assigment2MLDat158

November 15, 2023

1 House pricing prediction

[1]:
```
#biblotek

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import RandomizedSearchCV
```

2 Begynner med å importere dataen.

[3]:
```
trainingData=pd.read_csv('train.csv')
trainingData.head()
```

[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	…	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	…	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	…	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	…	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	…	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	…	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000

[5 rows x 81 columns]

[4]:
```
testData = pd.read_csv('test.csv')
testData.head()
```

[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	

	LandContour	Utilities	…	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature	\
0	Lvl	AllPub	…	120	0	NaN	MnPrv	NaN	
1	Lvl	AllPub	…	0	0	NaN	NaN	Gar2	
2	Lvl	AllPub	…	0	0	NaN	MnPrv	NaN	
3	Lvl	AllPub	…	0	0	NaN	NaN	NaN	
4	HLS	AllPub	…	144	0	NaN	NaN	NaN	

	MiscVal	MoSold	YrSold	SaleType	SaleCondition
0	0	6	2010	WD	Normal
1	12500	6	2010	WD	Normal
2	0	3	2010	WD	Normal
3	0	6	2010	WD	Normal
4	0	1	2010	WD	Normal

[5 rows x 80 columns]

3 Begynner med å ryddde opp i dataen

[6]:
```
trainingData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
```

```
12  Neighborhood   1460 non-null   object
13  Condition1     1460 non-null   object
14  Condition2     1460 non-null   object
15  BldgType       1460 non-null   object
16  HouseStyle     1460 non-null   object
17  OverallQual    1460 non-null   int64
18  OverallCond    1460 non-null   int64
19  YearBuilt      1460 non-null   int64
20  YearRemodAdd   1460 non-null   int64
21  RoofStyle      1460 non-null   object
22  RoofMatl       1460 non-null   object
23  Exterior1st    1460 non-null   object
24  Exterior2nd    1460 non-null   object
25  MasVnrType     588 non-null    object
26  MasVnrArea     1452 non-null   float64
27  ExterQual      1460 non-null   object
28  ExterCond      1460 non-null   object
29  Foundation     1460 non-null   object
30  BsmtQual       1423 non-null   object
31  BsmtCond       1423 non-null   object
32  BsmtExposure   1422 non-null   object
33  BsmtFinType1   1423 non-null   object
34  BsmtFinSF1     1460 non-null   int64
35  BsmtFinType2   1422 non-null   object
36  BsmtFinSF2     1460 non-null   int64
37  BsmtUnfSF      1460 non-null   int64
38  TotalBsmtSF    1460 non-null   int64
39  Heating        1460 non-null   object
40  HeatingQC      1460 non-null   object
41  CentralAir     1460 non-null   object
42  Electrical     1459 non-null   object
43  1stFlrSF       1460 non-null   int64
44  2ndFlrSF       1460 non-null   int64
45  LowQualFinSF   1460 non-null   int64
46  GrLivArea      1460 non-null   int64
47  BsmtFullBath   1460 non-null   int64
48  BsmtHalfBath   1460 non-null   int64
49  FullBath       1460 non-null   int64
50  HalfBath       1460 non-null   int64
51  BedroomAbvGr   1460 non-null   int64
52  KitchenAbvGr   1460 non-null   int64
53  KitchenQual    1460 non-null   object
54  TotRmsAbvGrd   1460 non-null   int64
55  Functional     1460 non-null   object
56  Fireplaces     1460 non-null   int64
57  FireplaceQu    770 non-null    object
58  GarageType     1379 non-null   object
59  GarageYrBlt    1379 non-null   float64
```

```
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

Dropper kolonner der nullverdiene utgjør mer en 50% av alle objektene. Så foreks kolonner som alley, PoolQc, Fence, og MiscFeature har veldig mangen null-verdier osm vil gå utover resultatet

[7]:
```
trainingData.drop(['Alley', 'PoolQC', 'Fence',␣
 ↪'MiscFeature','MasVnrType'],axis=1,inplace=True)
trainingData.drop(['Id'],axis=1,inplace=True)
```

Så må vi erstatte nullverdier i de andre kolonnene. Måten vi erstatter dem kommer litt ann på hvilke type kolonnen har. Med tall tar vi ofte bare å erstatter null-veridene med gjennomsnittet. Med kategoriske verdier er det kanskje lurere å erstatte dem med den mest hyppige typen.

Erstatter med gjennomsnitt:

[8]:
```
for column in trainingData.columns:
    if trainingData[column].dtype in ['int64', 'float64']:
        column_mean = trainingData[column].mean()
        trainingData[column].fillna(column_mean, inplace=True)
```

Erstatter med hyppigeste type

[9]:
```
for column in trainingData.columns:
    if trainingData[column].dtype == 'object':
        trainingData[column].fillna(trainingData[column].mode()[0],␣
 ↪inplace=True)
```

[10]:
```
trainingData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 75 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MSSubClass     1460 non-null   int64
 1   MSZoning       1460 non-null   object
 2   LotFrontage    1460 non-null   float64
 3   LotArea        1460 non-null   int64
 4   Street         1460 non-null   object
 5   LotShape       1460 non-null   object
 6   LandContour    1460 non-null   object
 7   Utilities      1460 non-null   object
 8   LotConfig      1460 non-null   object
 9   LandSlope      1460 non-null   object
 10  Neighborhood   1460 non-null   object
 11  Condition1     1460 non-null   object
 12  Condition2     1460 non-null   object
 13  BldgType       1460 non-null   object
 14  HouseStyle     1460 non-null   object
 15  OverallQual    1460 non-null   int64
 16  OverallCond    1460 non-null   int64
 17  YearBuilt      1460 non-null   int64
 18  YearRemodAdd   1460 non-null   int64
 19  RoofStyle      1460 non-null   object
 20  RoofMatl       1460 non-null   object
 21  Exterior1st    1460 non-null   object
 22  Exterior2nd    1460 non-null   object
 23  MasVnrArea     1460 non-null   float64
 24  ExterQual      1460 non-null   object
 25  ExterCond      1460 non-null   object
 26  Foundation     1460 non-null   object
 27  BsmtQual       1460 non-null   object
 28  BsmtCond       1460 non-null   object
 29  BsmtExposure   1460 non-null   object
 30  BsmtFinType1   1460 non-null   object
 31  BsmtFinSF1     1460 non-null   int64
 32  BsmtFinType2   1460 non-null   object
 33  BsmtFinSF2     1460 non-null   int64
 34  BsmtUnfSF      1460 non-null   int64
 35  TotalBsmtSF    1460 non-null   int64
 36  Heating        1460 non-null   object
 37  HeatingQC      1460 non-null   object
 38  CentralAir     1460 non-null   object
 39  Electrical     1460 non-null   object
 40  1stFlrSF       1460 non-null   int64
```

```
 41  2ndFlrSF       1460 non-null   int64
 42  LowQualFinSF   1460 non-null   int64
 43  GrLivArea      1460 non-null   int64
 44  BsmtFullBath   1460 non-null   int64
 45  BsmtHalfBath   1460 non-null   int64
 46  FullBath       1460 non-null   int64
 47  HalfBath       1460 non-null   int64
 48  BedroomAbvGr   1460 non-null   int64
 49  KitchenAbvGr   1460 non-null   int64
 50  KitchenQual    1460 non-null   object
 51  TotRmsAbvGrd   1460 non-null   int64
 52  Functional     1460 non-null   object
 53  Fireplaces     1460 non-null   int64
 54  FireplaceQu    1460 non-null   object
 55  GarageType     1460 non-null   object
 56  GarageYrBlt    1460 non-null   float64
 57  GarageFinish   1460 non-null   object
 58  GarageCars     1460 non-null   int64
 59  GarageArea     1460 non-null   int64
 60  GarageQual     1460 non-null   object
 61  GarageCond     1460 non-null   object
 62  PavedDrive     1460 non-null   object
 63  WoodDeckSF     1460 non-null   int64
 64  OpenPorchSF    1460 non-null   int64
 65  EnclosedPorch  1460 non-null   int64
 66  3SsnPorch      1460 non-null   int64
 67  ScreenPorch    1460 non-null   int64
 68  PoolArea       1460 non-null   int64
 69  MiscVal        1460 non-null   int64
 70  MoSold         1460 non-null   int64
 71  YrSold         1460 non-null   int64
 72  SaleType       1460 non-null   object
 73  SaleCondition  1460 non-null   object
 74  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(34), object(38)
memory usage: 855.6+ KB
```

Da har vi blitt kvitt alle null-verdier trainingData. Nå må vi gjøre det samme for testData

[11]:
```
testData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1459 non-null   int64
 1   MSSubClass     1459 non-null   int64
 2   MSZoning       1455 non-null   object
```

```
3   LotFrontage    1232 non-null   float64
4   LotArea        1459 non-null   int64
5   Street         1459 non-null   object
6   Alley          107 non-null    object
7   LotShape       1459 non-null   object
8   LandContour    1459 non-null   object
9   Utilities      1457 non-null   object
10  LotConfig      1459 non-null   object
11  LandSlope      1459 non-null   object
12  Neighborhood   1459 non-null   object
13  Condition1     1459 non-null   object
14  Condition2     1459 non-null   object
15  BldgType       1459 non-null   object
16  HouseStyle     1459 non-null   object
17  OverallQual    1459 non-null   int64
18  OverallCond    1459 non-null   int64
19  YearBuilt      1459 non-null   int64
20  YearRemodAdd   1459 non-null   int64
21  RoofStyle      1459 non-null   object
22  RoofMatl       1459 non-null   object
23  Exterior1st    1458 non-null   object
24  Exterior2nd    1458 non-null   object
25  MasVnrType     565 non-null    object
26  MasVnrArea     1444 non-null   float64
27  ExterQual      1459 non-null   object
28  ExterCond      1459 non-null   object
29  Foundation     1459 non-null   object
30  BsmtQual       1415 non-null   object
31  BsmtCond       1414 non-null   object
32  BsmtExposure   1415 non-null   object
33  BsmtFinType1   1417 non-null   object
34  BsmtFinSF1     1458 non-null   float64
35  BsmtFinType2   1417 non-null   object
36  BsmtFinSF2     1458 non-null   float64
37  BsmtUnfSF      1458 non-null   float64
38  TotalBsmtSF    1458 non-null   float64
39  Heating        1459 non-null   object
40  HeatingQC      1459 non-null   object
41  CentralAir     1459 non-null   object
42  Electrical     1459 non-null   object
43  1stFlrSF       1459 non-null   int64
44  2ndFlrSF       1459 non-null   int64
45  LowQualFinSF   1459 non-null   int64
46  GrLivArea      1459 non-null   int64
47  BsmtFullBath   1457 non-null   float64
48  BsmtHalfBath   1457 non-null   float64
49  FullBath       1459 non-null   int64
50  HalfBath       1459 non-null   int64
```

```
 51  BedroomAbvGr   1459 non-null   int64
 52  KitchenAbvGr   1459 non-null   int64
 53  KitchenQual    1458 non-null   object
 54  TotRmsAbvGrd   1459 non-null   int64
 55  Functional     1457 non-null   object
 56  Fireplaces     1459 non-null   int64
 57  FireplaceQu    729 non-null    object
 58  GarageType     1383 non-null   object
 59  GarageYrBlt    1381 non-null   float64
 60  GarageFinish   1381 non-null   object
 61  GarageCars     1458 non-null   float64
 62  GarageArea     1458 non-null   float64
 63  GarageQual     1381 non-null   object
 64  GarageCond     1381 non-null   object
 65  PavedDrive     1459 non-null   object
 66  WoodDeckSF     1459 non-null   int64
 67  OpenPorchSF    1459 non-null   int64
 68  EnclosedPorch  1459 non-null   int64
 69  3SsnPorch      1459 non-null   int64
 70  ScreenPorch    1459 non-null   int64
 71  PoolArea       1459 non-null   int64
 72  PoolQC         3 non-null      object
 73  Fence          290 non-null    object
 74  MiscFeature    51 non-null     object
 75  MiscVal        1459 non-null   int64
 76  MoSold         1459 non-null   int64
 77  YrSold         1459 non-null   int64
 78  SaleType       1458 non-null   object
 79  SaleCondition  1459 non-null   object
dtypes: float64(11), int64(26), object(43)
memory usage: 912.0+ KB
```

[12]:
```
testData.drop(['Alley', 'PoolQC', 'Fence',␣
 ↪'MiscFeature','MasVnrType'],axis=1,inplace=True)
testData.drop(['Id'],axis=1,inplace=True)
```

[13]:
```
for column in testData.columns:
    if testData[column].dtype in ['int64', 'float64']:
        column_mean = testData[column].mean()
        testData[column].fillna(column_mean, inplace=True)
```

[14]:
```
for column in testData.columns:
    if testData[column].dtype == 'object':
        testData[column].fillna(testData[column].mode()[0], inplace=True)
```

[15]:
```
testData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 74 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MSSubClass     1459 non-null   int64
 1   MSZoning       1459 non-null   object
 2   LotFrontage    1459 non-null   float64
 3   LotArea        1459 non-null   int64
 4   Street         1459 non-null   object
 5   LotShape       1459 non-null   object
 6   LandContour    1459 non-null   object
 7   Utilities      1459 non-null   object
 8   LotConfig      1459 non-null   object
 9   LandSlope      1459 non-null   object
 10  Neighborhood   1459 non-null   object
 11  Condition1     1459 non-null   object
 12  Condition2     1459 non-null   object
 13  BldgType       1459 non-null   object
 14  HouseStyle     1459 non-null   object
 15  OverallQual    1459 non-null   int64
 16  OverallCond    1459 non-null   int64
 17  YearBuilt      1459 non-null   int64
 18  YearRemodAdd   1459 non-null   int64
 19  RoofStyle      1459 non-null   object
 20  RoofMatl       1459 non-null   object
 21  Exterior1st    1459 non-null   object
 22  Exterior2nd    1459 non-null   object
 23  MasVnrArea     1459 non-null   float64
 24  ExterQual      1459 non-null   object
 25  ExterCond      1459 non-null   object
 26  Foundation     1459 non-null   object
 27  BsmtQual       1459 non-null   object
 28  BsmtCond       1459 non-null   object
 29  BsmtExposure   1459 non-null   object
 30  BsmtFinType1   1459 non-null   object
 31  BsmtFinSF1     1459 non-null   float64
 32  BsmtFinType2   1459 non-null   object
 33  BsmtFinSF2     1459 non-null   float64
 34  BsmtUnfSF      1459 non-null   float64
 35  TotalBsmtSF    1459 non-null   float64
 36  Heating        1459 non-null   object
 37  HeatingQC      1459 non-null   object
 38  CentralAir     1459 non-null   object
 39  Electrical     1459 non-null   object
 40  1stFlrSF       1459 non-null   int64
 41  2ndFlrSF       1459 non-null   int64
 42  LowQualFinSF   1459 non-null   int64
 43  GrLivArea      1459 non-null   int64
```

```
 44  BsmtFullBath   1459 non-null   float64
 45  BsmtHalfBath   1459 non-null   float64
 46  FullBath       1459 non-null   int64
 47  HalfBath       1459 non-null   int64
 48  BedroomAbvGr   1459 non-null   int64
 49  KitchenAbvGr   1459 non-null   int64
 50  KitchenQual    1459 non-null   object
 51  TotRmsAbvGrd   1459 non-null   int64
 52  Functional     1459 non-null   object
 53  Fireplaces     1459 non-null   int64
 54  FireplaceQu    1459 non-null   object
 55  GarageType     1459 non-null   object
 56  GarageYrBlt    1459 non-null   float64
 57  GarageFinish   1459 non-null   object
 58  GarageCars     1459 non-null   float64
 59  GarageArea     1459 non-null   float64
 60  GarageQual     1459 non-null   object
 61  GarageCond     1459 non-null   object
 62  PavedDrive     1459 non-null   object
 63  WoodDeckSF     1459 non-null   int64
 64  OpenPorchSF    1459 non-null   int64
 65  EnclosedPorch  1459 non-null   int64
 66  3SsnPorch      1459 non-null   int64
 67  ScreenPorch    1459 non-null   int64
 68  PoolArea       1459 non-null   int64
 69  MiscVal        1459 non-null   int64
 70  MoSold         1459 non-null   int64
 71  YrSold         1459 non-null   int64
 72  SaleType       1459 non-null   object
 73  SaleCondition  1459 non-null   object
dtypes: float64(11), int64(25), object(38)
memory usage: 843.6+ KB
```

Vi må håndtere alle kategoriske verdier. Dette er fordi maskinlærings modeller bruker ofte numeriske verdier. Lager så en metode som gjør dette med alle kolonnene som har kategoriske verdier.

[16]:
```
def dummyGen(multcolumns):
    df_final=df
    i=0
    for fields in multcolumns:


        df1=pd.get_dummies(df[fields],drop_first=True)

        df.drop([fields],axis=1,inplace=True)
        if i==0:
            df_final=df1.copy()
```

```
        else:

            df_final=pd.concat([df_final,df1],axis=1)
        i=i+1


    df_final=pd.concat([df,df_final],axis=1)

    return df_final
```

[17]:
```
kolonner = trainingData.select_dtypes(include=['object', 'category']).columns.
↪tolist()
```

[17]:
```
38
```

[18]:
```
testData.shape
```

[18]:
```
(1459, 74)
```

Vi vil unngå at test og data har ulike antall kategorier, det vil si at ikke alle kategorier av en type kommer opp. Det er viktig at test og training data har like verdier slik at modellen blir trent best mulig opp. Bruker da en concat å lage en ny tabell der trening og test data deler antall kolonner.

[19]:
```
df=pd.concat([trainingData, testData],axis=0)
```

[20]:
```
df=dummyGen(kolonner)
```

```
MSZoning
Street
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtExposure
```

```
BsmtFinType1
BsmtFinType2
Heating
HeatingQC
CentralAir
Electrical
KitchenQual
Functional
FireplaceQu
GarageType
GarageFinish
GarageQual
GarageCond
PavedDrive
SaleType
SaleCondition
```

Fjerner duplikater

[21]:
```
df = df.loc[:,~df.columns.duplicated()]
df.shape
```

[21]:
```
(2919, 176)
```

[22]:
```
df_Train=df.iloc[:1460,:]
df_Test=df.iloc[1460:,:]
df_Test.shape
```

[22]:
```
(1459, 176)
```

[23]:
```
df_Test.drop(['SalePrice'],axis=1,inplace=True)
```

```
/var/folders/xb/t_8qjfxd7wj4vhqfgglmrmy00000gn/T/ipykernel_23979/3985304647.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Test.drop(['SalePrice'],axis=1,inplace=True)
```

Oppsummring: Da har vi ryddet i både training og test settet. fjernet NaN-verdier,laget dummies av kategoriske verdier. Vi har brukt concat for å få samme antall kategorier i test og training. Og så splittet det opp igjen.

Nå gjenstår det å velge algorithme og modell. Så kan gi den df_Train, trene den og så utføre modellen på df_Test data og få predikasjoner. Jeg velger å bruke XGboost.

Splitter opp df_train i X og y.

[24]:
```
X_train = df_Train.drop(['SalePrice'],axis=1)
y_train = df_Train['SalePrice']
```

[25]:
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in
/Users/simonknutsson/anaconda3/lib/python3.11/site-packages (2.0.2)
Requirement already satisfied: numpy in
/Users/simonknutsson/anaconda3/lib/python3.11/site-packages (from xgboost)
(1.24.3)
Requirement already satisfied: scipy in
/Users/simonknutsson/anaconda3/lib/python3.11/site-packages (from xgboost)
(1.11.1)
```

Som sagt har jeg tenkt til å bruke XGBboost algoritmen. For å optimalisere algoritmen vil gjøre jeg litt hyperparameter tuning. Det vil si å finne dem parameterne som retunerer det beste resultatet.

[26]:
```
import xgboost
regressor=xgboost.XGBRegressor()
```

[ ]:
```
```

[27]:
```
from scipy.stats import randint, uniform

# Definerer hvilke parametere vi vil endre og gjøre bedre.
hyperparameter_grid = {
    'n_estimators': randint(100, 1500),
    'max_depth': randint(2, 15),
    'learning_rate': uniform(0.05, 0.2),
    'subsample': uniform(0.5, 0.5),
    'min_child_weight': randint(1, 4),
    'gamma': uniform(0, 1)
}
```

[28]:
```
random_cv = RandomizedSearchCV(estimator=regressor,
            param_distributions=hyperparameter_grid,
            cv=5, n_iter=50,
            scoring = 'neg_mean_absolute_error',n_jobs = 4,
            verbose = 5,
            return_train_score = True,
            random_state=42)
```

[29]:
```
random_cv.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
```

[29]:
```
RandomizedSearchCV(cv=5,
                   estimator=XGBRegressor(base_score=None, booster=None,
                                          callbacks=None,
```

```
                                              colsample_bylevel=None,
                                              colsample_bynode=None,
                                              colsample_bytree=None, device=None,
                                              early_stopping_rounds=None,
                                              enable_categorical=False,
                                              eval_metric=None, feature_types=None,
                                              gamma=None, grow_policy=None,
                                              importance_type=None,
                                              interaction_constraints=None,
                                              learning_rate=…
                                             'min_child_weight':
<scipy.stats._distn_infrastructure.rv_discrete_frozen object at 0x148ed8610>,
                                             'n_estimators':
<scipy.stats._distn_infrastructure.rv_discrete_frozen object at 0x148ecab10>,
                                             'subsample':
<scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x148ed8110>},
                   random_state=42, return_train_score=True,
                   scoring='neg_mean_absolute_error', verbose=5)
```

random_cv blir trent med X_train og y_train. Så retunerer den en model med de beste estimatorene. Den lagrer vi variabelen best_model

[30]:
```
best_model = random_cv.best_estimator_
```

Så trener vi den modelen med X_train og y_train

[31]:
```
best_model.fit(X_train,y_train)
```

[31]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=0.009197051616629648, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.07029430857320643,
             max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=4, max_leaves=None,
             min_child_weight=1, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=1013, n_jobs=None,
             num_parallel_tree=None, random_state=None, …)
```

Så kjører vi listen med modellen vi har trent og får resultatene

[32]:
```
y_pred = best_model.predict(df_Test)
```

[33]:
```
y_pred
```

[33]:
```
array([120716.016, 158398.17 , 191163.56 , …, 180101.25 , 115614.336,
       233217.98 ], dtype=float32)
```

Så oppretter vi en csv fil med Id nr og SalePrice siden det er den vi har trent modellen til å retunere.

[34]:
```
pred = pd.DataFrame(y_pred)
sub_df=pd.read_csv('sample_submission.csv')
datasets=pd.concat([sub_df['Id'], pred],axis=1)
datasets.columns=['Id', 'SalePrice']
datasets.to_csv('sample_submission.csv',index=False)
```

```
[CV 1/5] END gamma=0.3745401188473625, learning_rate=0.24014286128198326,
max_depth=12, min_child_weight=1, n_estimators=1144,
subsample=0.5780093202212182;, score=(train=-8.251, test=-17206.340) total time=
3.3s
[CV 3/5] END gamma=0.15599452033620265, learning_rate=0.061616722433639894,
max_depth=9, min_child_weight=1, n_estimators=1223,
subsample=0.5714334089609704;, score=(train=-0.171, test=-16572.982) total time=
5.0s
[CV 2/5] END gamma=0.6508884729488529, learning_rate=0.061282315805420054,
max_depth=9, min_child_weight=2, n_estimators=905,
subsample=0.5003893829205072;, score=(train=-81.072, test=-17704.979) total
time=   2.4s
[CV 1/5] END gamma=0.9922115592912175, learning_rate=0.17349630192554333,
max_depth=11, min_child_weight=2, n_estimators=352,
subsample=0.7159725093210578;, score=(train=-9.641, test=-16674.725) total time=
1.3s
[CV 2/5] END gamma=0.9922115592912175, learning_rate=0.17349630192554333,
max_depth=11, min_child_weight=2, n_estimators=352,
subsample=0.7159725093210578;, score=(train=-10.293, test=-18502.063) total
time=   1.3s
[CV 1/5] END gamma=0.2912291401980419, learning_rate=0.17237057894447588,
max_depth=11, min_child_weight=3, n_estimators=289,
subsample=0.5453032172664104;, score=(train=-107.979, test=-18157.299) total
time=   0.8s
[CV 4/5] END gamma=0.2912291401980419, learning_rate=0.17237057894447588,
max_depth=11, min_child_weight=3, n_estimators=289,
subsample=0.5453032172664104;, score=(train=-80.226, test=-15492.485) total
time=   0.8s
[CV 3/5] END gamma=0.6183860093330873, learning_rate=0.12649239825343256,
max_depth=5, min_child_weight=1, n_estimators=1254,
subsample=0.9299702033681603;, score=(train=-4.480, test=-17228.134) total time=
1.7s
[CV 2/5] END gamma=0.6803075385877797, learning_rate=0.1400998503939086,
max_depth=3, min_child_weight=1, n_estimators=415,
subsample=0.7816441089227697;, score=(train=-4035.259, test=-17341.672) total
time=   0.3s
[CV 5/5] END gamma=0.6803075385877797, learning_rate=0.1400998503939086,
max_depth=3, min_child_weight=1, n_estimators=415,
subsample=0.7816441089227697;, score=(train=-4017.569, test=-16149.804) total
time=   0.3s
[CV 3/5] END gamma=0.3854165025399161, learning_rate=0.05319325044404284,
max_depth=3, min_child_weight=3, n_estimators=1055,
```

```
subsample=0.5610191174223894;, score=(train=-4355.286, test=-15818.615) total
time=   0.7s
[CV 1/5] END gamma=0.4951769101112702, learning_rate=0.05687770422304368,
max_depth=2, min_child_weight=2, n_estimators=1125,
subsample=0.7125779372456224;, score=(train=-7541.582, test=-15165.893) total
time=   0.6s
[CV 4/5] END gamma=0.4951769101112702, learning_rate=0.05687770422304368,
max_depth=2, min_child_weight=2, n_estimators=1125,
subsample=0.7125779372456224;, score=(train=-7851.267, test=-13223.467) total
time=   0.6s
[CV 3/5] END gamma=0.20794166286818883, learning_rate=0.1635400655639983,
max_depth=14, min_child_weight=3, n_estimators=501,
subsample=0.8875664116805573;, score=(train=-0.309, test=-17868.612) total time=
1.8s
[CV 3/5] END gamma=0.9394989415641891, learning_rate=0.22896547008552975,
max_depth=9, min_child_weight=2, n_estimators=1375,
subsample=0.522613644455269;, score=(train=-0.282, test=-18248.251) total time=
1.9s
[CV 5/5] END gamma=0.9394989415641891, learning_rate=0.22896547008552975,
max_depth=9, min_child_weight=2, n_estimators=1375,
subsample=0.522613644455269;, score=(train=-0.270, test=-19045.187) total time=
1.7s
[CV 1/5] END gamma=0.965255307264138, learning_rate=0.17140684953733692,
max_depth=14, min_child_weight=1, n_estimators=1280,
subsample=0.9010984903770198;, score=(train=-7.642, test=-18918.269) total time=
2.4s
[CV 5/5] END gamma=0.965255307264138, learning_rate=0.17140684953733692,
max_depth=14, min_child_weight=1, n_estimators=1280,
subsample=0.9010984903770198;, score=(train=-0.366, test=-19910.246) total time=
1.9s
[CV 4/5] END gamma=0.07455064367977082, learning_rate=0.24737738732010345,
max_depth=9, min_child_weight=1, n_estimators=747,
subsample=0.5027610585618012;, score=(train=-8.478, test=-16916.420) total time=
2.4s
[CV 1/5] END gamma=0.926300878513349, learning_rate=0.18021540510038891,
max_depth=13, min_child_weight=3, n_estimators=300,
subsample=0.811649063413779;, score=(train=-17.424, test=-18528.755) total time=
1.0s
[CV 5/5] END gamma=0.926300878513349, learning_rate=0.18021540510038891,
max_depth=13, min_child_weight=3, n_estimators=300,
subsample=0.811649063413779;, score=(train=-9.626, test=-19023.190) total time=
1.1s
[CV 4/5] END gamma=0.3308980248526492, learning_rate=0.06271167005720474,
max_depth=8, min_child_weight=2, n_estimators=904, subsample=0.864803089169032;,
score=(train=-10.213, test=-15028.636) total time=   3.0s
[CV 5/5] END gamma=0.6375574713552131, learning_rate=0.2274425485152653,
max_depth=2, min_child_weight=3, n_estimators=712,
subsample=0.9858560476945519;, score=(train=-4496.686, test=-18655.218) total
```

```
time=   0.4s
[CV 3/5] END gamma=0.8489138242660839, learning_rate=0.19434590423297465,
max_depth=8, min_child_weight=3, n_estimators=497,
subsample=0.7468977981821954;, score=(train=-1.002, test=-18537.234) total time=
1.3s
[CV 2/5] END gamma=0.5227328293819941, learning_rate=0.13550820367170993,
max_depth=11, min_child_weight=1, n_estimators=963,
subsample=0.8182052056318903;, score=(train=-7.565, test=-17449.671) total time=
2.7s
[CV 1/5] END gamma=0.3143559810763267, learning_rate=0.15171413823294055,
max_depth=6, min_child_weight=3, n_estimators=330,
subsample=0.7051914615178149;, score=(train=-274.142, test=-15740.528) total
time=   0.7s
[CV 3/5] END gamma=0.3143559810763267, learning_rate=0.15171413823294055,
max_depth=6, min_child_weight=3, n_estimators=330,
subsample=0.7051914615178149;, score=(train=-259.507, test=-17858.372) total
time=   0.6s
[CV 1/5] END gamma=0.7555511385430487, learning_rate=0.0957596330983245,
max_depth=8, min_child_weight=3, n_estimators=1366,
subsample=0.5806106436270022;, score=(train=-8.702, test=-16033.333) total time=
3.0s
[CV 5/5] END gamma=0.7555511385430487, learning_rate=0.0957596330983245,
max_depth=8, min_child_weight=3, n_estimators=1366,
subsample=0.5806106436270022;, score=(train=-0.721, test=-17656.152) total time=
3.3s
[CV 2/5] END gamma=0.2184404372168336, learning_rate=0.13330198957407324,
max_depth=12, min_child_weight=2, n_estimators=1471,
subsample=0.6781489190384875;, score=(train=-7.988, test=-18155.612) total time=
3.7s
[CV 2/5] END gamma=0.906828441545754, learning_rate=0.10442644987692706,
max_depth=12, min_child_weight=2, n_estimators=732,
subsample=0.9090073829612466;, score=(train=-7.523, test=-18486.229) total time=
2.6s
[CV 1/5] END gamma=0.8607305832563434, learning_rate=0.051390426106238146,
max_depth=9, min_child_weight=3, n_estimators=748,
subsample=0.7424149856794916;, score=(train=-104.696, test=-16393.839) total
time=   2.1s
[CV 5/5] END gamma=0.8607305832563434, learning_rate=0.051390426106238146,
max_depth=9, min_child_weight=3, n_estimators=748,
subsample=0.7424149856794916;, score=(train=-221.081, test=-17075.460) total
time=   1.8s
[CV 4/5] END gamma=0.6924360328902703, learning_rate=0.1038824667597043,
max_depth=9, min_child_weight=3, n_estimators=1497,
subsample=0.6093821097865351;, score=(train=-8.206, test=-14874.525) total time=
3.3s
[CV 2/5] END gamma=0.5581020020173412, learning_rate=0.13076723421160819,
max_depth=2, min_child_weight=2, n_estimators=663,
subsample=0.6234380314193007;, score=(train=-6983.491, test=-17704.117) total
```

```
time=   0.4s
[CV 4/5] END gamma=0.5581020020173412, learning_rate=0.13076723421160819,
max_depth=2, min_child_weight=2, n_estimators=663,
subsample=0.6234380314193007;, score=(train=-7076.117, test=-14197.712) total
time=   0.4s
[CV 3/5] END gamma=0.6963042728397884, learning_rate=0.19245411798488843,
max_depth=6, min_child_weight=1, n_estimators=1051,
subsample=0.5184434736772664;, score=(train=-0.271, test=-18878.553) total time=
2.0s
[CV 1/5] END gamma=0.6095643339798968, learning_rate=0.15053580464577232,
max_depth=7, min_child_weight=3, n_estimators=1348,
subsample=0.8171756723506819;, score=(train=-8.379, test=-16058.925) total time=
2.4s
[CV 5/5] END gamma=0.6095643339798968, learning_rate=0.15053580464577232,
max_depth=7, min_child_weight=3, n_estimators=1348,
subsample=0.8171756723506819;, score=(train=-0.280, test=-17189.052) total time=
2.1s
[CV 5/5] END gamma=0.6807054515547668, learning_rate=0.15618691666342727,
max_depth=11, min_child_weight=2, n_estimators=346,
subsample=0.8360677737029393;, score=(train=-0.668, test=-18589.343) total time=
1.5s
[CV 4/5] END gamma=0.7616196153287176, learning_rate=0.09752750879847993,
max_depth=7, min_child_weight=1, n_estimators=983,
subsample=0.9917115704474215;, score=(train=-8.084, test=-14823.999) total time=
2.8s
[CV 3/5] END gamma=0.39882444244455306, learning_rate=0.21328637464387679,
max_depth=2, min_child_weight=1, n_estimators=1247,
subsample=0.7540993883703593;, score=(train=-2723.165, test=-16871.864) total
time=   0.8s
[CV 1/5] END gamma=0.6958128067908819, learning_rate=0.221671760962744,
max_depth=13, min_child_weight=3, n_estimators=502,
subsample=0.855574766219009;, score=(train=-7.767, test=-18240.581) total time=
1.6s
[CV 5/5] END gamma=0.6958128067908819, learning_rate=0.221671760962744,
max_depth=13, min_child_weight=3, n_estimators=502,
subsample=0.855574766219009;, score=(train=-0.300, test=-19053.487) total time=
1.6s
[CV 4/5] END gamma=0.8095010461397154, learning_rate=0.11973319745834587,
max_depth=8, min_child_weight=1, n_estimators=1028,
subsample=0.6987860105437611;, score=(train=-8.208, test=-14674.580) total time=
3.3s
[CV 3/5] END gamma=0.5177513505274801, learning_rate=0.2175420211814656,
max_depth=12, min_child_weight=1, n_estimators=572,
subsample=0.6045358103688568;, score=(train=-0.222, test=-18464.606) total time=
1.8s
[CV 1/5] END gamma=0.5414479738275658, learning_rate=0.18915687986901647,
max_depth=8, min_child_weight=2, n_estimators=723,
subsample=0.7776004057997312;, score=(train=-7.818, test=-17652.267) total time=
```

```
2.0s
[CV 4/5] END gamma=0.5414479738275658, learning_rate=0.18915687986901647,
max_depth=8, min_child_weight=2, n_estimators=723,
subsample=0.7776004057997312;, score=(train=-7.783, test=-15660.302) total time=
2.0s
[CV 4/5] END gamma=0.5296505783560065, learning_rate=0.09837045818009034,
max_depth=13, min_child_weight=1, n_estimators=1160,
subsample=0.9413181715946699;, score=(train=-7.411, test=-16540.738) total time=
3.6s
[CV 3/5] END gamma=0.18870710834137938, learning_rate=0.10577427051843638,
max_depth=4, min_child_weight=3, n_estimators=503,
subsample=0.9435432121325587;, score=(train=-1832.470, test=-17156.057) total
time=   0.6s
[CV 1/5] END gamma=0.7798755458576239, learning_rate=0.17840632923085759,
max_depth=12, min_child_weight=1, n_estimators=203,
subsample=0.803214529829795;, score=(train=-8.432, test=-17036.802) total time=
1.6s
[CV 5/5] END gamma=0.7798755458576239, learning_rate=0.17840632923085759,
max_depth=12, min_child_weight=1, n_estimators=203,
subsample=0.803214529829795;, score=(train=-0.352, test=-18282.833) total time=
1.5s
[CV 4/5] END gamma=0.009197051616629648, learning_rate=0.07029430857320643,
max_depth=4, min_child_weight=1, n_estimators=1013,
subsample=0.5804040257087493;, score=(train=-1139.123, test=-13647.389) total
time=   1.3s
[CV 3/5] END gamma=0.5487337893665861, learning_rate=0.18837903953853868,
max_depth=14, min_child_weight=1, n_estimators=260,
subsample=0.5090376818077604;, score=(train=-3.055, test=-18557.514) total time=
1.7s
[CV 2/5] END gamma=0.4938937151834346, learning_rate=0.08576454184426577,
max_depth=9, min_child_weight=3, n_estimators=1318,
subsample=0.8288064461501716;, score=(train=-7.667, test=-17253.853) total time=
3.8s
[CV 2/5] END gamma=0.5683086033354716, learning_rate=0.0687349535656185,
max_depth=4, min_child_weight=2, n_estimators=1498,
subsample=0.6252309093027921;, score=(train=-484.948, test=-16473.002) total
time=   1.8s
[CV 4/5] END gamma=0.5683086033354716, learning_rate=0.0687349535656185,
max_depth=4, min_child_weight=2, n_estimators=1498,
subsample=0.6252309093027921;, score=(train=-484.350, test=-13912.506) total
time=   1.7s
[CV 3/5] END gamma=0.5898708475605439, learning_rate=0.24577857165500183,
max_depth=4, min_child_weight=2, n_estimators=1406,
subsample=0.7171971827552144;, score=(train=-2.920, test=-18455.533) total time=
1.7s
[CV 3/5] END gamma=0.3500784076946757, learning_rate=0.17902067240611297,
max_depth=2, min_child_weight=1, n_estimators=1119,
subsample=0.8612260576307527;, score=(train=-3416.769, test=-16316.748) total
```

```
time=   0.7s
[CV 2/5] END gamma=0.2807723624408558, learning_rate=0.05486319328629077,
max_depth=2, min_child_weight=1, n_estimators=1295,
subsample=0.9702292921764571;, score=(train=-7508.306, test=-16868.996) total
time=   0.8s
[CV 1/5] END gamma=0.9539285770025874, learning_rate=0.23297287804408973,
max_depth=4, min_child_weight=1, n_estimators=1160,
subsample=0.9641592812938626;, score=(train=-22.714, test=-16133.214) total
time=   1.4s
[CV 5/5] END gamma=0.9539285770025874, learning_rate=0.23297287804408973,
max_depth=4, min_child_weight=1, n_estimators=1160,
subsample=0.9641592812938626;, score=(train=-7.199, test=-17761.710) total time=
1.4s
[CV 4/5] END gamma=0.42818414831731433, learning_rate=0.24333096380873392,
max_depth=4, min_child_weight=1, n_estimators=1239,
subsample=0.6472244460347929;, score=(train=-36.771, test=-15414.345) total
time=   1.5s
[CV 5/5] END gamma=0.38509772860192526, learning_rate=0.22022733430337138,
max_depth=4, min_child_weight=2, n_estimators=665,
subsample=0.8480148983374864;, score=(train=-145.075, test=-17521.729) total
time=   0.7s
[CV 3/5] END gamma=0.3745401188473625, learning_rate=0.24014286128198326,
max_depth=12, min_child_weight=1, n_estimators=1144,
subsample=0.5780093202212182;, score=(train=-0.177, test=-19229.299) total time=
2.0s
[CV 5/5] END gamma=0.3745401188473625, learning_rate=0.24014286128198326,
max_depth=12, min_child_weight=1, n_estimators=1144,
subsample=0.5780093202212182;, score=(train=-0.173, test=-18574.294) total time=
1.7s
[CV 4/5] END gamma=0.15599452033620265, learning_rate=0.061616722433639894,
max_depth=9, min_child_weight=1, n_estimators=1223,
subsample=0.5714334089609704;, score=(train=-8.440, test=-13681.908) total time=
4.8s
[CV 3/5] END gamma=0.6508884729488529, learning_rate=0.061282315805420054,
max_depth=9, min_child_weight=2, n_estimators=905,
subsample=0.5003893829205072;, score=(train=-76.600, test=-16401.367) total
time=   2.2s
[CV 5/5] END gamma=0.6508884729488529, learning_rate=0.061282315805420054,
max_depth=9, min_child_weight=2, n_estimators=905,
subsample=0.5003893829205072;, score=(train=-81.208, test=-16784.974) total
time=   2.4s
[CV 5/5] END gamma=0.9922115592912175, learning_rate=0.17349630192554333,
max_depth=11, min_child_weight=2, n_estimators=352,
subsample=0.7159725093210578;, score=(train=-1.254, test=-19300.795) total time=
1.2s
[CV 5/5] END gamma=0.2912291401980419, learning_rate=0.17237057894447588,
max_depth=11, min_child_weight=3, n_estimators=289,
subsample=0.5453032172664104;, score=(train=-113.658, test=-18606.301) total
```

```
time=   0.7s
[CV 4/5] END gamma=0.6183860093330873, learning_rate=0.12649239825343256,
max_depth=5, min_child_weight=1, n_estimators=1254,
subsample=0.9299702033681603;, score=(train=-16.805, test=-14868.344) total
time=   1.7s
[CV 3/5] END gamma=0.6803075385877797, learning_rate=0.1400998503939086,
max_depth=3, min_child_weight=1, n_estimators=415,
subsample=0.7816441089227697;, score=(train=-3860.242, test=-17108.439) total
time=   0.3s
[CV 1/5] END gamma=0.3854165025399161, learning_rate=0.05319325044404284,
max_depth=3, min_child_weight=3, n_estimators=1055,
subsample=0.5610191174223894;, score=(train=-4648.438, test=-15167.716) total
time=   0.7s
[CV 4/5] END gamma=0.3854165025399161, learning_rate=0.05319325044404284,
max_depth=3, min_child_weight=3, n_estimators=1055,
subsample=0.5610191174223894;, score=(train=-4772.729, test=-13173.534) total
time=   0.7s
[CV 3/5] END gamma=0.4951769101112702, learning_rate=0.05687770422304368,
max_depth=2, min_child_weight=2, n_estimators=1125,
subsample=0.7125779372456224;, score=(train=-7329.754, test=-15741.460) total
time=   0.6s
[CV 2/5] END gamma=0.20794166286818883, learning_rate=0.1635400655639983,
max_depth=14, min_child_weight=3, n_estimators=501,
subsample=0.8875664116805573;, score=(train=-7.527, test=-17821.464) total time=
1.8s
[CV 1/5] END gamma=0.9394989415641891, learning_rate=0.22896547008552975,
max_depth=9, min_child_weight=2, n_estimators=1375,
subsample=0.522613644455269;, score=(train=-10.022, test=-18665.805) total time=
2.7s
[CV 1/5] END gamma=0.32533033076326434, learning_rate=0.12773545793789642,
max_depth=3, min_child_weight=3, n_estimators=1176,
subsample=0.7933755828319241;, score=(train=-1154.765, test=-15264.382) total
time=   0.8s
[CV 3/5] END gamma=0.32533033076326434, learning_rate=0.12773545793789642,
max_depth=3, min_child_weight=3, n_estimators=1176,
subsample=0.7933755828319241;, score=(train=-1088.002, test=-17568.772) total
time=   1.0s
[CV 2/5] END gamma=0.965255307264138, learning_rate=0.17140684953733692,
max_depth=14, min_child_weight=1, n_estimators=1280,
subsample=0.9010984903770198;, score=(train=-7.657, test=-18913.546) total time=
2.4s
[CV 2/5] END gamma=0.07455064367977082, learning_rate=0.24737738732010345,
max_depth=9, min_child_weight=1, n_estimators=747,
subsample=0.5027610585618012;, score=(train=-9.352, test=-19186.417) total time=
2.4s
[CV 2/5] END gamma=0.8154614284548342, learning_rate=0.19137146876952343,
max_depth=4, min_child_weight=3, n_estimators=388,
subsample=0.8029799873905057;, score=(train=-1043.022, test=-17136.960) total
```

```
time=   0.6s
[CV 4/5] END gamma=0.8154614284548342, learning_rate=0.19137146876952343,
max_depth=4, min_child_weight=3, n_estimators=388,
subsample=0.8029799873905057;, score=(train=-1097.159, test=-14638.313) total
time=   0.4s
[CV 2/5] END gamma=0.926300878513349, learning_rate=0.18021540510038891,
max_depth=13, min_child_weight=3, n_estimators=300,
subsample=0.811649063413779;, score=(train=-10.734, test=-18533.447) total time=
1.2s
[CV 1/5] END gamma=0.3308980248526492, learning_rate=0.06271167005720474,
max_depth=8, min_child_weight=2, n_estimators=904, subsample=0.864803089169032;,
score=(train=-11.913, test=-16565.598) total time=   2.9s
[CV 5/5] END gamma=0.3308980248526492, learning_rate=0.06271167005720474,
max_depth=8, min_child_weight=2, n_estimators=904, subsample=0.864803089169032;,
score=(train=-3.874, test=-17270.018) total time=   2.7s
[CV 1/5] END gamma=0.5227328293819941, learning_rate=0.13550820367170993,
max_depth=11, min_child_weight=1, n_estimators=963,
subsample=0.8182052056318903;, score=(train=-7.634, test=-18047.574) total time=
2.7s
[CV 5/5] END gamma=0.5227328293819941, learning_rate=0.13550820367170993,
max_depth=11, min_child_weight=1, n_estimators=963,
subsample=0.8182052056318903;, score=(train=-0.264, test=-18341.384) total time=
2.1s
[CV 4/5] END gamma=0.7555511385430487, learning_rate=0.0957596330983245,
max_depth=8, min_child_weight=3, n_estimators=1366,
subsample=0.5806106436270022;, score=(train=-8.516, test=-14671.481) total time=
3.0s
[CV 3/5] END gamma=0.9296976523425731, learning_rate=0.2116240759128834,
max_depth=10, min_child_weight=1, n_estimators=745,
subsample=0.7282672852414551;, score=(train=-0.316, test=-18294.252) total time=
1.5s
[CV 4/5] END gamma=0.9296976523425731, learning_rate=0.2116240759128834,
max_depth=10, min_child_weight=1, n_estimators=745,
subsample=0.7282672852414551;, score=(train=-7.751, test=-15654.346) total time=
2.0s
[CV 4/5] END gamma=0.2184404372168336, learning_rate=0.13330198957407324,
max_depth=12, min_child_weight=2, n_estimators=1471,
subsample=0.6781489190384875;, score=(train=-7.724, test=-14986.406) total time=
3.6s
[CV 3/5] END gamma=0.906828441545754, learning_rate=0.10442644987692706,
max_depth=12, min_child_weight=2, n_estimators=732,
subsample=0.9090073829612466;, score=(train=-0.406, test=-17659.139) total time=
2.5s
[CV 2/5] END gamma=0.8607305832563434, learning_rate=0.051390426106238146,
max_depth=9, min_child_weight=3, n_estimators=748,
subsample=0.7424149856794916;, score=(train=-139.481, test=-17712.274) total
time=   2.0s
[CV 1/5] END gamma=0.6924360328902703, learning_rate=0.1038824667597043,
```

```
max_depth=9, min_child_weight=3, n_estimators=1497,
subsample=0.6093821097865351;, score=(train=-8.404, test=-16542.808) total time=
3.5s
[CV 5/5] END gamma=0.6924360328902703, learning_rate=0.1038824667597043,
max_depth=9, min_child_weight=3, n_estimators=1497,
subsample=0.6093821097865351;, score=(train=-0.283, test=-17685.234) total time=
3.4s
[CV 4/5] END gamma=0.6963042728397884, learning_rate=0.19245411798488843,
max_depth=6, min_child_weight=1, n_estimators=1051,
subsample=0.5184434736772664;, score=(train=-10.208, test=-15383.016) total
time=   2.2s
[CV 3/5] END gamma=0.6095643339798968, learning_rate=0.15053580464577232,
max_depth=7, min_child_weight=3, n_estimators=1348,
subsample=0.8171756723506819;, score=(train=-0.288, test=-17470.886) total time=
2.2s
[CV 2/5] END gamma=0.6807054515547668, learning_rate=0.15618691666342727,
max_depth=11, min_child_weight=2, n_estimators=346,
subsample=0.8360677737029393;, score=(train=-8.717, test=-18870.406) total time=
1.6s
[CV 2/5] END gamma=0.7616196153287176, learning_rate=0.09752750879847993,
max_depth=7, min_child_weight=1, n_estimators=983,
subsample=0.9917115704474215;, score=(train=-8.578, test=-17529.957) total time=
3.0s
[CV 1/5] END gamma=0.39882444244455306, learning_rate=0.21328637464387679,
max_depth=2, min_child_weight=1, n_estimators=1247,
subsample=0.7540993883703593;, score=(train=-2802.375, test=-15717.210) total
time=   0.8s
[CV 4/5] END gamma=0.39882444244455306, learning_rate=0.21328637464387679,
max_depth=2, min_child_weight=1, n_estimators=1247,
subsample=0.7540993883703593;, score=(train=-2893.501, test=-14480.384) total
time=   0.8s
[CV 2/5] END gamma=0.6958128067908819, learning_rate=0.221671760962744,
max_depth=13, min_child_weight=3, n_estimators=502,
subsample=0.855574766219009;, score=(train=-7.647, test=-19481.987) total time=
1.7s
[CV 1/5] END gamma=0.8095010461397154, learning_rate=0.11973319745834587,
max_depth=8, min_child_weight=1, n_estimators=1028,
subsample=0.6987860105437611;, score=(train=-8.272, test=-16324.013) total time=
3.5s
[CV 5/5] END gamma=0.8095010461397154, learning_rate=0.11973319745834587,
max_depth=8, min_child_weight=1, n_estimators=1028,
subsample=0.6987860105437611;, score=(train=-0.314, test=-17847.357) total time=
2.9s
[CV 2/5] END gamma=0.5414479738275658, learning_rate=0.18915687986901647,
max_depth=8, min_child_weight=2, n_estimators=723,
subsample=0.7776004057997312;, score=(train=-8.151, test=-18131.279) total time=
2.1s
[CV 1/5] END gamma=0.5296505783560065, learning_rate=0.09837045818009034,
```

```
max_depth=13, min_child_weight=1, n_estimators=1160,
subsample=0.9413181715946699;, score=(train=-7.432, test=-18299.112) total time=
3.8s
[CV 5/5] END gamma=0.5296505783560065, learning_rate=0.09837045818009034,
max_depth=13, min_child_weight=1, n_estimators=1160,
subsample=0.9413181715946699;, score=(train=-0.319, test=-18612.162) total time=
3.0s
[CV 4/5] END gamma=0.7798755458576239, learning_rate=0.17840632923085759,
max_depth=12, min_child_weight=1, n_estimators=203,
subsample=0.803214529829795;, score=(train=-7.887, test=-15139.534) total time=
1.6s
[CV 3/5] END gamma=0.009197051616629648, learning_rate=0.07029430857320643,
max_depth=4, min_child_weight=1, n_estimators=1013,
subsample=0.5804040257087493;, score=(train=-1102.876, test=-16036.844) total
time=   1.3s
[CV 2/5] END gamma=0.5487337893665861, learning_rate=0.18837903953853868,
max_depth=14, min_child_weight=1, n_estimators=260,
subsample=0.5090376818077604;, score=(train=-13.034, test=-18660.409) total
time=   1.7s
[CV 1/5] END gamma=0.4938937151834346, learning_rate=0.08576454184426577,
max_depth=9, min_child_weight=3, n_estimators=1318,
subsample=0.8288064461501716;, score=(train=-7.552, test=-16183.125) total time=
3.6s
[CV 5/5] END gamma=0.4938937151834346, learning_rate=0.08576454184426577,
max_depth=9, min_child_weight=3, n_estimators=1318,
subsample=0.8288064461501716;, score=(train=-0.269, test=-17922.210) total time=
3.1s
[CV 1/5] END gamma=0.5898708475605439, learning_rate=0.24577857165500183,
max_depth=4, min_child_weight=2, n_estimators=1406,
subsample=0.7171971827552144;, score=(train=-22.588, test=-16983.736) total
time=   1.7s
[CV 5/5] END gamma=0.5898708475605439, learning_rate=0.24577857165500183,
max_depth=4, min_child_weight=2, n_estimators=1406,
subsample=0.7171971827552144;, score=(train=-2.573, test=-18389.611) total time=
1.7s
[CV 1/5] END gamma=0.2807723624408558, learning_rate=0.05486319328629077,
max_depth=2, min_child_weight=1, n_estimators=1295,
subsample=0.9702292921764571;, score=(train=-7486.787, test=-14673.024) total
time=   0.8s
[CV 5/5] END gamma=0.2807723624408558, learning_rate=0.05486319328629077,
max_depth=2, min_child_weight=1, n_estimators=1295,
subsample=0.9702292921764571;, score=(train=-7176.337, test=-16399.461) total
time=   0.8s
[CV 4/5] END gamma=0.9539285770025874, learning_rate=0.23297287804408973,
max_depth=4, min_child_weight=1, n_estimators=1160,
subsample=0.9641592812938626;, score=(train=-22.711, test=-14276.043) total
time=   1.4s
[CV 3/5] END gamma=0.42818414831731433, learning_rate=0.24333096380873392,
```

```
max_depth=4, min_child_weight=1, n_estimators=1239,
subsample=0.6472244460347929;, score=(train=-5.511, test=-18686.280) total time=
1.5s
[CV 2/5] END gamma=0.38509772860192526, learning_rate=0.22022733430337138,
max_depth=4, min_child_weight=2, n_estimators=665,
subsample=0.8480148983374864;, score=(train=-201.585, test=-18035.469) total
time=   0.8s
[CV 4/5] END gamma=0.38509772860192526, learning_rate=0.22022733430337138,
max_depth=4, min_child_weight=2, n_estimators=665,
subsample=0.8480148983374864;, score=(train=-183.737, test=-15151.213) total
time=   0.7s
[CV 4/5] END gamma=0.3745401188473625, learning_rate=0.24014286128198326,
max_depth=12, min_child_weight=1, n_estimators=1144,
subsample=0.5780093202212182;, score=(train=-8.526, test=-16508.045) total time=
3.0s
[CV 1/5] END gamma=0.15599452033620265, learning_rate=0.061616722433639894,
max_depth=9, min_child_weight=1, n_estimators=1223,
subsample=0.5714334089609704;, score=(train=-8.041, test=-16033.350) total time=
4.7s
[CV 1/5] END gamma=0.6508884729488529, learning_rate=0.061282315805420054,
max_depth=9, min_child_weight=2, n_estimators=905,
subsample=0.5003893829205072;, score=(train=-59.775, test=-15802.380) total
time=   2.5s
[CV 4/5] END gamma=0.6508884729488529, learning_rate=0.061282315805420054,
max_depth=9, min_child_weight=2, n_estimators=905,
subsample=0.5003893829205072;, score=(train=-81.554, test=-13678.064) total
time=   2.4s
[CV 4/5] END gamma=0.9922115592912175, learning_rate=0.17349630192554333,
max_depth=11, min_child_weight=2, n_estimators=352,
subsample=0.7159725093210578;, score=(train=-8.473, test=-15488.546) total time=
1.3s
[CV 3/5] END gamma=0.2912291401980419, learning_rate=0.17237057894447588,
max_depth=11, min_child_weight=3, n_estimators=289,
subsample=0.5453032172664104;, score=(train=-165.462, test=-16936.569) total
time=   0.7s
[CV 2/5] END gamma=0.6183860093330873, learning_rate=0.12649239825343256,
max_depth=5, min_child_weight=1, n_estimators=1254,
subsample=0.9299702033681603;, score=(train=-17.466, test=-16652.067) total
time=   1.8s
[CV 1/5] END gamma=0.6803075385877797, learning_rate=0.1400998503939086,
max_depth=3, min_child_weight=1, n_estimators=415,
subsample=0.7816441089227697;, score=(train=-4136.572, test=-15306.927) total
time=   0.3s
[CV 4/5] END gamma=0.6803075385877797, learning_rate=0.1400998503939086,
max_depth=3, min_child_weight=1, n_estimators=415,
subsample=0.7816441089227697;, score=(train=-4198.829, test=-13779.869) total
time=   0.3s
[CV 2/5] END gamma=0.3854165025399161, learning_rate=0.05319325044404284,
```

```
max_depth=3, min_child_weight=3, n_estimators=1055,
subsample=0.5610191174223894;, score=(train=-4665.175, test=-16869.708) total
time=   0.7s
[CV 5/5] END gamma=0.3854165025399161, learning_rate=0.05319325044404284,
max_depth=3, min_child_weight=3, n_estimators=1055,
subsample=0.5610191174223894;, score=(train=-4291.571, test=-17106.684) total
time=   0.7s
[CV 1/5] END gamma=0.20794166286818883, learning_rate=0.1635400655639983,
max_depth=14, min_child_weight=3, n_estimators=501,
subsample=0.8875664116805573;, score=(train=-7.584, test=-17945.894) total time=
1.7s
[CV 5/5] END gamma=0.20794166286818883, learning_rate=0.1635400655639983,
max_depth=14, min_child_weight=3, n_estimators=501,
subsample=0.8875664116805573;, score=(train=-0.236, test=-18819.147) total time=
1.6s
[CV 4/5] END gamma=0.9394989415641891, learning_rate=0.22896547008552975,
max_depth=9, min_child_weight=2, n_estimators=1375,
subsample=0.522613644455269;, score=(train=-10.595, test=-16111.482) total time=
2.5s
[CV 5/5] END gamma=0.32533033076326434, learning_rate=0.12773545793789642,
max_depth=3, min_child_weight=3, n_estimators=1176,
subsample=0.7933755828319241;, score=(train=-1158.502, test=-17572.620) total
time=   1.0s
[CV 3/5] END gamma=0.965255307264138, learning_rate=0.17140684953733692,
max_depth=14, min_child_weight=1, n_estimators=1280,
subsample=0.9010984903770198;, score=(train=-0.357, test=-18516.841) total time=
1.9s
[CV 1/5] END gamma=0.07455064367977082, learning_rate=0.24737738732010345,
max_depth=9, min_child_weight=1, n_estimators=747,
subsample=0.5027610585618012;, score=(train=-10.137, test=-17815.542) total
time=   2.3s
[CV 5/5] END gamma=0.07455064367977082, learning_rate=0.24737738732010345,
max_depth=9, min_child_weight=1, n_estimators=747,
subsample=0.5027610585618012;, score=(train=-0.082, test=-18542.690) total time=
1.8s
[CV 4/5] END gamma=0.926300878513349, learning_rate=0.18021540510038891,
max_depth=13, min_child_weight=3, n_estimators=300,
subsample=0.811649063413779;, score=(train=-10.992, test=-15605.162) total time=
1.1s
[CV 3/5] END gamma=0.3308980248526492, learning_rate=0.06271167005720474,
max_depth=8, min_child_weight=2, n_estimators=904, subsample=0.864803089169032;,
score=(train=-3.475, test=-16521.377) total time=   2.9s
[CV 1/5] END gamma=0.6375574713552131, learning_rate=0.2274425485152653,
max_depth=2, min_child_weight=3, n_estimators=712,
subsample=0.9858560476945519;, score=(train=-4702.805, test=-15605.749) total
time=   0.5s
[CV 3/5] END gamma=0.6375574713552131, learning_rate=0.2274425485152653,
max_depth=2, min_child_weight=3, n_estimators=712,
```

```
subsample=0.9858560476945519;, score=(train=-4540.175, test=-17484.260) total
time=   0.5s
[CV 2/5] END gamma=0.8489138242660839, learning_rate=0.19434590423297465,
max_depth=8, min_child_weight=3, n_estimators=497,
subsample=0.7468977981821954;, score=(train=-10.218, test=-17458.190) total
time=   1.2s
[CV 4/5] END gamma=0.8489138242660839, learning_rate=0.19434590423297465,
max_depth=8, min_child_weight=3, n_estimators=497,
subsample=0.7468977981821954;, score=(train=-8.533, test=-15075.076) total time=
1.3s
[CV 3/5] END gamma=0.5227328293819941, learning_rate=0.13550820367170993,
max_depth=11, min_child_weight=1, n_estimators=963,
subsample=0.8182052056318903;, score=(train=-0.261, test=-18021.616) total time=
2.2s
[CV 2/5] END gamma=0.3143559810763267, learning_rate=0.15171413823294055,
max_depth=6, min_child_weight=3, n_estimators=330,
subsample=0.7051914615178149;, score=(train=-274.921, test=-17976.325) total
time=   0.6s
[CV 5/5] END gamma=0.3143559810763267, learning_rate=0.15171413823294055,
max_depth=6, min_child_weight=3, n_estimators=330,
subsample=0.7051914615178149;, score=(train=-323.947, test=-17857.999) total
time=   0.6s
[CV 3/5] END gamma=0.7555511385430487, learning_rate=0.0957596330983245,
max_depth=8, min_child_weight=3, n_estimators=1366,
subsample=0.5806106436270022;, score=(train=-1.401, test=-16838.658) total time=
2.9s
[CV 2/5] END gamma=0.9296976523425731, learning_rate=0.2116240759128834,
max_depth=10, min_child_weight=1, n_estimators=745,
subsample=0.7282672852414551;, score=(train=-7.611, test=-18807.161) total time=
1.9s
[CV 5/5] END gamma=0.9296976523425731, learning_rate=0.2116240759128834,
max_depth=10, min_child_weight=1, n_estimators=745,
subsample=0.7282672852414551;, score=(train=-0.316, test=-17731.031) total time=
1.5s
[CV 3/5] END gamma=0.2184404372168336, learning_rate=0.13330198957407324,
max_depth=12, min_child_weight=2, n_estimators=1471,
subsample=0.6781489190384875;, score=(train=-0.150, test=-17617.515) total time=
2.9s
[CV 1/5] END gamma=0.906828441545754, learning_rate=0.10442644987692706,
max_depth=12, min_child_weight=2, n_estimators=732,
subsample=0.9090073829612466;, score=(train=-7.564, test=-17437.885) total time=
2.4s
[CV 5/5] END gamma=0.906828441545754, learning_rate=0.10442644987692706,
max_depth=12, min_child_weight=2, n_estimators=732,
subsample=0.9090073829612466;, score=(train=-0.388, test=-18312.694) total time=
2.3s
[CV 3/5] END gamma=0.8607305832563434, learning_rate=0.051390426106238146,
max_depth=9, min_child_weight=3, n_estimators=748,
```

```
subsample=0.7424149856794916;, score=(train=-186.010, test=-16741.309) total
time=   1.9s
[CV 2/5] END gamma=0.6924360328902703, learning_rate=0.1038824667597043,
max_depth=9, min_child_weight=3, n_estimators=1497,
subsample=0.6093821097865351;, score=(train=-8.297, test=-16606.396) total time=
3.6s
[CV 3/5] END gamma=0.5581020020173412, learning_rate=0.13076723421160819,
max_depth=2, min_child_weight=2, n_estimators=663,
subsample=0.6234380314193007;, score=(train=-6627.724, test=-15869.402) total
time=   0.4s
[CV 1/5] END gamma=0.6963042728397884, learning_rate=0.19245411798488843,
max_depth=6, min_child_weight=1, n_estimators=1051,
subsample=0.5184434736772664;, score=(train=-11.363, test=-17752.577) total
time=   2.1s
[CV 5/5] END gamma=0.6963042728397884, learning_rate=0.19245411798488843,
max_depth=6, min_child_weight=1, n_estimators=1051,
subsample=0.5184434736772664;, score=(train=-0.266, test=-17501.773) total time=
2.0s
[CV 4/5] END gamma=0.6095643339798968, learning_rate=0.15053580464577232,
max_depth=7, min_child_weight=3, n_estimators=1348,
subsample=0.8171756723506819;, score=(train=-8.095, test=-14119.953) total time=
2.5s
[CV 4/5] END gamma=0.6807054515547668, learning_rate=0.15618691666342727,
max_depth=11, min_child_weight=2, n_estimators=346,
subsample=0.8360677737029393;, score=(train=-8.345, test=-16324.876) total time=
1.6s
[CV 3/5] END gamma=0.7616196153287176, learning_rate=0.09752750879847993,
max_depth=7, min_child_weight=1, n_estimators=983,
subsample=0.9917115704474215;, score=(train=-0.931, test=-17465.821) total time=
2.9s
[CV 2/5] END gamma=0.39882444244455306, learning_rate=0.21328637464387679,
max_depth=2, min_child_weight=1, n_estimators=1247,
subsample=0.7540993883703593;, score=(train=-2816.162, test=-17115.561) total
time=   0.9s
[CV 5/5] END gamma=0.39882444244455306, learning_rate=0.21328637464387679,
max_depth=2, min_child_weight=1, n_estimators=1247,
subsample=0.7540993883703593;, score=(train=-2650.287, test=-17481.876) total
time=   0.8s
[CV 3/5] END gamma=0.6958128067908819, learning_rate=0.221671760962744,
max_depth=13, min_child_weight=3, n_estimators=502,
subsample=0.855574766219009;, score=(train=-0.305, test=-18501.488) total time=
1.7s
[CV 2/5] END gamma=0.8095010461397154, learning_rate=0.11973319745834587,
max_depth=8, min_child_weight=1, n_estimators=1028,
subsample=0.6987860105437611;, score=(train=-7.930, test=-17313.947) total time=
3.3s
[CV 2/5] END gamma=0.5177513505274801, learning_rate=0.2175420211814656,
max_depth=12, min_child_weight=1, n_estimators=572,
```

```
subsample=0.6045358103688568;, score=(train=-9.582, test=-18179.784) total time=
2.5s
[CV 5/5] END gamma=0.5177513505274801, learning_rate=0.2175420211814656,
max_depth=12, min_child_weight=1, n_estimators=572,
subsample=0.6045358103688568;, score=(train=-0.221, test=-17770.187) total time=
1.8s
[CV 3/5] END gamma=0.5414479738275658, learning_rate=0.18915687986901647,
max_depth=8, min_child_weight=2, n_estimators=723,
subsample=0.7776004057997312;, score=(train=-0.262, test=-17828.434) total time=
1.9s
[CV 2/5] END gamma=0.5296505783560065, learning_rate=0.09837045818009034,
max_depth=13, min_child_weight=1, n_estimators=1160,
subsample=0.9413181715946699;, score=(train=-7.425, test=-18494.403) total time=
3.7s
[CV 2/5] END gamma=0.18870710834137938, learning_rate=0.10577427051843638,
max_depth=4, min_child_weight=3, n_estimators=503,
subsample=0.9435432121325587;, score=(train=-1888.619, test=-16854.215) total
time=   0.6s
[CV 5/5] END gamma=0.18870710834137938, learning_rate=0.10577427051843638,
max_depth=4, min_child_weight=3, n_estimators=503,
subsample=0.9435432121325587;, score=(train=-1933.859, test=-16054.185) total
time=   0.6s
[CV 3/5] END gamma=0.7798755458576239, learning_rate=0.17840632923085759,
max_depth=12, min_child_weight=1, n_estimators=203,
subsample=0.803214529829795;, score=(train=-0.372, test=-18634.650) total time=
1.7s
[CV 2/5] END gamma=0.009197051616629648, learning_rate=0.07029430857320643,
max_depth=4, min_child_weight=1, n_estimators=1013,
subsample=0.5804040257087493;, score=(train=-1170.283, test=-16087.775) total
time=   1.3s
[CV 1/5] END gamma=0.5487337893665861, learning_rate=0.18837903953853868,
max_depth=14, min_child_weight=1, n_estimators=260,
subsample=0.5090376818077604;, score=(train=-10.730, test=-17853.950) total
time=   1.7s
[CV 5/5] END gamma=0.5487337893665861, learning_rate=0.18837903953853868,
max_depth=14, min_child_weight=1, n_estimators=260,
subsample=0.5090376818077604;, score=(train=-2.808, test=-17840.679) total time=
1.8s
[CV 4/5] END gamma=0.4938937151834346, learning_rate=0.08576454184426577,
max_depth=9, min_child_weight=3, n_estimators=1318,
subsample=0.8288064461501716;, score=(train=-7.678, test=-14832.204) total time=
3.6s
[CV 3/5] END gamma=0.5683086033354716, learning_rate=0.0687349535656185,
max_depth=4, min_child_weight=2, n_estimators=1498,
subsample=0.6252309093027921;, score=(train=-452.756, test=-16697.572) total
time=   1.7s
[CV 2/5] END gamma=0.5898708475605439, learning_rate=0.24577857165500183,
max_depth=4, min_child_weight=2, n_estimators=1406,
```

```
subsample=0.7171971827552144;, score=(train=-29.759, test=-16567.997) total
time=   1.7s
[CV 1/5] END gamma=0.3500784076946757, learning_rate=0.17902067240611297,
max_depth=2, min_child_weight=1, n_estimators=1119,
subsample=0.8612260576307527;, score=(train=-3583.756, test=-15208.171) total
time=   0.7s
[CV 2/5] END gamma=0.3500784076946757, learning_rate=0.17902067240611297,
max_depth=2, min_child_weight=1, n_estimators=1119,
subsample=0.8612260576307527;, score=(train=-3642.883, test=-17315.397) total
time=   0.8s
[CV 5/5] END gamma=0.3500784076946757, learning_rate=0.17902067240611297,
max_depth=2, min_child_weight=1, n_estimators=1119,
subsample=0.8612260576307527;, score=(train=-3505.888, test=-16095.607) total
time=   0.8s
[CV 4/5] END gamma=0.2807723624408558, learning_rate=0.05486319328629077,
max_depth=2, min_child_weight=1, n_estimators=1295,
subsample=0.9702292921764571;, score=(train=-7694.487, test=-13394.297) total
time=   0.8s
[CV 3/5] END gamma=0.9539285770025874, learning_rate=0.23297287804408973,
max_depth=4, min_child_weight=1, n_estimators=1160,
subsample=0.9641592812938626;, score=(train=-9.409, test=-17532.007) total time=
1.4s
[CV 2/5] END gamma=0.42818414831731433, learning_rate=0.24333096380873392,
max_depth=4, min_child_weight=1, n_estimators=1239,
subsample=0.6472244460347929;, score=(train=-39.222, test=-18300.671) total
time=   1.5s
[CV 1/5] END gamma=0.38509772860192526, learning_rate=0.22022733430337138,
max_depth=4, min_child_weight=2, n_estimators=665,
subsample=0.8480148983374864;, score=(train=-181.939, test=-16986.356) total
time=   0.8s
[CV 3/5] END gamma=0.38509772860192526, learning_rate=0.22022733430337138,
max_depth=4, min_child_weight=2, n_estimators=665,
subsample=0.8480148983374864;, score=(train=-129.068, test=-17333.750) total
time=   0.8s
[CV 2/5] END gamma=0.3745401188473625, learning_rate=0.24014286128198326,
max_depth=12, min_child_weight=1, n_estimators=1144,
subsample=0.5780093202212182;, score=(train=-8.496, test=-18184.567) total time=
3.1s
[CV 2/5] END gamma=0.15599452033620265, learning_rate=0.061616722433639894,
max_depth=9, min_child_weight=1, n_estimators=1223,
subsample=0.5714334089609704;, score=(train=-8.764, test=-17313.866) total time=
4.6s
[CV 5/5] END gamma=0.15599452033620265, learning_rate=0.061616722433639894,
max_depth=9, min_child_weight=1, n_estimators=1223,
subsample=0.5714334089609704;, score=(train=-0.161, test=-16460.813) total time=
4.8s
[CV 3/5] END gamma=0.9922115592912175, learning_rate=0.17349630192554333,
max_depth=11, min_child_weight=2, n_estimators=352,
```

```
subsample=0.7159725093210578;, score=(train=-1.640, test=-18242.181) total time=
1.3s
[CV 2/5] END gamma=0.2912291401980419, learning_rate=0.17237057894447588,
max_depth=11, min_child_weight=3, n_estimators=289,
subsample=0.5453032172664104;, score=(train=-113.174, test=-19292.055) total
time=   0.8s
[CV 1/5] END gamma=0.6183860093330873, learning_rate=0.12649239825343256,
max_depth=5, min_child_weight=1, n_estimators=1254,
subsample=0.9299702033681603;, score=(train=-18.429, test=-15631.360) total
time=   1.8s
[CV 5/5] END gamma=0.6183860093330873, learning_rate=0.12649239825343256,
max_depth=5, min_child_weight=1, n_estimators=1254,
subsample=0.9299702033681603;, score=(train=-5.693, test=-17107.346) total time=
1.8s
[CV 2/5] END gamma=0.4951769101112702, learning_rate=0.05687770422304368,
max_depth=2, min_child_weight=2, n_estimators=1125,
subsample=0.7125779372456224;, score=(train=-7710.327, test=-16930.351) total
time=   0.6s
[CV 5/5] END gamma=0.4951769101112702, learning_rate=0.05687770422304368,
max_depth=2, min_child_weight=2, n_estimators=1125,
subsample=0.7125779372456224;, score=(train=-7306.251, test=-16597.868) total
time=   0.6s
[CV 4/5] END gamma=0.20794166286818883, learning_rate=0.1635400655639983,
max_depth=14, min_child_weight=3, n_estimators=501,
subsample=0.8875664116805573;, score=(train=-7.518, test=-16487.267) total time=
1.6s
[CV 2/5] END gamma=0.9394989415641891, learning_rate=0.22896547008552975,
max_depth=9, min_child_weight=2, n_estimators=1375,
subsample=0.522613644455269;, score=(train=-9.360, test=-18599.356) total time=
2.7s
[CV 2/5] END gamma=0.32533033076326434, learning_rate=0.12773545793789642,
max_depth=3, min_child_weight=3, n_estimators=1176,
subsample=0.7933755828319241;, score=(train=-1129.420, test=-17608.984) total
time=   0.8s
[CV 4/5] END gamma=0.32533033076326434, learning_rate=0.12773545793789642,
max_depth=3, min_child_weight=3, n_estimators=1176,
subsample=0.7933755828319241;, score=(train=-1171.407, test=-13675.421) total
time=   1.1s
[CV 4/5] END gamma=0.965255307264138, learning_rate=0.17140684953733692,
max_depth=14, min_child_weight=1, n_estimators=1280,
subsample=0.9010984903770198;, score=(train=-7.472, test=-16749.113) total time=
2.4s
[CV 3/5] END gamma=0.07455064367977082, learning_rate=0.24737738732010345,
max_depth=9, min_child_weight=1, n_estimators=747,
subsample=0.5027610585618012;, score=(train=-0.080, test=-19118.136) total time=
1.9s
[CV 1/5] END gamma=0.8154614284548342, learning_rate=0.19137146876952343,
max_depth=4, min_child_weight=3, n_estimators=388,
```

```
subsample=0.8029799873905057;, score=(train=-1043.652, test=-16036.346) total
time=   0.6s
[CV 3/5] END gamma=0.8154614284548342, learning_rate=0.19137146876952343,
max_depth=4, min_child_weight=3, n_estimators=388,
subsample=0.8029799873905057;, score=(train=-1053.029, test=-17156.931) total
time=   0.5s
[CV 5/5] END gamma=0.8154614284548342, learning_rate=0.19137146876952343,
max_depth=4, min_child_weight=3, n_estimators=388,
subsample=0.8029799873905057;, score=(train=-1131.524, test=-17840.111) total
time=   0.5s
[CV 3/5] END gamma=0.926300878513349, learning_rate=0.18021540510038891,
max_depth=13, min_child_weight=3, n_estimators=300,
subsample=0.811649063413779;, score=(train=-14.708, test=-18026.014) total time=
1.1s
[CV 2/5] END gamma=0.3308980248526492, learning_rate=0.06271167005720474,
max_depth=8, min_child_weight=2, n_estimators=904, subsample=0.864803089169032;,
score=(train=-12.890, test=-17092.070) total time=   3.0s
[CV 2/5] END gamma=0.6375574713552131, learning_rate=0.2274425485152653,
max_depth=2, min_child_weight=3, n_estimators=712,
subsample=0.9858560476945519;, score=(train=-4708.874, test=-16864.719) total
time=   0.4s
[CV 4/5] END gamma=0.6375574713552131, learning_rate=0.2274425485152653,
max_depth=2, min_child_weight=3, n_estimators=712,
subsample=0.9858560476945519;, score=(train=-4901.395, test=-14005.792) total
time=   0.5s
[CV 1/5] END gamma=0.8489138242660839, learning_rate=0.19434590423297465,
max_depth=8, min_child_weight=3, n_estimators=497,
subsample=0.7468977981821954;, score=(train=-8.136, test=-16591.328) total time=
1.3s
[CV 5/5] END gamma=0.8489138242660839, learning_rate=0.19434590423297465,
max_depth=8, min_child_weight=3, n_estimators=497,
subsample=0.7468977981821954;, score=(train=-0.733, test=-17904.247) total time=
1.4s
[CV 4/5] END gamma=0.5227328293819941, learning_rate=0.13550820367170993,
max_depth=11, min_child_weight=1, n_estimators=963,
subsample=0.8182052056318903;, score=(train=-8.115, test=-15214.355) total time=
2.6s
[CV 4/5] END gamma=0.3143559810763267, learning_rate=0.15171413823294055,
max_depth=6, min_child_weight=3, n_estimators=330,
subsample=0.7051914615178149;, score=(train=-276.073, test=-14349.072) total
time=   0.6s
[CV 2/5] END gamma=0.7555511385430487, learning_rate=0.0957596330983245,
max_depth=8, min_child_weight=3, n_estimators=1366,
subsample=0.5806106436270022;, score=(train=-10.059, test=-17220.166) total
time=   2.9s
[CV 1/5] END gamma=0.9296976523425731, learning_rate=0.2116240759128834,
max_depth=10, min_child_weight=1, n_estimators=745,
subsample=0.7282672852414551;, score=(train=-7.772, test=-17332.680) total time=
```

```
2.0s
[CV 1/5] END gamma=0.2184404372168336, learning_rate=0.13330198957407324,
max_depth=12, min_child_weight=2, n_estimators=1471,
subsample=0.6781489190384875;, score=(train=-8.168, test=-17099.834) total time=
3.7s
[CV 5/5] END gamma=0.2184404372168336, learning_rate=0.13330198957407324,
max_depth=12, min_child_weight=2, n_estimators=1471,
subsample=0.6781489190384875;, score=(train=-0.150, test=-18152.537) total time=
3.0s
[CV 4/5] END gamma=0.906828441545754, learning_rate=0.10442644987692706,
max_depth=12, min_child_weight=2, n_estimators=732,
subsample=0.9090073829612466;, score=(train=-7.602, test=-15784.241) total time=
2.4s
[CV 4/5] END gamma=0.8607305832563434, learning_rate=0.051390426106238146,
max_depth=9, min_child_weight=3, n_estimators=748,
subsample=0.7424149856794916;, score=(train=-88.544, test=-14775.547) total
time=   2.1s
[CV 3/5] END gamma=0.6924360328902703, learning_rate=0.1038824667597043,
max_depth=9, min_child_weight=3, n_estimators=1497,
subsample=0.6093821097865351;, score=(train=-0.312, test=-17118.865) total time=
3.3s
[CV 1/5] END gamma=0.5581020020173412, learning_rate=0.13076723421160819,
max_depth=2, min_child_weight=2, n_estimators=663,
subsample=0.6234380314193007;, score=(train=-6894.284, test=-15704.855) total
time=   0.4s
[CV 5/5] END gamma=0.5581020020173412, learning_rate=0.13076723421160819,
max_depth=2, min_child_weight=2, n_estimators=663,
subsample=0.6234380314193007;, score=(train=-6704.493, test=-17120.593) total
time=   0.4s
[CV 2/5] END gamma=0.6963042728397884, learning_rate=0.19245411798488843,
max_depth=6, min_child_weight=1, n_estimators=1051,
subsample=0.5184434736772664;, score=(train=-10.116, test=-18205.597) total
time=   2.1s
[CV 2/5] END gamma=0.6095643339798968, learning_rate=0.15053580464577232,
max_depth=7, min_child_weight=3, n_estimators=1348,
subsample=0.8171756723506819;, score=(train=-8.336, test=-17091.870) total time=
2.5s
[CV 1/5] END gamma=0.6807054515547668, learning_rate=0.15618691666342727,
max_depth=11, min_child_weight=2, n_estimators=346,
subsample=0.8360677737029393;, score=(train=-8.028, test=-17680.011) total time=
1.5s
[CV 3/5] END gamma=0.6807054515547668, learning_rate=0.15618691666342727,
max_depth=11, min_child_weight=2, n_estimators=346,
subsample=0.8360677737029393;, score=(train=-3.888, test=-18029.026) total time=
1.3s
[CV 1/5] END gamma=0.7616196153287176, learning_rate=0.09752750879847993,
max_depth=7, min_child_weight=1, n_estimators=983,
subsample=0.9917115704474215;, score=(train=-8.945, test=-16678.254) total time=
```

```
2.8s
[CV 5/5] END gamma=0.7616196153287176, learning_rate=0.09752750879847993,
max_depth=7, min_child_weight=1, n_estimators=983,
subsample=0.9917115704474215;, score=(train=-0.899, test=-17685.678) total time=
2.8s
[CV 4/5] END gamma=0.6958128067908819, learning_rate=0.221671760962744,
max_depth=13, min_child_weight=3, n_estimators=502,
subsample=0.855574766219009;, score=(train=-7.762, test=-16212.478) total time=
1.6s
[CV 3/5] END gamma=0.8095010461397154, learning_rate=0.11973319745834587,
max_depth=8, min_child_weight=1, n_estimators=1028,
subsample=0.6987860105437611;, score=(train=-0.309, test=-16625.633) total time=
2.8s
[CV 1/5] END gamma=0.5177513505274801, learning_rate=0.2175420211814656,
max_depth=12, min_child_weight=1, n_estimators=572,
subsample=0.6045358103688568;, score=(train=-10.052, test=-17482.793) total
time=   2.4s
[CV 4/5] END gamma=0.5177513505274801, learning_rate=0.2175420211814656,
max_depth=12, min_child_weight=1, n_estimators=572,
subsample=0.6045358103688568;, score=(train=-9.734, test=-15736.554) total time=
2.2s
[CV 5/5] END gamma=0.5414479738275658, learning_rate=0.18915687986901647,
max_depth=8, min_child_weight=2, n_estimators=723,
subsample=0.7776004057997312;, score=(train=-0.262, test=-19016.782) total time=
1.9s
[CV 3/5] END gamma=0.5296505783560065, learning_rate=0.09837045818009034,
max_depth=13, min_child_weight=1, n_estimators=1160,
subsample=0.9413181715946699;, score=(train=-0.309, test=-18160.633) total time=
3.3s
[CV 1/5] END gamma=0.18870710834137938, learning_rate=0.10577427051843638,
max_depth=4, min_child_weight=3, n_estimators=503,
subsample=0.9435432121325587;, score=(train=-2012.558, test=-15298.186) total
time=   0.6s
[CV 4/5] END gamma=0.18870710834137938, learning_rate=0.10577427051843638,
max_depth=4, min_child_weight=3, n_estimators=503,
subsample=0.9435432121325587;, score=(train=-1900.865, test=-14080.135) total
time=   0.6s
[CV 2/5] END gamma=0.7798755458576239, learning_rate=0.17840632923085759,
max_depth=12, min_child_weight=1, n_estimators=203,
subsample=0.803214529829795;, score=(train=-7.836, test=-18686.062) total time=
1.7s
[CV 1/5] END gamma=0.009197051616629648, learning_rate=0.07029430857320643,
max_depth=4, min_child_weight=1, n_estimators=1013,
subsample=0.5804040257087493;, score=(train=-1090.900, test=-14663.066) total
time=   1.3s
[CV 5/5] END gamma=0.009197051616629648, learning_rate=0.07029430857320643,
max_depth=4, min_child_weight=1, n_estimators=1013,
subsample=0.5804040257087493;, score=(train=-1073.945, test=-16565.489) total
```

```
time=   1.3s
[CV 4/5] END gamma=0.5487337893665861, learning_rate=0.18837903953853868,
max_depth=14, min_child_weight=1, n_estimators=260,
subsample=0.5090376818077604;, score=(train=-11.355, test=-15938.775) total
time=   1.8s
[CV 3/5] END gamma=0.4938937151834346, learning_rate=0.08576454184426577,
max_depth=9, min_child_weight=3, n_estimators=1318,
subsample=0.8288064461501716;, score=(train=-0.277, test=-17012.799) total time=
3.6s
[CV 1/5] END gamma=0.5683086033354716, learning_rate=0.0687349535656185,
max_depth=4, min_child_weight=2, n_estimators=1498,
subsample=0.6252309093027921;, score=(train=-501.516, test=-14662.001) total
time=   1.8s
[CV 5/5] END gamma=0.5683086033354716, learning_rate=0.0687349535656185,
max_depth=4, min_child_weight=2, n_estimators=1498,
subsample=0.6252309093027921;, score=(train=-444.461, test=-16879.124) total
time=   1.8s
[CV 4/5] END gamma=0.5898708475605439, learning_rate=0.24577857165500183,
max_depth=4, min_child_weight=2, n_estimators=1406,
subsample=0.7171971827552144;, score=(train=-24.208, test=-15396.716) total
time=   1.7s
[CV 4/5] END gamma=0.3500784076946757, learning_rate=0.17902067240611297,
max_depth=2, min_child_weight=1, n_estimators=1119,
subsample=0.8612260576307527;, score=(train=-3792.814, test=-13736.330) total
time=   0.7s
[CV 3/5] END gamma=0.2807723624408558, learning_rate=0.05486319328629077,
max_depth=2, min_child_weight=1, n_estimators=1295,
subsample=0.9702292921764571;, score=(train=-7277.462, test=-16010.043) total
time=   0.8s
[CV 2/5] END gamma=0.9539285770025874, learning_rate=0.23297287804408973,
max_depth=4, min_child_weight=1, n_estimators=1160,
subsample=0.9641592812938626;, score=(train=-25.959, test=-17417.986) total
time=   1.4s
[CV 1/5] END gamma=0.42818414831731433, learning_rate=0.24333096380873392,
max_depth=4, min_child_weight=1, n_estimators=1239,
subsample=0.6472244460347929;, score=(train=-36.121, test=-16267.581) total
time=   1.6s
[CV 5/5] END gamma=0.42818414831731433, learning_rate=0.24333096380873392,
max_depth=4, min_child_weight=1, n_estimators=1239,
subsample=0.6472244460347929;, score=(train=-5.010, test=-17668.675) total time=
1.5s
```

[57]:
```python
import gradio as gr

def create_dummies(df, columns):
    for column in columns:
        if column in df.columns:
```

```
            dummies = pd.get_dummies(df[column], prefix=column, drop_first=True)
            df = pd.concat([df, dummies], axis=1)
            df.drop([column], axis=1, inplace=True)
    return df

# Funksjon for å gjøre prediksjoner
def predict_price(input_file):
    # Les inn data fra CSV-filen
    input_df = pd.read_csv(input_file.name)  # Tilgang til filnavnet fra␣
 ↪filobjektet


    # Bruk modellen til å gjøre prediksjon
    prediction = best_model.predict(input_df)
    return prediction[0]

# Opprett Gradio-grensesnittet med en inndatakomponent for filopplasting
iface = gr.Interface(
    fn=predict_price,
    inputs=gr.File(label="Last opp CSV-fil", type="file"),  # Liste over␣
 ↪gyldige filtyper
    outputs="text"  # Du kan endre dette basert på prediksjonens output-format
)

# Start Gradio-grensesnittet
iface.launch()
```

```
Running on local URL:  http://127.0.0.1:7870

To create a public link, set `share=True` in `launch()`.
```

<IPython.core.display.HTML object>

[57]:

[ ]: