GSCV GBT),r)
#File pickle akan tersimpan di folder yang sama dengan file notebook
print("Model GBT berhasil disimpan")
NameError
                                          Traceback (most recent call
last)
Cell In[2], line 5
      3 #simpan model menggunakan library Pickle
      4 with open('BestModel CLF GBT pytorch.pkl','wb') as r:
            pickle.dump((GSCV GBT),r)
      7 #File pickle akan tersimpan di folder yang sama dengan file
notebook
      8 print("Model GBT berhasil disimpan")
NameError: name 'GSCV GBT' is not defined
```

import pandas as pd
import numpy as np

df\_harga =pd.read\_csv(r'D:\semester 5\Mesin Learning\UTS\Dataset
UTS\_Gasal 2425.csv')
df\_harga.head(20)

					63		
	squaremeters	numberofrooms	nasyard	naspool	floors	citycode	\
0	75523	3	no	yes	63	9373	
1	55712	58	no	yes	19	34457	
2	86929	100	yes	no	11	98155	
3	51522	3	no	no	61	9047	
4	96470	74	yes	no	21	92029	
5	79770	3	no	yes	69	54812	
6	75985	60	yes	no	67	6517	
7	64169	88	no	yes	6	61711	
8	92383	12	no	no	78	71982	
9	95121	46	no	yes	3	9382	
10	76485	47	yes	no	9	90254	
11	87060	27	no	yes	91	51803	
12	66683	19	yes	yes	6	50801	
13	84559	29	no	yes	69	53057	
14	76091	38	yes	no	32	59451	
15	92696	49	yes	no	38	74381	
16	59800	47	no	yes	27	44815	
17	54836	25	no	yes	53	64601	
18	70021	52	yes	no	28	95678	
19	54368	11	yes	yes	20	55761	
	3 1300		, 5	, 63	20	23,01	

	cityp	artrange	numprevowners	made	isnewbuilt	hasstormprotector
base	ement	\				
0		3	8	2005	old	yes
4313	3					
1		6	8	2021	old	no
2937	7					
2		3	4	2003	new	no
6326	5					
3		8	3	2012	new	yes
632						
4		4	2	2011	new	yes
5414	4					
5		10	5	2018	old	yes
8871	l					
6		6	9	2009	new	yes
4878	3					
7		3	9	2011	new	yes
3054	1					
8		3	7	2000	old	no
7507	7					
9		7	9	1994	old	no

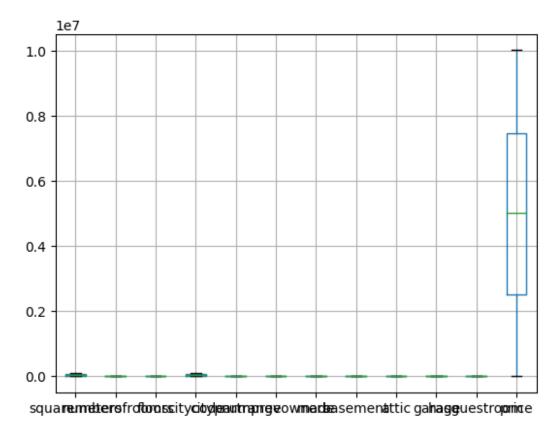
615							
10 286	O	2	9	2008	n	ew	no
11	U	8	10	2000	0	ld	no
662	9	_					
12		6	2	2001	0	ld	no
747	3	_	7	2000			
13 357	3	7	7	2000	n.	ew	no
14	J	5	8	2016	n	ew	no
815	0		_				
15		9	2	2021	0	ld	no
155	9	6	0	2021	_	ا ما	
16 507	5	6	9	2021	0	ld	no
17	J	10	5	2020	n	ew	no
527	8		_				
18		4	6	1992	0	ld	yes
448	0	2	7	2021		1 4	
19 231		3	7	2021	0	ld	no
231							
	attic	garage	hasstorageroom	hasgu	estroom	price	category
0	9005	956	no		7	7559081.5	
1	8852	135	yes		9	5574642.1	
2	4748	654	no		10	8696869.3	
3 4	5792 1172	807 716	yes yes		5 9	5154055.2 9652258.1	
5	7117	240	no		7	7986665.8	
6	281	384	yes		5	7607322.9	
7	129	726	no		9	6420823.1	
8	9056	892	yes		1	9244344.0	
9	1221	328	no		10	9515440.4	•
10	3129	982	no		1	7653300.8	•
11	435	512	no		7	8711426.0	•
12 13	796 9556	237 918	yes		3 8	6677649.1 8460604.0	
13 14	6037	930	yes no		7	7614076.6	
15	5111	957	yes		2	9272740.1	
16	3104	864	no		4	5984462.1	
17	1059	313	yes		6	5492532.0	
18	6919	680	yes		1	7005572.2	
19	1939	223	no		8	5446398.1	Middle
	harga2 harga2.		rga.drop([ˈcateg	jory'],	axis= <mark>1</mark> )		
	squarem	eters r	numberofrooms ha	svard	haspool	floors c	itycode \
0	•	75523	3	no	yes	63	9373
1		55712	58	no	yes	19	34457
					•		

## Column   Non-Null Count Dtype   Columns   C					
basement \ 0	3	51522	3 no	no	61 9047
0 3 8 2005 old yes 4313 1 6 8 2021 old no 2937 2 3 4 2003 new no 6326 3 8 3 2012 new yes 632 4 4 2 2011 new yes 6314  attic garage hasstorageroom hasguestroom price 0 9005 956 no 7 7559081.5 1 8852 135 yes 9 5574642.1 2 4748 654 no 10 8696869.3 3 5792 807 yes 5 5154055.2 4 1172 716 yes 9 9652258.1  df_harga2.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 10000 entries, 0 to 9999 Data columns (total 17 columns): # Column Non-Null Count Dtype</class>			revowners made	isnewbuilt has	stormprotector
1 6 8 2021 old no. 2937 2 3 4 2003 new no. 6326 3 8 3 2012 new ye. 632 4 4 2 2011 new ye. 5414  attic garage hasstorageroom hasguestroom price 0 9005 956 no 7 7559081.5 1 8852 135 yes 9 5574642.1 2 4748 654 no 10 8696869.3 3 5792 807 yes 5 5154055.2 4 1172 716 yes 9 9652258.1  df_harga2.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 10000 entries, 0 to 9999 Data columns (total 17 columns): # Column Non-Null Count Dtype</class>	0	3	8 2005	old	yes
2	1	6	8 2021	old	no
3 8 3 2012 new yes 632 4 4 4 2 2011 new yes 5414  attic garage hasstorageroom hasguestroom price 0 9005 956 no 7 7559081.5 1 8852 135 yes 9 5574642.1 2 4748 654 no 10 8696869.3 3 5792 807 yes 5 5154055.2 4 1172 716 yes 9 9652258.1  df_harga2.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 10000 entries, 0 to 9999 Data columns (total 17 columns): # Column Non-Null Count Dtype 0 squaremeters 10000 non-null int64 1 numberofrooms 10000 non-null int64 2 hasyard 10000 non-null object 3 haspool 10000 non-null object 4 floors 10000 non-null int64 5 citycode 10000 non-null int64 6 citypartrange 10000 non-null int64 7 numprevowners 10000 non-null int64</class>	2	3	4 2003	new	no
4 4 2 2011 new yes 5414  attic garage hasstorageroom hasguestroom price 0 9005 956 no 7 7559081.5 1 8852 135 yes 9 5574642.1 2 4748 654 no 10 8696869.3 3 5792 807 yes 5 5154055.2 4 1172 716 yes 9 9652258.1  df_harga2.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 10000 entries, 0 to 9999 Data columns (total 17 columns): # Column Non-Null Count Dtype</class>	3	8	3 2012	new	yes
attic garage hasstorageroom hasguestroom price 0 9005 956 no 7 7559081.5 1 8852 135 yes 9 5574642.1 2 4748 654 no 10 8696869.3 3 5792 807 yes 5 5154055.2 4 1172 716 yes 9 9652258.1  df_harga2.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 10000 entries, 0 to 9999 Data columns (total 17 columns): # Column Non-Null Count Dtype</class>	4	4	2 2011	new	yes
8 made       10000 non-null int64         9 isnewbuilt       10000 non-null object         10 hasstormprotector       10000 non-null int64         11 basement       10000 non-null int64         12 attic       10000 non-null int64         13 garage       10000 non-null object         14 hasstorageroom       10000 non-null int64         15 hasguestroom       10000 non-null float64         16 price       10000 non-null float64	0 9005 1 8852 2 4748 3 5792 4 1172 df_harga2 <class #="" 'parageindex="" 0="" 1="" 10="" 11="" 12="" 13="" 14="" 15="" 2="" 3="" 4="" 5="" 6="" 7="" 8="" 9="" attia="" basea="" citya="" colum="" data="" floo="" garaa="" hasga<="" haspa="" hassa="" hasya="" isnex="" made="" number="" nump="" squa="" td=""><td>956 135 654 807 716 2.info() candas.core.fra ex: 10000 entri umns (total 17 umn aremeters cerofrooms yard cool cors ycode ypartrange crevowners e ewbuilt stormprotector ement ic age storageroom guestroom</td><td>no yes no yes yes yes  ame.DataFrame'&gt; ies, 0 to 9999 columns): Non-Null Count 10000 non-null 10000 non-null</td><td>7 75590 9 55746 10 86968 5 51540 9 96522 Dtype int64 int64 object object int64</td><td>81.5 42.1 69.3 955.2</td></class>	956 135 654 807 716 2.info() candas.core.fra ex: 10000 entri umns (total 17 umn aremeters cerofrooms yard cool cors ycode ypartrange crevowners e ewbuilt stormprotector ement ic age storageroom guestroom	no yes no yes yes yes  ame.DataFrame'> ies, 0 to 9999 columns): Non-Null Count 10000 non-null	7 75590 9 55746 10 86968 5 51540 9 96522 Dtype int64 int64 object object int64	81.5 42.1 69.3 955.2

df_harga2.	describe()			
squ citypartra	naremeters	numberofrooms	floors	citycode
	0000.00000	10000.000000	10000.000000	10000.000000
	9870.13120	50.358400	50.276300	50225.486100
	3774.37535	28.816696	28.889171	29006.675799
min 1.000000	89.00000	1.000000	1.000000	3.000000
	098.50000	25.000000	25.000000	24693.750000
	105.50000	50.000000	50.000000	50693.000000
	1609.75000	75.000000	76.000000	75683.250000
	999.00000	100.000000	100.000000	99953.000000
	nrovounors	made	basement	attic
garage \	nprevowners	illaue	Dasement	attic
J J - ·	0000.000000	10000.00000	10000.000000	10000.00000
mean 553.12120	5.521700	2005.48850	5033.103900	5028.01060
std 262.05017	2.856667	9.30809	2876.729545	2894.33221
min 100.00000	1.000000	1990.00000	0.000000	1.00000
25% 327.75000	3.000000	1997.00000	2559.750000	2512.00000
50% 554.00000	5.000000	2005.50000	5092.500000	5045.00000
75% 777.25000	8.000000	2014.00000	7511.250000	7540.50000
max 1000.00000	10.000000	2021.00000	10000.000000	10000.00000
1000.00000	,			
	guestroom	price		
count 10	0000.00000 4.99460	1.000000e+04 4.993448e+06		
std	3.17641	2.877424e+06		
min	0.00000	1.031350e+04		
25% 5.0%	2.00000	2.516402e+06		
50% 75%	5.00000 8.00000	5.016180e+06 7.469092e+06		
max	10.00000	1.000677e+07		

```
print(df harga2['price'].value counts())
price
7559081.5
              1
2600292.1
              1
              1
3804577.4
3658559.7
              1
2316639.4
              1
5555606.6
             1
5501007.5
             1
9986201.2
              1
              1
9104801.8
146708.4
              1
Name: count, Length: 10000, dtype: int64
print("data null \n", df_harga2.isnull().sum())
print("data kosong \n", df_harga2.empty)
print("data nan \n", df_harga2.isna().sum())
data null
                       0
 squaremeters
numberofrooms
                      0
                      0
hasyard
                      0
haspool
floors
                      0
                      0
citycode
                      0
citypartrange
numprevowners
                      0
made
                      0
                      0
isnewbuilt
                      0
hasstormprotector
                      0
basement
                      0
attic
garage
                      0
                      0
hasstorageroom
                      0
hasguestroom
                      0
price
dtype: int64
data kosong
False
data nan
                       0
 squaremeters
numberofrooms
                      0
                      0
hasyard
                      0
haspool
floors
                      0
                      0
citycode
                      0
citypartrange
                      0
numprevowners
```

```
made
                      0
isnewbuilt
                      0
hasstormprotector
                      0
                      0
basement
                      0
attic
                      0
garage
hasstorageroom
                      0
hasquestroom
                      0
price
dtype: int64
#cek data outlier
import matplotlib.pyplot as plt
df harga2.boxplot()
plt.show()
```

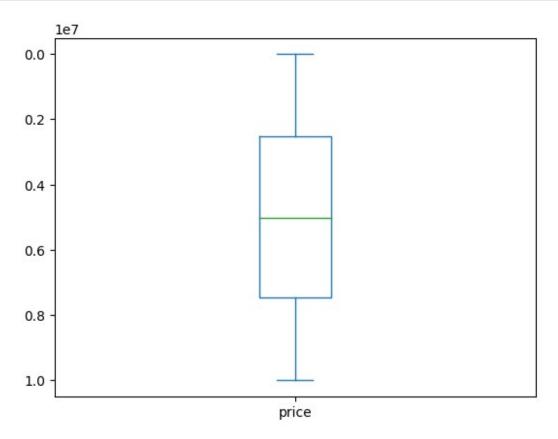


```
print("Sebelum pengecekan data duplikat, ", df_harga.shape)
df_harga3 = df_harga2.drop_duplicates(keep = 'last')
print("Setelah pengecekan data duplikat, ", df_harga2.shape)

Sebelum pengecekan data duplikat, (10000, 18)
Setelah pengecekan data duplikat, (10000, 17)
```

```
import matplotlib.pyplot as plt

df_harga3.price.plot(kind='box')
plt.gca().invert_yaxis()
plt.show()
```



```
from sklearn.model_selection import train_test_split

x_regress = df_harga3.drop(columns=['price'], axis = 1)
y_regress = df_harga3['price']

x_train_price, x_test_price, y_train_price, y_test_price =
train_test_split(x_regress, y_regress, test_size= 0.25, random_state=
78)

print(x_train_price.shape)
print(x_test_price.shape)

(7500, 16)
(2500, 16)

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer

category_column = ['hasyard', 'haspool', 'isnewbuilt',
```

```
'hasstormprotector', 'hasstorageroom']
transform = make column transformer((OneHotEncoder(),
category column), remainder = 'passthrough')
x train price enc = transform.fit transform(x train price)
x test price enc = transform.fit transform(x test price)
df train enc = pd.DataFrame(x train price enc, columns =
transform.get feature names out())
df_test_enc = pd.DataFrame(x_test_price_enc, columns =
transform.get_feature_names_out())
df train enc.head(20)
df test enc.head(20)
                                 onehotencoder__hasyard_yes \
    onehotencoder__hasyard_no
0
                            1.0
                                                          0.0
1
                           0.0
                                                         1.0
2
                            1.0
                                                         0.0
3
                            1.0
                                                         0.0
4
                            0.0
                                                         1.0
5
                           0.0
                                                         1.0
6
                           0.0
                                                         1.0
7
                           0.0
                                                         1.0
8
                           0.0
                                                         1.0
9
                            1.0
                                                         0.0
10
                           0.0
                                                         1.0
11
                                                         1.0
                           0.0
12
                           1.0
                                                         0.0
13
                           0.0
                                                         1.0
14
                            1.0
                                                         0.0
15
                           0.0
                                                         1.0
16
                            1.0
                                                         0.0
17
                           0.0
                                                         1.0
18
                           0.0
                                                         1.0
19
                            1.0
                                                         0.0
    onehotencoder__haspool_no
                                 onehotencoder__haspool_yes \
0
                           0.0
                                                          1.0
1
                            0.0
                                                          1.0
2
                            1.0
                                                         0.0
3
                                                         1.0
                            0.0
4
                            1.0
                                                         0.0
5
                           0.0
                                                         1.0
6
                           0.0
                                                         1.0
7
                           0.0
                                                         1.0
8
                            1.0
                                                         0.0
9
                            1.0
                                                         0.0
10
                           0.0
                                                         1.0
```

```
11
                             1.0
                                                            0.0
12
                             1.0
                                                            0.0
13
                                                            0.0
                             1.0
14
                             1.0
                                                            0.0
15
                             1.0
                                                            0.0
16
                             1.0
                                                            0.0
17
                             0.0
                                                            1.0
18
                             1.0
                                                            0.0
19
                             1.0
                                                            0.0
    onehotencoder__isnewbuilt_new
                                       onehotencoder__isnewbuilt_old \
0
                                  0.0
                                                                    1.0
                                 0.0
1
                                                                    1.0
2
                                 0.0
                                                                    1.0
3
                                 1.0
                                                                    0.0
4
                                 1.0
                                                                    0.0
5
                                 0.0
                                                                    1.0
6
                                 0.0
                                                                    1.0
7
                                 0.0
                                                                    1.0
8
                                 1.0
                                                                    0.0
9
                                  1.0
                                                                    0.0
10
                                 0.0
                                                                    1.0
11
                                 0.0
                                                                    1.0
12
                                 1.0
                                                                    0.0
13
                                 1.0
                                                                    0.0
14
                                 0.0
                                                                    1.0
15
                                 1.0
                                                                    0.0
16
                                 1.0
                                                                    0.0
17
                                 1.0
                                                                    0.0
18
                                 1.0
                                                                    0.0
19
                                 1.0
                                                                    0.0
    onehotencoder__hasstormprotector_no
onehotencoder__hasstormprotector_yes \
                                        1.0
0.0
                                        0.0
1
1.0
2
                                        1.0
0.0
3
                                        1.0
0.0
4
                                        1.0
0.0
                                        0.0
5
1.0
                                        0.0
6
1.0
7
                                        0.0
```

8	1.0		
1.0 99			
9			0.0
0.0 100			1.0
10	9		1.0
0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0			1 0
11			1.0
1.0 12			0 0
12			0.0
1.0 13			0.0
13			0.0
0.0 14			1.0
14			
0.0 15	14		1.0
15	0.0		
16	15		1.0
0.0 17	0.0		
17 0.0 18 0.0 18 0.0 19 1.0 0.0  Onehotencoder_hasstorageroom_no onehotencoder_hasstorageroom_yes \ 0 0.0 1.0 0.0 0	16		1.0
0.0 18	0.0		
18			1.0
0.0 19	0.0		1.0
19			1.0
<pre>0.0</pre>			1 0
onehotencoder_hasstorageroom_no onehotencoder_hasstorageroom_yes \ 0			1.0
onehotencoderhasstorageroom_yes \ 0	0.0		
onehotencoderhasstorageroom_yes \ 0		onehotencoder hass	storageroom no
$egin{array}{cccccccccccccccccccccccccccccccccccc$	oneh	otencoder hasstore	
$egin{array}{cccccccccccccccccccccccccccccccccccc$	^	ocencoder nasscore	ageroom ves \
1.0          2       0.0         1.0          3       0.0         1.0          6       0.0         1.0          7       0.0         1.0          8       0.0         1.0          9       0.0         1.0	0		ageroom_yes \ 1.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0 0.0		ageroom_yes \ 1.0
1.0          3       0.0         1.0          4       1.0         0.0          5       0.0         1.0          6       0.0         1.0          8       0.0         1.0          9       0.0         1.0	0.0 1		1.0
3       0.0         1.0       1.0         0.0          5       0.0         1.0          6       0.0         1.0          7       0.0         1.0          9       0.0         1.0	0.0 1 1.0		0.0
1.0          4       1.0         0.0          5       0.0         1.0          6       0.0         1.0          7       0.0         1.0          8       0.0         1.0          9       0.0         1.0	0.0 1 1.0 2		0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2		1.0 0.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3		1.0 0.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3		1.0 0.0 0.0 0.0
1.0 6 0.0 1.0 7 0.0 1.0 8 0.0 1.0 9 0.0 1.0	0.0 1 1.0 2 1.0 3 1.0 4		1.0 0.0 0.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3 1.0 4 0.0		1.0 0.0 0.0 0.0 1.0
1.0 7	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5		1.0 0.0 0.0 0.0 1.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5		1.0 0.0 0.0 0.0 1.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0		1.0 0.0 0.0 0.0 1.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0		1.0 0.0 0.0 0.0 1.0 0.0 0.0
1.0 9	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0		1.0 0.0 0.0 0.0 1.0 0.0 0.0
$egin{array}{cccccccccccccccccccccccccccccccccccc$	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0 7		1.0 0.0 0.0 0.0 1.0 0.0 0.0
1.0	1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0 7		1.0 0.0 0.0 0.0 1.0 0.0 0.0
	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0 7 1.0 8		1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 6 1.0 7 1.0 8 1.0		1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
	0.0 1 1.0 2 1.0 3 1.0 4 0.0 5 1.0 7 1.0 8 1.0 9		1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0

1.0			
11 0.0		1.0	
12		0.0	
1.0 13	• • •	0.0	
1.0 14		1.0	
0.0			
15 1.0		0.0	
16 1.0		0.0	
17	• • •	1.0	
0.0 18		0.0	
1.0 19		0.0	
1.0		0.0	
	remaindernumberofrooms	remainderfloors	
rema 0	indercitycode \ 73.0	13.0	42855.0
1	95.0	3.0	75381.0
2	39.0	8.0	91674.0
3	47.0	63.0	58471.0
4	64.0	83.0	30779.0
5	91.0	67.0	65183.0
6	48.0	70.0	21012.0
7	93.0	35.0	12062.0
8	74.0	14.0	76662.0
9	72.0	76.0	87732.0
10	18.0	66.0	38920.0
11	37.0	26.0	96016.0
12	76.0	95.0	76985.0
13	26.0	99.0	38185.0
14	69.0	45.0	88591.0

15	9.0	25.0	33740.0
16	29.0	58.0	13202.0
17	91.0	43.0	93072.0
18	68.0	62.0	97608.0
19	48.0	53.0	34588.0
remainder remainderma		remaindernumprevowners	
0	9.0	6.0	
2015.0	5.0	6.0	
2003.0			
2 2009.0	2.0	2.0	
3	10.0	1.0	
1990.0			
4 1992.0	6.0	4.0	
5	1.0	10.0	
2019.0 6	5.0	3.0	
2007.0	5.0	3.0	
7	10.0	9.0	
1998.0 8	9.0	6.0	
2004.0			
9 2017.0	2.0	8.0	
10	7.0	2.0	
2007.0 11	3.0	6.0	
2000.0	3.0	0.0	
12	4.0	7.0	
2009.0 13	7.0	7.0	
1994.0			
14 1995.0	1.0	8.0	
15	2.0	1.0	
2009.0	6.0	Γ.0	
16 1993.0	6.0	5.0	
17	1.0	7.0	
2012.0			

19	18	1.0	9.0	
0		9.0	6.0	
remainder_hasguestroom  10.0  10.0  13.0  2.0  3.0  0.0  4.5.0  5.0  6.0  6.0  7.5.0  8.1.0  9.7.0  10.0  10.0  11.0  10.0  12.7.0  13.3.0  14.3.0  15.0.0  16.5.0  17.8.0  18.0  10.0	0 2560.0 1 6810.0 2 1477.0 3 1730.0 4 594.0 5 6824.0 6 5861.0 7 2064.0 8 9501.0 9 6414.0 10 4949.0 11 1261.0 12 3586.0 13 7053.0 14 3429.0 15 7818.0 16 6740.0 17 1528.0 18 131.0	6823.0 1391.0 3153.0 7967.0 8310.0 7141.0 2750.0 2720.0 9579.0 6111.0 5811.0 6205.0 6017.0 1109.0 8363.0 1494.0 5829.0 4961.0 336.0	239.0 556.0 952.0 722.0 898.0 956.0 652.0 315.0 768.0 613.0 185.0 850.0 472.0 827.0 876.0 567.0 484.0 893.0 656.0	
[20 rows x 21 columns]	0	0.0 3.0 2.0 0.0 5.0 6.0 0.0 5.0 1.0 7.0 6.0 0.0 7.0 3.0 3.0 0.0 5.0		

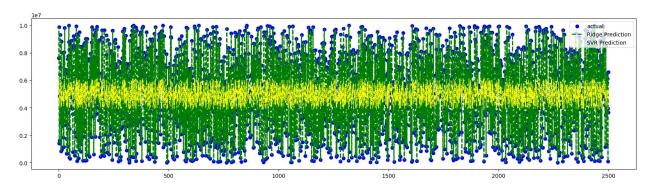
```
from sklearn.linear model import Ridge
from sklearn.model selection import GridSearchCV
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.feature selection import SelectKBest, f regression
from sklearn.metrics import mean absolute error, mean squared error
pipe Ridge = Pipeline(steps=[
    ('scale', StandardScaler()),
    ('feature_selection', SelectKBest(score_func=f_regression)),
    ('reg', Ridge())
])
param grid Ridge = {
    'reg alpha': [0.01, 0.1, 1, 10, 100],
    'feature selection k': np.arange(1, 20)
}
GSCV RR = GridSearchCV(pipe Ridge, param grid Ridge, cv=5,
scoring='neg mean squared error', error score='raise')
GSCV RR.fit(x train price enc, y train price)
print("Best model: {}".format(GSCV RR.best estimator ))
print("Ridge best parameters: {}".format(GSCV RR.best params ))
print("Koefisien/bobot:
{}".format(GSCV RR.best estimator .named steps['reg'].coef ))
print("Intercept/bias:
{}".format(GSCV RR.best estimator .named steps['reg'].intercept ))
Ridge predict = GSCV RR.predict(x test price enc)
mse_Ridge = mean_squared_error(y_test_price, Ridge_predict)
mae_Ridge = mean_absolute_error(y_test_price, Ridge_predict)
print("Ridge Mean Squared Error (MSE): {}".format(mse Ridge))
print("Ridge Mean Absolute Error (MAE): {}".format(mae Ridge))
print("Ridge Root Mean Squared Error: {}".format(np.sqrt(mse Ridge)))
Best model: Pipeline(steps=[('scale', StandardScaler()),
                ('feature selection',
                 SelectKBest(k=19,
                             score_func=<function f_regression at</pre>
0x000001B4370873A0>)),
                ('reg', Ridge(alpha=0.01))])
Ridge best parameters: {'feature selection k': 19, 'reg alpha':
0.01
Koefisien/bobot: [-7.56045182e+02 7.56045183e+02 -7.41944533e+02
7.41944533e+02
  4.65683060e+01 -4.65683083e+01 -4.05487999e+01 4.05488010e+01
 -5.34405351e+00 5.34405341e+00 2.88881677e+06 7.40651732e+00
```

```
1.55973632e+03 1.41411464e+02 -8.98869834e+00 -1.35881746e+01
 -1.85257372e+01 3.70581728e+01 -1.76438214e+011
Intercept/bias: 5003139.741906667
Ridge Mean Squared Error (MSE): 3463319.5471278434
Ridge Mean Absolute Error (MAE): 1436.9858096538003
Ridge Root Mean Squared Error: 1860.9996096527918
df results = pd.DataFrame(y test price, columns=['price'])
df_results = pd.DataFrame(y_test_price)
df results['Ridge Prediction'] = Ridge predict
df results['Selisih price RR'] = df results['Ridge Prediction'] -
df results['price']
df results.head()
          price
                 Ridge Prediction Selisih price RR
4208
     7639752.5
                     7.639003e+06
                                        -749.746498
3619
     9873512.3
                     9.874681e+06
                                        1169.094552
5826
     1397748.9
                     1.398107e+06
                                         357.831109
6538
      1620485.0
                     1.622079e+06
                                        1594.441132
8787 4872012.2
                     4.872000e+06
                                         -12.548969
df results.describe()
              price
                     Ridge Prediction
                                      Selisih_price_RR
count 2.500000e+03
                         2.500000e+03
                                            2500.000000
mean
       4.964371e+06
                         4.964383e+06
                                              12.219481
                                            1861.331796
std
       2.842791e+06
                         2.842819e+06
       1.443130e+04
                         1.647641e+04
                                           -6523.351317
min
      2.567703e+06
                         2.567287e+06
                                           -1146.123872
25%
50%
      4.998880e+06
                         4.999651e+06
                                              42.904055
75%
       7.391681e+06
                         7.392256e+06
                                            1169.838143
      1.000294e+07
                         1.000120e+07
                                            7046.231520
max
from sklearn.svm import SVR
from sklearn.model selection import GridSearchCV
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.feature selection import SelectKBest, f regression
from sklearn.metrics import mean absolute error, mean squared error
pipe SVR = Pipeline(steps=[
    ('scale', StandardScaler()),
    ('feature_selection', SelectKBest(score_func=f_regression)),
    ('reg', SVR(kernel='linear'))
1)
param grid SVR = {
    'reg__C': [0.01, 0.1, 1, 10, 100],
    'reg epsilon': [0.1, 0.2, 0.5, 1],
```

```
'feature selection k': np.arange(1, 20)
}
GSCV_SVR = GridSearchCV(pipe_SVR, param_grid_SVR, cv=5,
scoring='neg mean squared error')
GSCV SVR.fit(x train price enc, y train price)
print("Best model: {}".format(GSCV SVR.best estimator ))
print("SVR best parameters: {}".format(GSCV_SVR.best_params_))
print("Koefisien/bobot:
{}".format(GSCV_SVR.best estimator .named steps['reg'].coef ))
print("Intercept/bias:
{}".format(GSCV SVR.best estimator .named steps['reg'].intercept ))
SVR predict = GSCV SVR.predict(x_test_price_enc)
mse_SVR = mean_squared_error(y_test_price, SVR_predict)
mae_SVR = mean_absolute_error(y_test_price, SVR_predict)
print("SVR Mean Squared Error (MSE): {}".format(mse SVR))
print("SVR Mean Absolute Error (MAE): {}".format(mae SVR))
print("SVR Root Mean Squared Error: {}".format(np.sqrt(mse SVR)))
Best model: Pipeline(steps=[('scale', StandardScaler()),
                ('feature selection',
                 SelectKBest(k=4,
                             score func=<function f regression at
0 \times 000001B4370873A0 > )),
                ('reg', SVR(C=100, kernel='linear'))])
SVR best parameters: {'feature selection k': 4, 'reg C': 100,
'reg epsilon': 0.1}
Koefisien/bobot: [[649047.75506463 12391.30466663 14718.77900439 -
7693.2361229711
Intercept/bias: [5010394.44877383]
SVR Mean Squared Error (MSE): 4858447421454.491
SVR Mean Absolute Error (MAE): 1895594.4965093078
SVR Root Mean Squared Error: 2204188.6084122863
df_results['SVR Prediction'] = SVR_predict
df results = pd.DataFrame(y test price)
df results['SVR Prediction'] = SVR predict
df results['Selisih price SVR'] = df results['SVR Prediction'] -
df_results['price']
df results.head()
          price SVR Prediction Selisih price SVR
4208
     7639752.5
                   5.624434e+06
                                     -2.015318e+06
3619
      9873512.3
                   6.126296e+06
                                     -3.747217e+06
5826
      1397748.9
                   4.167064e+06
                                      2.769315e+06
```

```
4.220983e+06
                                     2.600498e+06
6538
     1620485.0
                  4.968442e+06
                                     9.642940e+04
8787 4872012.2
df results.describe()
              price SVR Prediction
                                    Selisih price SVR
count 2.500000e+03
                      2.500000e+03
                                         2.500000e+03
mean 4.964371e+06
                       5.001619e+06
                                         3.724797e+04
std
      2.842791e+06
                      6.388973e+05
                                         2.204315e+06
min
     1.443130e+04
                      3.852418e+06
                                         -3.888487e+06
25%
      2.567703e+06
                      4.459952e+06
                                         -1.836367e+06
                                         1.100832e+04
50%
      4.998880e+06
                      5.006322e+06
75%
      7.391681e+06
                      5.544842e+06
                                         1.896727e+06
max 1.000294e+07
                      6.163456e+06
                                         3.877589e+06
print(df results.columns)
Index(['price', 'Ridge Prediction'], dtype='object')
print(len(y test price), len(Ridge predict), len(x test price enc)),
2500 2500 2500
(None,)
print(Ridge predict)
[7639002.75350174 9874681.39455215 1398106.7311088 ...
87279,62484006
 6585554.73333456 3660613.32301272]
# Pastikan Anda tidak membuat ulang df results setiap kali
df results = pd.DataFrame(y test price, columns=['price'])
# Tambahkan prediksi SVR terlebih dahulu
df results['SVR Prediction'] = SVR predict
# Tambahkan prediksi Ridge Regression setelahnya
df results['Ridge Prediction'] = Ridge predict
# Pastikan kedua kolom ada
print(df results.columns)
# Buat plot setelah memastikan semua kolom ada
plt.figure(figsize=(20, 5))
data len = range(len(y test price))
plt.scatter(data len, df results['price'], label="actual",
color="blue")
plt.plot(data len, df results['Ridge Prediction'], label="Ridge
Prediction", color="green", linewidth=2, linestyle="dashed")
plt.plot(data_len, df_results['SVR Prediction'], label="SVR
```

```
Prediction", color="yellow", linewidth=1, linestyle="-.")
plt.legend()
plt.show()
Index(['price', 'SVR Prediction', 'Ridge Prediction'], dtype='object')
```



```
from sklearn.metrics import mean absolute error, mean squared error
import numpy as np
mae ridge = mean absolute error(df results['price'], df results['Ridge
Prediction'l)
rmse ridge = np.sqrt(mean squared error(df results['price'],
df_results['Ridge Prediction']))
ridge feature count = GSCV RR.best params ['feature selection k']
mae svr = mean absolute error(df results['price'], df results['SVR
Prediction'l)
rmse svr = np.sqrt(mean squared error(df results['price'],
df results['SVR Prediction']))
svr feature count = GSCV SVR.best params ['feature_selection_k']
print(f"Ridge MAE: {mae ridge}, Ridge RMSE: {rmse ridge}, Ridge
Feature Count: {ridge_feature_count}")
print(f"SVR MAE: {mae svr}, SVR RMSE: {rmse svr}, SVR Feature Count:
{svr feature count}")
Ridge MAE: 1436.9858096538003, Ridge RMSE: 1860.9996096527918, Ridge
Feature Count: 19
SVR MAE: 1895594.4965093078, SVR RMSE: 2204188.6084122863, SVR Feature
Count: 4
```

```
615
10
                  2
                                   9
                                      2008
                                                   new
                                                                         no
2860
                  8
                                  10
                                      2000
                                                   old
11
                                                                         no
6629
                  6
                                   2
12
                                      2001
                                                   old
                                                                         no
7473
13
                  7
                                   7
                                      2000
                                                   new
                                                                         no
3573
                  5
                                   8
14
                                      2016
                                                   new
                                                                         no
8150
                  9
                                   2
                                      2021
                                                   old
15
                                                                         no
1559
                  6
                                   9
                                      2021
                                                   old
16
                                                                         no
5075
                                   5
                 10
                                      2020
17
                                                   new
                                                                         no
5278
                                   6
                                      1992
                                                   old
18
                  4
                                                                        yes
4480
19
                  3
                                   7
                                      2021
                                                   old
                                                                         no
231
    attic
            garage hasstorageroom
                                      hasguestroom
                                                          price category
0
               956
                                                      7559081.5
     9005
                                                                   Luxury
                                  no
                                                  7
     8852
               135
1
                                yes
                                                  9
                                                      5574642.1
                                                                   Middle
2
     4748
               654
                                                 10
                                                      8696869.3
                                                                   Luxury
                                 no
3
     5792
               807
                                                  5
                                                      5154055.2
                                                                   Middle
                                yes
4
     1172
               716
                                                  9
                                                      9652258.1
                                                                   Luxury
                                yes
5
                                                  7
     7117
               240
                                                      7986665.8
                                 no
                                                                   Luxury
6
      281
                                                  5
               384
                                                      7607322.9
                                                                   Luxury
                                yes
7
                                                  9
      129
               726
                                                                   Middle
                                                      6420823.1
                                  no
8
     9056
                                                  1
                                                      9244344.0
               892
                                yes
                                                                   Luxury
9
     1221
               328
                                                 10
                                                      9515440.4
                                 no
                                                                   Luxury
10
     3129
               982
                                  no
                                                  1
                                                      7653300.8
                                                                   Luxury
                                                  7
11
      435
               512
                                 no
                                                      8711426.0
                                                                   Luxury
                                                  3
12
      796
               237
                                                      6677649.1
                                                                   Middle
                                yes
13
     9556
               918
                                                  8
                                yes
                                                      8460604.0
                                                                   Luxury
14
     6037
               930
                                                  7
                                                      7614076.6
                                                                   Luxury
                                 no
15
     5111
               957
                                                  2
                                                      9272740.1
                                                                   Luxury
                                yes
16
     3104
               864
                                                  4
                                                      5984462.1
                                                                   Middle
                                 no
17
     1059
               313
                                                  6
                                                      5492532.0
                                                                   Middle
                                yes
18
                                                  1
     6919
               680
                                                      7005572.2
                                                                   Luxury
                                yes
19
     1939
               223
                                                  8
                                 no
                                                      5446398.1
                                                                   Middle
df_harga2 = df_harga.drop(['category'], axis = 1)
df harga2.head(20)
    squaremeters
                    numberofrooms hasyard haspool
                                                       floors
                                                                citycode \
0
                                                                    9373
                                 3
                                                           63
            75523
                                         no
                                                 yes
1
                                58
                                                           19
                                                                   34457
            55712
                                         no
                                                 yes
```

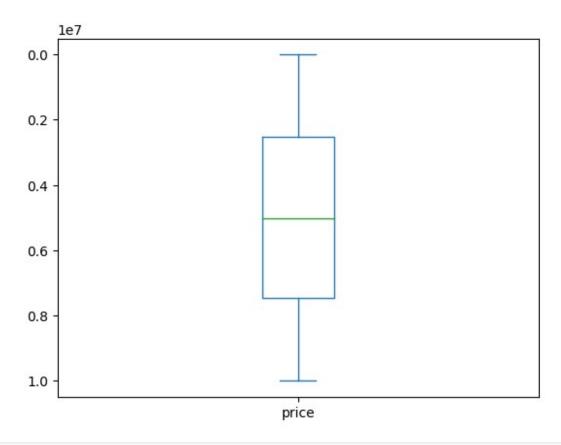
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	86929 51522 96470 79770 75985 64169 92383 95121 76485 87060 66683 84559 76091 92696 59800 54836 70021 54368	100 3 74 3 60 88 12 46 47 27 19 29 38 49 47 25 52 11	yes no yes no no no yes no yes no yes no yes yes no yes yes yes	no no yes no yes no yes yes yes no no yes	11 61 21 69 67 6 78 3 9 91 6 69 32 38 27 53 28 20	98155 9047 92029 54812 6517 61711 71982 9382 90254 51803 50801 53057 59451 74381 44815 64601 95678 55761
	artrange	numprevowners	_	yes isnewbuilt		
basement	\	•			iiass coriii	
0 4313	3	8	2005	old		yes
1	6	8	2021	old		no
2937 2	3	4	2003	new		no
6326				new		110
3 632	8	3	2012	new		yes
4	4	2	2011	new		yes
5414	10	_	2010	1.		-
5 8871	10	5	2018	old		yes
6	6	9	2009	new		yes
4878 7	3	9	2011	no. /		V05
3054	3	9	2011	new		yes
8	3	7	2000	old		no
7507 9	7	9	1994	old		no
615		9	1334	o cu		110
10	2	9	2008	new		no
2860 11	8	10	2000	old		no
6629						110
12	6	2	2001	old		no
7473 13	7	7	2000	new		no
3573	,					

14		5		8	2016		ne	W	no
8150 15	j	9		2	2021		οl	d	no
1559 16	)	6		9	2021		ol	Ч	no
5075	5								
17 5278	3	10		5	2020		ne	W	no
18 4480		4		6	1992		οl	d	yes
19	,	3		7	2021		οl	d	no
231									
<class< td=""><td>geIndex</td><td>ndas.com : 10000 ns (tota</td><td>re.frame.Dat entries, 0 al 17 column</td><td>no yes no yes yes no yes no yes no yes yes no yes yes no aFra 9 s):</td><td>ıme'&gt;</td><td>estroom 7 9 10 5 9 7 5 9 1 10 1 7 3 8 7 2 4 6 1 8</td><td>7 9 9 9 9 9 9 9 9 9</td><td>price 7559081.5 5574642.1 8696869.3 5154055.2 9652258.1 7986665.8 7607322.9 6420823.1 9244344.0 9515440.4 7653300.8 8711426.0 6677649.1 8460604.0 7614076.6 9272740.1 5984462.1 5984462.1 5492532.0 7005572.2 5446398.1</td><td></td></class<>	geIndex	ndas.com : 10000 ns (tota	re.frame.Dat entries, 0 al 17 column	no yes no yes yes no yes no yes no yes yes no yes yes no aFra 9 s):	ıme'>	estroom 7 9 10 5 9 7 5 9 1 10 1 7 3 8 7 2 4 6 1 8	7 9 9 9 9 9 9 9 9 9	price 7559081.5 5574642.1 8696869.3 5154055.2 9652258.1 7986665.8 7607322.9 6420823.1 9244344.0 9515440.4 7653300.8 8711426.0 6677649.1 8460604.0 7614076.6 9272740.1 5984462.1 5984462.1 5492532.0 7005572.2 5446398.1	
0	squar	- emeters	10000	non	 -null	 int64			
1 2		rofrooms	s 10000	non	-null -null	int64 object			
3 4	haspo	ol	10000	non	-null -null	object int64			
5	cityc	ode	10000	non	-null	int64			
6 7		artrange evowners			ı-null ı-null	int64 int64			

9 is 10 ha 11 ba 12 as 13 ga 14 ha 15 ha 16 p dtypes memory	aseme ttic arage assto asgue rice : flo usag	ermprotect ent e erageroom estroom	10000 non-r 10000 non-r 10000 non-r 10000 non-r 10000 non-r int64(11), obje	null object null object null int64 null int64 null int64 null object null int64 null int64	
ar_nar				67	
citypa		remeters	numberofrooms	floors	citycode
count	100	00.00000	10000.000000	10000.000000	10000.000000
10000. mean		00 370.13120	50.358400	50.276300	50225.486100
5.5101	00				
std 2.8720		74.37535	28.816696	28.889171	29006.675799
min		89.00000	1.000000	1.000000	3.000000
1.0000 25%		98.50000	25.000000	25.000000	24693.750000
3.0000 50%		.05.50000	50.000000	50.000000	50693.000000
5.0000		.00.000	30.000000	30.000000	30093.000000
75% 8.0000		09.75000	75.000000	76.000000	75683.250000
max	999	99.00000	100.000000	100.000000	99953.000000
10.000	000				
as no a o		revowners	made	basement	attic
garage count 10000.	100	00.00000	10000.00000	10000.000000	10000.00000
mean		5.521700	2005.48850	5033.103900	5028.01060
553.12 std	120	2.856667	9.30809	2876.729545	2894.33221
262.05	017				
min 100.00	000	1.000000	1990.00000	0.000000	1.00000
25% 327.75		3.000000	1997.00000	2559.750000	2512.00000
50%		5.000000	2005.50000	5092.500000	5045.00000
554.000 75%	000	8.000000	2014.00000	7511.250000	7540.50000
777.25	000				
max		10.000000	2021.00000	10000.000000	10000.00000

```
1000.00000
       hasguestroom
                             price
        10000.00000
                      1.000000e+04
count
            4.99460
                     4.993448e+06
mean
            3.17641
                      2.877424e+06
std
                     1.031350e+04
            0.00000
min
25%
            2.00000 2.516402e+06
            5.00000 5.016180e+06
50%
            8.00000 7.469092e+06
75%
           10.00000 1.000677e+07
max
print(df harga2['price'].value counts())
price
7559081.5
             1
2600292.1
             1
3804577.4
             1
3658559.7
             1
2316639.4
             1
5555606.6
             1
5501007.5
             1
9986201.2
             1
9104801.8
             1
146708.4
             1
Name: count, Length: 10000, dtype: int64
print("\tdata null\n", df_harga2.isnull().sum())
print("\n\tdata kosong\n", df_harga2.empty)
print("\n\tdata nan\n", df harga2.isna().sum())
     data null
                      0
 squaremeters
numberofrooms
                      0
                      0
hasyard
                      0
haspool
floors
                      0
                      0
citycode
                      0
citypartrange
                      0
numprevowners
made
                      0
                      0
isnewbuilt
                      0
hasstormprotector
                      0
basement
                      0
attic
                      0
garage
                      0
hasstorageroom
                      0
hasguestroom
                      0
price
```

```
dtype: int64
      data kosong
 False
      data nan
 squaremeters
                        0
numberofrooms
                       0
hasyard
                       0
haspool
                       0
                       0
floors
                       0
citycode
                       0
citypartrange
numprevowners
                       0
made
                       0
                       0
isnewbuilt
                       0
hasstormprotector
                       0
basement
attic
                       0
                       0
garage
                       0
hasstorageroom
                       0
hasquestroom
                       0
price
dtype: int64
print("Sebelum Pengecekan data duplikat, ",df_harga2.shape)
df_harga3 = df_harga2.drop_duplicates(keep = 'last')
print("Setelah Pengecekan data duplikat, ",df_harga2.shape)
Sebelum Pengecekan data duplikat, (10000, 17)
Setelah Pengecekan data duplikat, (10000, 17)
import matplotlib.pyplot as plt
df harga3.price.plot(kind='box')
plt.gca().invert yaxis()
plt.show
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
from sklearn.model selection import train test split
x regress = df harga3.drop(columns=['price'], axis = 1)
y regress = df harga3['price']
x train price, x test price, y train price, y test price =
train test split(x regress, y regress, test size = 0.25, random state
= 78)
print(x_train_price.shape)
print(x_test_price.shape)
(7500, 16)
(2500, 16)
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make column transformer
category_column = ['hasyard', 'haspool', 'isnewbuilt',
'hasstormprotector', 'hasstorageroom']
transform = make column transformer((OneHotEncoder(),
category column), remainder = 'passthrough')
```

```
x train price enc = transform.fit transform(x train price)
x test price enc = transform.fit transform(x test price)
df train enc = pd.DataFrame(x train price enc, columns =
transform.get feature names out())
df_test_enc = pd.DataFrame(x_test_price_enc, columns =
transform.get_feature_names_out())
df train enc.head(20)
df test enc.head(20)
    onehotencoder__hasyard_no
                                  onehotencoder__hasyard_yes \
                             \overline{1}.0
0
                                                           0.0
1
                            0.0
                                                           1.0
2
                                                           0.0
                             1.0
3
                             1.0
                                                           0.0
4
                            0.0
                                                           1.0
5
                            0.0
                                                           1.0
6
                                                           1.0
                            0.0
7
                                                           1.0
                            0.0
8
                            0.0
                                                           1.0
9
                            1.0
                                                           0.0
10
                            0.0
                                                           1.0
                                                           1.0
11
                            0.0
12
                             1.0
                                                           0.0
13
                            0.0
                                                           1.0
14
                             1.0
                                                           0.0
15
                            0.0
                                                           1.0
16
                             1.0
                                                           0.0
17
                            0.0
                                                           1.0
18
                            0.0
                                                           1.0
19
                                                           0.0
                             1.0
    onehotencoder__haspool_no
                                  onehotencoder__haspool_yes \
0
                                                            1.0
                             0.0
1
                            0.0
                                                           1.0
2
                             1.0
                                                           0.0
3
                            0.0
                                                           1.0
4
                                                           0.0
                             1.0
5
                            0.0
                                                           1.0
6
                            0.0
                                                           1.0
7
                            0.0
                                                           1.0
8
                            1.0
                                                           0.0
9
                            1.0
                                                           0.0
10
                            0.0
                                                           1.0
11
                             1.0
                                                           0.0
12
                             1.0
                                                           0.0
13
                             1.0
                                                           0.0
14
                             1.0
                                                           0.0
15
                             1.0
                                                           0.0
```

```
16
                             1.0
                                                            0.0
17
                                                            1.0
                             0.0
                                                            0.0
18
                             1.0
19
                                                            0.0
                             1.0
                                       onehotencoder__isnewbuilt_old \
    onehotencoder__isnewbuilt_new
0
                                 0.0
                                                                    1.0
1
                                 0.0
                                                                    1.0
2
                                 0.0
                                                                    1.0
                                 1.0
                                                                    0.0
4
                                 1.0
                                                                    0.0
5
6
                                 0.0
                                                                    1.0
                                 0.0
                                                                    1.0
7
                                 0.0
                                                                    1.0
8
                                 1.0
                                                                    0.0
9
                                 1.0
                                                                    0.0
10
                                 0.0
                                                                    1.0
11
                                 0.0
                                                                    1.0
12
                                 1.0
                                                                    0.0
13
                                 1.0
                                                                    0.0
14
                                 0.0
                                                                    1.0
15
                                 1.0
                                                                    0.0
16
                                 1.0
                                                                    0.0
17
                                 1.0
                                                                    0.0
18
                                 1.0
                                                                    0.0
19
                                 1.0
                                                                    0.0
    onehotencoder__hasstormprotector_no
onehotencoder_hasstormprotector_yes \
                                        1.0
0.0
                                        0.0
1
1.0
2
                                        1.0
0.0
                                        1.0
3
0.0
                                        1.0
4
0.0
                                        0.0
5
1.0
                                        0.0
6
1.0
                                        0.0
7
1.0
8
                                        0.0
1.0
9
                                        1.0
0.0
```

10		1.0
0.0		0.0
11 1.0		0.0
12		0.0
1.0		
13		1.0
0.0		1.0
14		1.0
0.0 15		1.0
0.0		1.0
16		1.0
0.0		
17		1.0
0.0		
18		1.0
0.0 19		1.0
0.0		1.0
0.0		
	oneho	otencoderhasstorageroom_no
oneh	oten	coderhasstorageroom_yes \
0		1.0
0.0 1		0.0
1.0		0.0
2		0.0
2 1.0		
3		0.0
1.0		
4		1.0
0.0		0.0
5 1.0		0.0
6		0.0
6 1.0		
7		0.0
1.0		
8		0.0
1.0		0.0
9 1.0		0.0
10		0.0
1.0		
11		1.0
0.0		
12		0.0
1.0		

13 1.0		0.0	
14	• • •	1.0	
0.0 15	• • •	0.0	
1.0 16		0.0	
1.0 17		1.0	
0.0 18		0.0	
1.0 19		0.0	
1.0		0.0	
rom	remaindernumberofrooms	remainderfloors	
0	aindercitycode \ 73.0	13.0	42855.0
1	95.0	3.0	75381.0
2	39.0	8.0	91674.0
3	47.0	63.0	58471.0
4	64.0	83.0	30779.0
5	91.0	67.0	65183.0
6	48.0	70.0	21012.0
7	93.0	35.0	12062.0
8	74.0	14.0	76662.0
9	72.0	76.0	87732.0
10	18.0	66.0	38920.0
11	37.0	26.0	96016.0
12	76.0	95.0	76985.0
13	26.0	99.0	38185.0
14	69.0	45.0	88591.0
15	9.0	25.0	33740.0
16	29.0	58.0	13202.0

17	91.0	43.0	93072.0
18	68.0	62.0	97608.0
19	48.0	53.0	34588.0
	.0.0	33.10	5.300.0
ren	naindercitypartrange	remaindernumprevowners	
remaind 0	lermade \ 9.0	6.0	
2015.0 1		6.0	
2003.0	5.0		
2 2009.0	2.0	2.0	
3	10.0	1.0	
1990.0 4	6.0	4.0	
1992.0 5	1.0	10.0	
2019.0			
6 2007.0	5.0	3.0	
7 1998.0	10.0	9.0	
8	9.0	6.0	
2004.0	2.0	8.0	
2017.0 10	7.0	2.0	
2007.0			
11 2000.0	3.0	6.0	
12	4.0	7.0	
2009.0 13	7.0	7.0	
1994.0 14	1.0	8.0	
1995.0			
15 2009.0	2.0	1.0	
16 1993.0	6.0	5.0	
17	1.0	7.0	
2012.0 18	1.0	9.0	
2001.0 19	9.0	6.0	
2003.0	3.0	0.0	

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	remainderbasement	remainderattic 6823.0 1391.0 3153.0 7967.0 8310.0 7141.0 2750.0 2720.0 9579.0 6111.0 5811.0 6205.0 6017.0 1109.0 8363.0 1494.0 5829.0 4961.0 336.0 3418.0	remaindergarage 239.0 556.0 952.0 722.0 898.0 956.0 652.0 315.0 768.0 613.0 185.0 850.0 472.0 827.0 827.0 876.0 567.0 484.0 893.0 656.0 162.0	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	remainderhasguestr		102.10	

## [20 rows x 21 columns]

```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV, KFold
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.feature_selection import SelectKBest, f_regression,
```

```
SelectPercentile
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean absolute error, mean squared error
pipe Lasso = Pipeline(steps = [
    ('scale', StandardScaler()),
    ('feature', SelectKBest(score func = f regression)),
    ('reg', Lasso(max iter = 1000))
])
param grid Lasso = [
    {
        'scale' : [StandardScaler()],
        'feature' : [SelectKBest(f regression)],
        'feature k': np.arange(1, 20),
        'reg alpha': [0.01, 0.1, 1, 10, 100]
    },
    {
        'scale' : [StandardScaler()],
        'feature' : [SelectPercentile(f_regression)],
        'feature percentile': np.arange(10, 100, 10),
        'reg alpha': [0.01, 0.1, 1, 10, 100]
    },
    {
        'scale' : [MinMaxScaler()],
        'feature' : [SelectKBest(f regression)],
        'feature_k' : np.arange(1, 20),
        'reg alpha' : [0.01, 0.1, 1, 10, 100]
    },
    {
        'scale' : [MinMaxScaler()],
        'feature' : [SelectPercentile(f regression)],
        'feature percentile' : np.arange(10, 100, 10),
        'reg alpha': [0.01, 0.1, 1, 10, 100]
    }
1
KF = KFold(n splits = 5, shuffle = True, random state = 78)
GSCV Lasso = GridSearchCV(pipe Lasso, param grid Lasso, cv = KF,
scoring = 'neg mean squared error')
GSCV Lasso.fit(x train price enc, y train price)
print("Best Model: {}".format(GSCV Lasso.best estimator ))
mask = GSCV Lasso.best estimator .named steps['feature'].get support()
print("Best features:",df train enc.columns[mask])
```

```
print("Koefisien/Bobot:
{}".format(GSCV Lasso.best_estimator_.named_steps['reg'].coef_))
print("Intercept/Bias:
{}".format(GSCV Lasso.best estimator .named steps['reg'].intercept ))
Lasso predict = GSCV Lasso.predict(x test price enc)
mae Lasso = mean absolute error(y test price, Lasso predict)
mse_Lasso = mean_squared_error(y_test_price, Lasso_predict)
print("Lasso Regression MAE: {}".format(mae Lasso))
print("Lasso Regression MSE: {}".format(mse_Lasso))
print("Lasso Regression RMSE: {}".format(np.sqrt(mse Lasso)))
Best Model: Pipeline(steps=[('scale', StandardScaler()),
                ('feature',
                 SelectKBest(k=19,
                             score_func=<function f_regression at</pre>
0 \times 000001E948B663A0 >)),
                ('reg', Lasso(alpha=10))])
Best features: Index(['onehotencoder hasyard no',
'onehotencoder hasyard yes',
       'onehotencoder haspool no', 'onehotencoder haspool yes',
       'onehotencoder__isnewbuilt_new',
'onehotencoder isnewbuilt old',
       'onehotencoder hasstormprotector no',
       'onehotencoder_hasstormprotector_yes',
       'onehotencoder hasstorageroom no',
'onehotencoder hasstorageroom yes',
       'remainder__squaremeters', 'remainder__numberofrooms',
       'remainder floors', 'remainder citypartrange',
       'remainder numprevowners', 'remainder__basement',
'remainder attic',
       'remainder garage', 'remainder hasguestroom'],
      dtype='object')
Koefisien/Bobot: [-1.50189031e+03 4.77302819e-13 -1.47466707e+03
1.14475066e-13
  8.29146159e+01 -0.00000000e+00 -7.13309651e+01 5.42883451e-12
 -7.55801132e-01 5.86364573e-11 2.88881051e+06 0.00000000e+00
 1.54979947e+03 1.31187184e+02 -0.00000000e+00 -3.74045759e+00
 -8.09905705e+00 2.77422673e+01 -7.45876850e+001
Intercept/Bias: 5003139.741906667
Lasso Regression MAE: 1436.6510138328122
Lasso Regression MSE: 3460594.830991738
Lasso Regression RMSE: 1860.2674084635623
df results = pd.DataFrame(y test price, columns=['price'])
df results = pd.DataFrame(y test price)
```

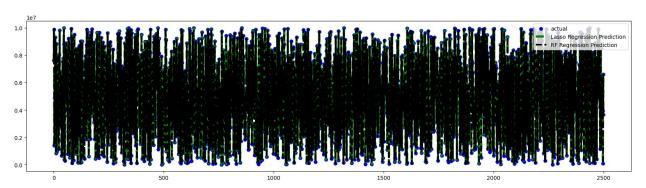
```
df results['Lasso Regression Prediction'] = Lasso predict
df results['Selisih Harga LR'] = df results['Lasso Regression
Prediction'] - df results['price']
df results.head()
          price Lasso Regression Prediction Selisih Harga LR
4208
      7639752.5
                                                   -703.338378
                                7.639049e+06
3619
      9873512.3
                                                   1124.307238
                                9.874637e+06
5826
      1397748.9
                                1.398120e+06
                                                    371.343414
6538
     1620485.0
                                1.622021e+06
                                                   1535.868119
                                4.871971e+06
                                                    -40.732719
8787 4872012.2
df results.describe()
                     Lasso Regression Prediction
                                                  Selisih Harga LR
              price
count 2.500000e+03
                                    2.500000e+03
                                                       2500.000000
       4.964371e+06
mean
                                    4.964382e+06
                                                         11.600983
       2.842791e+06
                                    2.842814e+06
                                                       1860.603393
std
min
      1.443130e+04
                                    1.647411e+04
                                                       -6534.806896
25%
      2.567703e+06
                                    2.567290e+06
                                                      -1158.542087
50%
      4.998880e+06
                                    4.999656e+06
                                                         45.301221
75%
       7.391681e+06
                                    7.392305e+06
                                                       1161.233831
       1.000294e+07
                                    1.000122e+07
                                                       6976.939177
max
from sklearn.ensemble import RandomForestRegressor
from sklearn.model selection import GridSearchCV, KFold
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.feature selection import SelectKBest, f regression,
SelectPercentile
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean absolute error, mean squared error
pipe RF = Pipeline(steps = [
    ('scale', StandardScaler()),
    ('feature', SelectKBest(score func = f regression)),
    ('reg', RandomForestRegressor(random state = 78))
])
param_grid_RF = [
        'scale' : [StandardScaler()],
        'feature' : [SelectKBest(f regression)],
        'feature k' : np.arange(1, 20),
        'reg n estimators' : [100, 150],
        'reg max depth' : [4, 5, 6]
    },
    {
```

```
'scale' : [StandardScaler()],
        'feature' : [SelectPercentile(f regression)],
        'feature__percentile' : np.arange(10, 100, 10),
        'reg__n_estimators' : [100, 150],
        'reg max depth' : [4, 5, 6]
    },
    {
        'scale' : [MinMaxScaler()],
        'feature' : [SelectKBest(f regression)],
        'feature__k' : np.arange(1, 20),
        'reg n estimators' : [100, 150],
        'reg max depth' : [4, 5, 6]
    },
    {
        'scale' : [MinMaxScaler()],
        'feature' : [SelectPercentile(f regression)],
        'feature__percentile' : np.arange(10, 100, 10),
        'reg n estimators' : [100, 150],
        'reg max depth' : [4, 5, 6]
    }
]
KF = KFold(n splits = 5, shuffle = True, random state = 78)
GSCV RF = GridSearchCV(pipe RF, param grid RF, cv = KF, scoring =
'neg mean squared error')
GSCV RF.fit(x train price enc, y train price)
print("Best Model: {}".format(GSCV RF.best estimator ))
mask = GSCV_RF.best_estimator_.named_steps['feature'].get_support()
print("Best features:",df train enc.columns[mask])
RF predict = GSCV RF.predict(x test price enc)
mae_RF = mean_absolute_error(y_test_price, RF_predict)
mse RF = mean squared error(y test price, RF predict)
print("Random Forest Regression MAE: {}".format(mae RF))
print("Random Forest Regression MSE: {}".format(mse RF))
print("Random Forest Regression RMSE: {}".format(np.sqrt(mse RF)))
Best Model: Pipeline(steps=[('scale', StandardScaler()),
                ('feature'.
                 SelectKBest(k=6,
                             score func=<function f regression at
0 \times 000001E948B663A0 > )),
                ('reg',
```

```
RandomForestRegressor(max depth=6, n estimators=150,
                                       random state=78))])
Best features: Index(['onehotencoder_hasstormprotector_yes',
'remainder squaremeters',
       'remainder__numberofrooms', 'remainder numprevowners',
       'remainder__attic', 'remainder__garage'],
      dtype='object')
Random Forest Regression MAE: 16083.918675581288
Random Forest Regression MSE: 412641335.1781159
Random Forest Regression RMSE: 20313.575145161325
df results['RF Regression Prediction'] = RF predict
df results = pd.DataFrame(y test price)
df results['RF Regression Prediction'] = RF predict
df results['Selisih Harga RFR'] = df results['RF Regression
Prediction'] - df results['price']
df results.head()
          price RF Regression Prediction Selisih Harga RFR
4208
     7639752.5
                             7.638896e+06
                                                 -856.207487
3619 9873512.3
                             9.921015e+06
                                                47502.930129
                                                49411.957540
5826
     1397748.9
                             1.447161e+06
                            1.631266e+06
                                                10780.865479
6538 1620485.0
8787 4872012.2
                            4.886406e+06
                                               14393.606394
df results.describe()
              price RF Regression Prediction Selisih Harga RFR
count 2.500000e+03
                                 2.500000e+03
                                                     2500.000000
      4.964371e+06
                                4.964135e+06
                                                     -235.718277
mean
                                                    20316.271125
std
      2.842791e+06
                                2.842884e+06
      1.443130e+04
                                8.363800e+04
                                                   -81929.769871
min
25%
      2.567703e+06
                                2.554733e+06
                                                   -14193.902928
50%
      4.998880e+06
                                5.022442e+06
                                                       20.626003
75%
      7.391681e+06
                                7.407247e+06
                                                    12956.223520
max 1.000294e+07
                                9.921015e+06
                                               69206.702595
df results = pd.DataFrame({'price': y_test_price})
df results['Lasso Regression Prediction'] = Lasso predict
df results['Selisih Harga LR'] = df results['price'] -
df results['Lasso Regression Prediction']
df results['RF Regression Prediction'] = RF predict
df results['Selisih Harga RFR'] = df results['price'] - df results['RF
Regression Prediction']
df results.head()
```

```
Selisih Harga LR \
                 Lasso Regression Prediction
          price
                                                      703.338378
4208
      7639752.5
                                 7.639049e+06
3619
      9873512.3
                                 9.874637e+06
                                                    -1124.307238
5826
      1397748.9
                                 1.398120e+06
                                                     -371.343414
6538
      1620485.0
                                 1.622021e+06
                                                    -1535.868119
8787
      4872012.2
                                 4.871971e+06
                                                       40.732719
      RF Regression Prediction
                                 Selisih Harga RFR
                                        856, 207487
4208
                  7.638896e+06
3619
                                     -47502.930129
                  9.921015e+06
5826
                  1.447161e+06
                                     -49411.957540
6538
                  1.631266e+06
                                     -10780.865479
8787
                  4.886406e+06
                                     -14393.606394
df results.describe()
                     Lasso Regression Prediction
                                                   Selisih Harga LR \
              price
       2.500000e+03
                                     2.500000e+03
                                                         2500.000000
count
mean
       4.964371e+06
                                     4.964382e+06
                                                          -11.600983
                                                         1860.603393
       2.842791e+06
                                     2.842814e+06
std
min
       1.443130e+04
                                     1.647411e+04
                                                        -6976.939177
                                     2.567290e+06
25%
       2.567703e+06
                                                        -1161.233831
50%
       4.998880e+06
                                     4.999656e+06
                                                          -45.301221
75%
       7.391681e+06
                                     7.392305e+06
                                                         1158.542087
       1.000294e+07
                                     1.000122e+07
max
                                                         6534.806896
       RF Regression Prediction Selisih Harga RFR
                   2.500000e+03
                                        2500,000000
count
                   4.964135e+06
                                         235.718277
mean
                   2.842884e+06
                                       20316.271125
std
                   8.363800e+04
                                      -69206.702595
min
25%
                   2.554733e+06
                                      -12956.223520
50%
                   5.022442e+06
                                         -20.626003
75%
                   7.407247e+06
                                       14193.902928
                   9.921015e+06
                                       81929.769871
max
import matplotlib.pyplot as plt
plt.figure(figsize = (20,5))
data_len = range(len(y_test_price))
plt.scatter(data len, df results.price, label = "actual", color =
"blue")
plt.plot(data len, df results['Lasso Regression Prediction'], label =
"Lasso Regression Prediction", color = "green", linewidth = 4,
linestyle = "dashed")
plt.plot(data len, df results['RF Regression Prediction'], label = "RF
Regression Prediction, color = "black, linewidth = 3, linestyle =
" - - " )
```

```
plt.legend()
plt.show
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
from sklearn.metrics import mean absolute error, mean squared error
import numpy as np
mae lasso = mean absolute error(df results['price'], df results['Lasso
Regression Prediction'])
rmse lasso = np.sqrt(mean squared error(df results['price'],
df results['Lasso Regression Prediction']))
lasso feature count = GSCV Lasso.best params ['feature k']
mae RFR = mean absolute error(df results['price'], df results['RF
Regression Prediction'])
rmse RFR = np.sqrt(mean squared error(df results['price'],
df results['RF Regression Prediction']))
RFR feature count = GSCV RF.best params ['feature k']
print(f"Lasso MAE: {mae lasso}, Lasso RMSE: {rmse lasso}, Lasso
Feature Count: {lasso feature count}")
print(f"RFR MAE: {mae RFR}, RFR RMSE: {rmse RFR}, RFR Feature Count:
{RFR feature count}")
Lasso MAE: 1436.6510138328122, Lasso RMSE: 1860.2674084635623, Lasso
Feature Count: 19
RFR MAE: 16083.918675581288, RFR RMSE: 20313.575145161325, RFR Feature
Count: 6
```