

House_Prices.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text RAM Disk

Sequential on House_Prices Dataset

Libraries and Packages Required

```
[70] import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()

     import tensorflow as tf
     from tensorflow import keras
     from tensorflow.keras import layers, models

     from sklearn.metrics import root_mean_squared_error, mean_squared_error, mean_absolute_error, r2_score
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.feature_selection import RFE
     from sklearn.decomposition import PCA
```

Loading Datasets

- datasets consist 3 different CSV format files
- train, test, and one with predicted prices

```
[71] train = pd.read_csv('train.csv')
     test = pd.read_csv('test.csv')
     predicted =pd.read_csv('sample_submission.csv')

[72] train.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 MasVnType 588 non-null object
26 MasVnArea 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 LowQualInSF 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

```
[73] test.info()
```

24 Exterior2nd 1458 non-null object
25 MasVnType 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 LowQualInSF 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

```

▼ Basic EDA

```

✓ [74] # separating the categorical and numerical features for treatment of null values
categorical_features = []
numerical_features = []
for i in train.columns:
    if train[i].dtype==object:
        categorical_features.append(i)
    else:
        numerical_features.append(i)

test_categorical_features = []
test_numerical_features = []
for i in test.columns:
    if test[i].dtype==object:
        test_categorical_features.append(i)
    else:
        test_numerical_features.append(i)

```

✓ ⏪ test_categorical_features, categorical_features

```

↪ ([ 'MSZoning',
  'Street',
  'Alley',
  'Lotshape',
  'LandContour',
  'Utilities',
  'Lotconfig',
  'LandSlope',
  'Neighborhood',
  'Condition1',
  'Condition2',
  'BldgType',
  'HouseStyle',
  'RoofStyle',
  'RoofMatl',
  'Exterior1st',
  'Exterior2nd',
  'MasVnrType',
  'ExterQual',
  'ExterCond',
  'Foundation',
  'BsmtQual',
  'BsmtCond',
  'BsmtExposure',
  'BsmtFinType1',
  'BsmtFinType2',
  'Heating',
  'HeatingQC',
  'CentralAir',
  'Electrical',
  'Kitchenqual',
  'Functional',
  'FireplaceQu'
])

```

+ Code + Text

```
'FireplaceQu',
'GarageType',
'GarageFinish',
'GarageQual',
'GarageCond',
'PavedDrive',
'PoolQC',
'Fence',
'MiscFeature',
'SaleType',
['SaleCondition'],  
[ 'MSZoning',
'Street',
'Alley',
'LotShape',
'LandContour',
'Utilities',
'LotConfig',
'LandSlope',
'Neighborhood',
'Condition1',
'Condition2',
'BldgType',
'HouseStyle',
'RoofStyle',
'RoofMatl',
```

```
[76] train.loc[:,categorical_features].info()
```

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 43 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   MSZoning    1460 non-null   object 
 1   Street      1460 non-null   object 
 2   Alley       91 non-null    object 
 3   LotShape    1460 non-null   object 
 4   LandContour 1460 non-null   object 
 5   Utilities   1460 non-null   object 
 6   LotConfig   1460 non-null   object 
 7   LandSlope   1460 non-null   object 
 8   Neighborhood 1460 non-null   object 
 9   Condition1  1460 non-null   object 
 10  Condition2  1460 non-null   object 
 11  BldgType    1460 non-null   object 
 12  HouseStyle  1460 non-null   object 
 13  RoofStyle   1460 non-null   object 
 14  RoofMatl   1460 non-null   object 
 15  Exterior1st 1460 non-null   object 
 16  Exterior2nd 1460 non-null   object 
 17  MasVnrType  588 non-null   object 
 18  ExterQual   1460 non-null   object 
 19  ExterCond   1460 non-null   object 
 20  Foundation  1460 non-null   object 
 21  BsmtQual   1423 non-null   object 
 22  BsmtCond   1423 non-null   object 
 23  BsmtExposure 1422 non-null   object 
 24  BsmtFinType1 1423 non-null   object 
 25  BsmtFinType2 1422 non-null   object 
 26  Heating     1460 non-null   object 
 27  HeatingQC   1460 non-null   object 
 28  CentralAir  1460 non-null   object 
 29  Electrical  1459 non-null   object 
 30  KitchenQual 1460 non-null   object 
 31  Functional  1460 non-null   object 
 32  FireplaceQu 770 non-null   object 
 33  GarageType   1379 non-null   object 
 34  GarageFinish 1379 non-null   object 
 35  GarageQual   1379 non-null   object 
 36  GarageCond   1379 non-null   object 
 37  PavedDrive   1460 non-null   object 
 38  PoolQC      7 non-null    object 
 39  Fence        281 non-null   object 
 40  MiscFeature  54 non-null   object 
 41  SaleType     1460 non-null   object 
 42  SaleCondition 1460 non-null   object 
dtypes: object(43)
memory usage: 490.6+ KB
```

```
[77] test.loc[:,test_categorical_features].info()
```

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 43 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   MSZoning    1455 non-null   object 
 1   Street      1459 non-null   object 
 2   Alley       107 non-null    object 
 3   LotShape    1459 non-null   object 
 4   LandContour 1459 non-null   object 
 5   Utilities   1457 non-null   object 
 6   LotConfig   1459 non-null   object 
 7   LandSlope   1459 non-null   object 
 8   Neighborhood 1459 non-null   object 
 9   Condition1  1459 non-null   object 
 10  Condition2  1459 non-null   object 
 11  BldgType    1459 non-null   object 
 12  HouseStyle  1459 non-null   object 
 13  RoofStyle   1459 non-null   object 
 14  RoofMatl   1459 non-null   object 
 15  Exterior1st 1458 non-null   object 
 16  Exterior2nd 1458 non-null   object 
 17  MasVnrType  565 non-null   object 
 18  ExterQual   1459 non-null   object 
 19  ExterCond   1459 non-null   object 
 20  Foundation  1459 non-null   object 
 21  BsmtQual   1415 non-null   object 
 22  BsmtCond   1414 non-null   object 
 23  BsmtExposure 1415 non-null   object 
 24  BsmtFinType1 1417 non-null   object 
 25  BsmtFinType2 1417 non-null   object 
 26  Heating     1459 non-null   object 
 27  HeatingQC   1459 non-null   object
```

```

28 CentralAir    1459 non-null object
29 Electrical   1459 non-null object
30 KitchenQual  1458 non-null object
31 Functional   1457 non-null object
32 FireplaceQu  729 non-null object
33 GarageType    1383 non-null object
34 GarageFinish  1381 non-null object
35 GarageQual    1381 non-null object
36 GarageCond    1381 non-null object
37 PavedDrive   1459 non-null object
38 PoolQC       3 non-null object
39 Fence        290 non-null object
40 MiscFeature   51 non-null object
41 SaleType      1458 non-null object
42 SaleCondition 1459 non-null object
dtypes: object(43)
memory usage: 490.3+ KB

```

Removing Features

```

[78] # removing the listed below columns from categorical_features as percentile of null values are more than actual ones
drop_list = ['Alley', 'MasVnrType', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']
print("Dropping Features List : {drop_list}")

print(f"length of categorical_feature : {len(categorical_features)}")
print(f"length of test_categorical_feature : {len(test_categorical_features)}")

for i in drop_list:
    categorical_features.remove(i)
    test_categorical_features.remove(i)
print("features removed successfully")
print(f"length of categorical_feature : {len(categorical_features)}")
print(f"length of test_categorical_feature : {len(test_categorical_features)}")

→ Dropping Features List : ['Alley', 'MasVnrType', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']
length of categorical_feature : 43
length of test_categorical_feature : 43
features removed successfully
length of categorical_feature : 37
length of test_categorical_feature : 37

```

Replacing Null values in some Features

```

[79] treating_feature = train.loc[:,categorical_features].columns[train.loc[:,categorical_features].isna().any()].tolist()
treating_feature_test = test.loc[:,test_categorical_features].columns[test.loc[:,test_categorical_features].isna().any()].tolist()

for i in treating_feature:
    train[i]= train[i].fillna(train[i].mode()[0])

for i in treating_feature_test:
    test[i] = test[i].fillna(test[i].mode()[0])

```

```
[80] test.loc[:,test_categorical_features].info()
```

```

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 37 columns):
 #   Column      Non-Null Count  Dtype  
--- 
0   MSZoning    1459 non-null   object 
1   Street      1459 non-null   object 
2   LotShape    1459 non-null   object 
3   LandContour 1459 non-null   object 
4   Utilities   1459 non-null   object 
5   LotConfig   1459 non-null   object 
6   LandSlope   1459 non-null   object 
7   Neighborhood 1459 non-null   object 
8   Condition1  1459 non-null   object 
9   Condition2  1459 non-null   object 
10  BldgType    1459 non-null   object 
11  HouseStyle  1459 non-null   object 
12  RoofStyle   1459 non-null   object 
13  RoofMatl   1459 non-null   object 
14  Exterior1st 1459 non-null   object 
15  Exterior2nd 1459 non-null   object 
16  ExterQual   1459 non-null   object 
17  ExterCond   1459 non-null   object 
18  Foundation  1459 non-null   object 
19  BsmtQual   1459 non-null   object 
20  BsmtCond   1459 non-null   object 
21  BsmtExposure 1459 non-null   object 
22  BsmtFinType1 1459 non-null   object 
23  BsmtFinType2 1459 non-null   object 
24  Heating     1459 non-null   object 
25  HeatingQC   1459 non-null   object 
26  CentralAir  1459 non-null   object 
27  Electrical  1459 non-null   object 
28  KitchenQual 1459 non-null   object 
29  Functional  1459 non-null   object 
30  GarageType  1459 non-null   object 
31  GarageFinish 1459 non-null   object 
32  GarageQual  1459 non-null   object 
33  GarageCond  1459 non-null   object 
34  PavedDrive  1459 non-null   object 
35  SaleType    1459 non-null   object 
36  SaleCondition 1459 non-null   object
dtypes: object(37)
memory usage: 421.9+ KB

```

Double-click (or enter) to edit

```

[81] train.loc[:,categorical_features].info()

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 37 columns):
 #   Column      Non-Null Count  Dtype  
--- 
0   MSZoning    1460 non-null   object 
1   Street      1460 non-null   object 
2   LotShape    1460 non-null   object 
3   LandContour 1460 non-null   object 
4   Utilities   1460 non-null   object 
5   LotConfig   1460 non-null   object 
6   LandSlope   1460 non-null   object 
7   Neighborhood 1460 non-null   object 
8   Condition1  1460 non-null   object 
9   Condition2  1460 non-null   object 
10  BldgType    1460 non-null   object 
11  HouseStyle  1460 non-null   object 
12  RoofStyle   1460 non-null   object 
13  RoofMatl   1460 non-null   object 
14  Exterior1st 1460 non-null   object 
15  Exterior2nd 1460 non-null   object 
16  ExterQual   1460 non-null   object 
17  ExterCond   1460 non-null   object 
18  Foundation  1460 non-null   object 
19  BsmtQual   1460 non-null   object 
20  BsmtCond   1460 non-null   object 
21  BsmtExposure 1460 non-null   object 
22  BsmtFinType1 1460 non-null   object 
23  BsmtFinType2 1460 non-null   object 
24  Heating     1460 non-null   object 
25  HeatingQC   1460 non-null   object 
26  CentralAir  1460 non-null   object 
27  Electrical  1460 non-null   object 
28  KitchenQual 1460 non-null   object 
29  Functional  1460 non-null   object 
30  GarageType  1460 non-null   object 
31  GarageFinish 1460 non-null   object 
32  GarageQual  1460 non-null   object 
33  GarageCond  1460 non-null   object 
34  PavedDrive  1460 non-null   object 
35  SaleType    1460 non-null   object 
36  SaleCondition 1460 non-null   object
dtypes: object(37)
memory usage: 421.9+ KB

```

```

0 MSZoning      1460 non-null object
1 Street        1460 non-null object
2 Lotshape       1460 non-null object
3 LandContour    1460 non-null object
4 Utilities      1460 non-null object
5 LotConfig      1460 non-null object
6 LandSlope      1460 non-null object
7 Neighborhood   1460 non-null object
8 Condition1     1460 non-null object
9 Condition2     1460 non-null object
10 BldgType      1460 non-null object
11 HouseStyle     1460 non-null object
12 RoofStyle      1460 non-null object
13 RoofMatl      1460 non-null object
14 Exterior1st    1460 non-null object
15 Exterior2nd    1460 non-null object
16 ExterQual     1460 non-null object
17 ExterCond      1460 non-null object
18 Foundation     1460 non-null object
19 BsmtQual      1460 non-null object
20 BsmtCond      1460 non-null object
21 BsmtExposure   1460 non-null object
22 BsmtFinType1   1460 non-null object
23 BsmtFinType2   1460 non-null object
24 Heating         1460 non-null object
25 HeatingQC      1460 non-null object
26 CentralAir     1460 non-null object
27 Electrical      1460 non-null object
28 KitchenQual    1460 non-null object
29 Functional      1460 non-null object
30 GarageType     1460 non-null object
31 GarageFinish    1460 non-null object
32 GarageQual     1460 non-null object
33 GarageCond     1460 non-null object
34 PavedDrive     1460 non-null object
35 SaleType        1460 non-null object
36 SaleCondition   1460 non-null object
dtypes: object(37)
memory usage: 422.2+ KB

```

Replacing null values in numerical_features

```

[82] treating_numerical_feature = train.loc[:,numerical_features].columns[train.loc[:,numerical_features].isna().any()].tolist()
treating_numerical_feature_test = test.loc[:,test_numerical_features].columns[test.loc[:,test_numerical_features].isna().any()].tolist()

for i in treating_numerical_feature:
    train[i] = train[i].fillna(train[i].mean())

for i in treating_numerical_feature_test:
    test[i] = test[i].fillna(test[i].mean())

```

```
[83] train.loc[:,numerical_features].info()
```

```

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 38 columns):
 #   Column      Non-Null Count Dtype  
 ---  --          --          --      
 0   Id          1460 non-null  int64  
 1   MSSubClass   1460 non-null  int64  
 2   LotFrontage  1460 non-null  float64 
 3   LotArea      1460 non-null  int64  
 4   OverallQual  1460 non-null  int64  
 5   OverallCond  1460 non-null  int64  
 6   YearBuilt    1460 non-null  int64  
 7   YearRemodAdd 1460 non-null  int64  
 8   MasVnrArea   1460 non-null  float64 
 9   BsmtFinSF1   1460 non-null  int64  
 10  BsmtFinSF2   1460 non-null  int64  
 11  BsmtUnfSF    1460 non-null  int64  
 12  TotalBsmtSF  1460 non-null  int64  
 13  1stFlrSF     1460 non-null  int64  
 14  2ndFlrSF     1460 non-null  int64  
 15  LowQualInSF  1460 non-null  int64  
 16  GrLivArea    1460 non-null  int64  
 17  BsmtFullBath 1460 non-null  int64  
 18  BsmtHalfBath 1460 non-null  int64  
 19  FullBath     1460 non-null  int64  
 20  HalfBath     1460 non-null  int64  
 21  BedrommAbvGr 1460 non-null  int64  
 22  KitchenAbvGr 1460 non-null  int64  
 23  TotRmsAbvGrd 1460 non-null  int64  
 24  Fireplaces   1460 non-null  int64  
 25  GarageYrBlt   1460 non-null  float64 
 26  GarageCars    1460 non-null  int64  
 27  GarageArea    1460 non-null  int64  
 28  WoodDeckSF    1460 non-null  int64  
 29  OpenPorchSF   1460 non-null  int64  
 30  EnclosedPorch 1460 non-null  int64  
 31  3SsnPorch    1460 non-null  int64  
 32  ScreenPorch   1460 non-null  int64  
 33  PoolArea      1460 non-null  int64  
 34  MiscVal       1460 non-null  int64  
 35  MoSold        1460 non-null  int64  
 36  YrSold        1460 non-null  int64  
 37  SalePrice     1460 non-null  int64  
dtypes: float64(3), int64(35)
memory usage: 433.6 KB

```

```
[84] test.loc[:,test_numerical_features].info()
```

```

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 37 columns):
 #   Column      Non-Null Count Dtype  
 ---  --          --          --      
 0   Id          1459 non-null  int64  
 1   MSSubClass   1459 non-null  int64  
 2   LotFrontage  1459 non-null  float64 
 3   LotArea      1459 non-null  int64  
 4   OverallQual  1459 non-null  int64  
 5   OverallCond  1459 non-null  int64

```

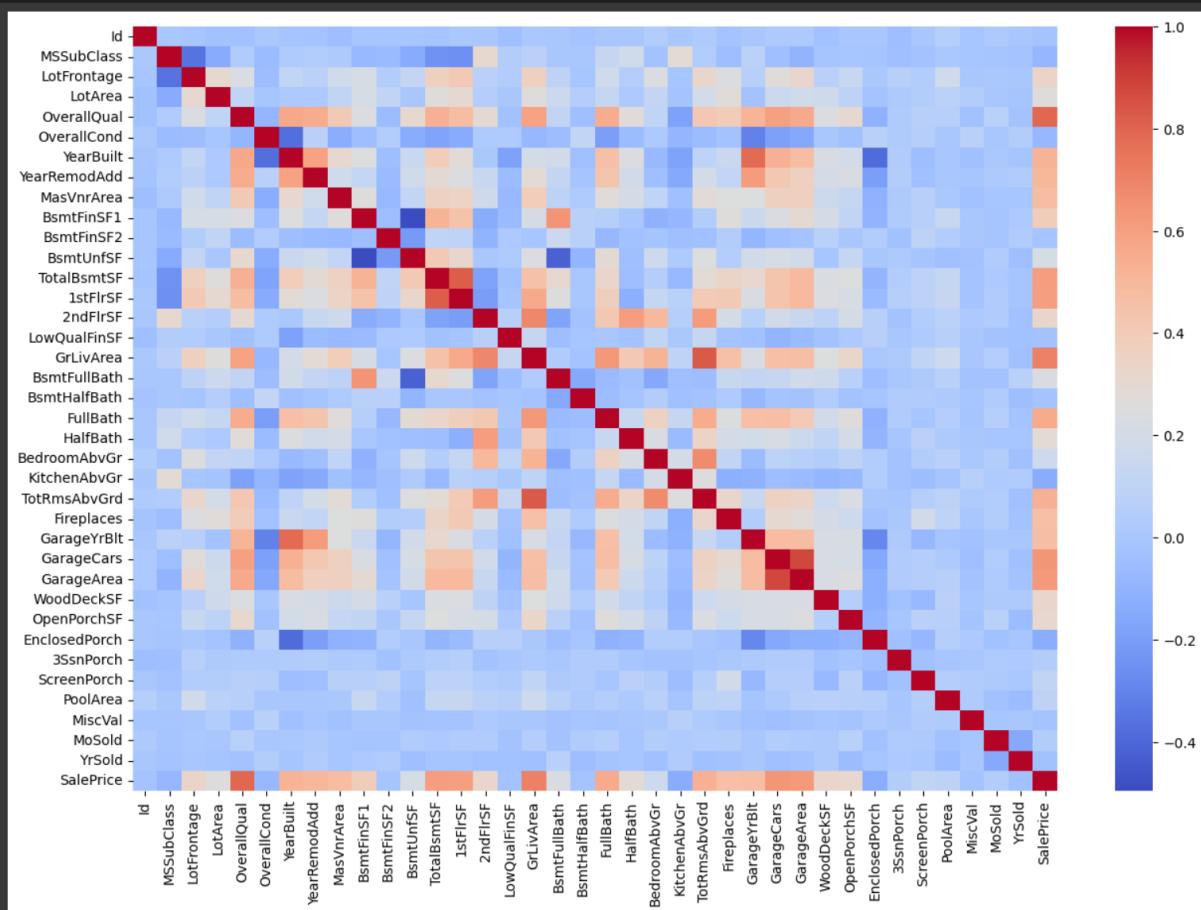
```

5 OverallCond 1459 non-null int64
6 YearBuilt 1459 non-null int64
7 YearRemodAdd 1459 non-null int64
8 MasVnrArea 1459 non-null float64
9 BsmtFinSF1 1459 non-null float64
10 BsmtFinSF2 1459 non-null float64
11 BsmtUnfSF 1459 non-null float64
12 TotalBsmtSF 1459 non-null float64
13 1stFlrSF 1459 non-null int64
14 2ndFlrSF 1459 non-null int64
15 LowQualFinSF 1459 non-null int64
16 GrLivArea 1459 non-null int64
17 BsmtFullBath 1459 non-null float64
18 BsmtHalfBath 1459 non-null float64
19 FullBath 1459 non-null int64
20 HalfBath 1459 non-null int64
21 BedroomAbvGr 1459 non-null int64
22 KitchenAbvGr 1459 non-null int64
23 TotRmsAbvGrd 1459 non-null int64
24 Fireplaces 1459 non-null int64
25 GarageYrBlt 1459 non-null float64
26 GarageCars 1459 non-null float64
27 GarageArea 1459 non-null float64
28 WoodDeckSF 1459 non-null int64
29 OpenPorchSF 1459 non-null int64
30 EnclosedPorch 1459 non-null int64
31 3SsnPorch 1459 non-null int64
32 ScreenPorch 1459 non-null int64
33 PoolArea 1459 non-null int64
34 MiscVal 1459 non-null int64
35 MoSold 1459 non-null int64
36 YrSold 1459 non-null int64
dtypes: float64(11), int64(26)
memory usage: 421.9 KB

```

Correlation between Numerical_features

```
[85] plt.figure(figsize=(15,10))
sns.heatmap(train.loc[:,numerical_features].corr(),cmap='coolwarm')
plt.show()
```

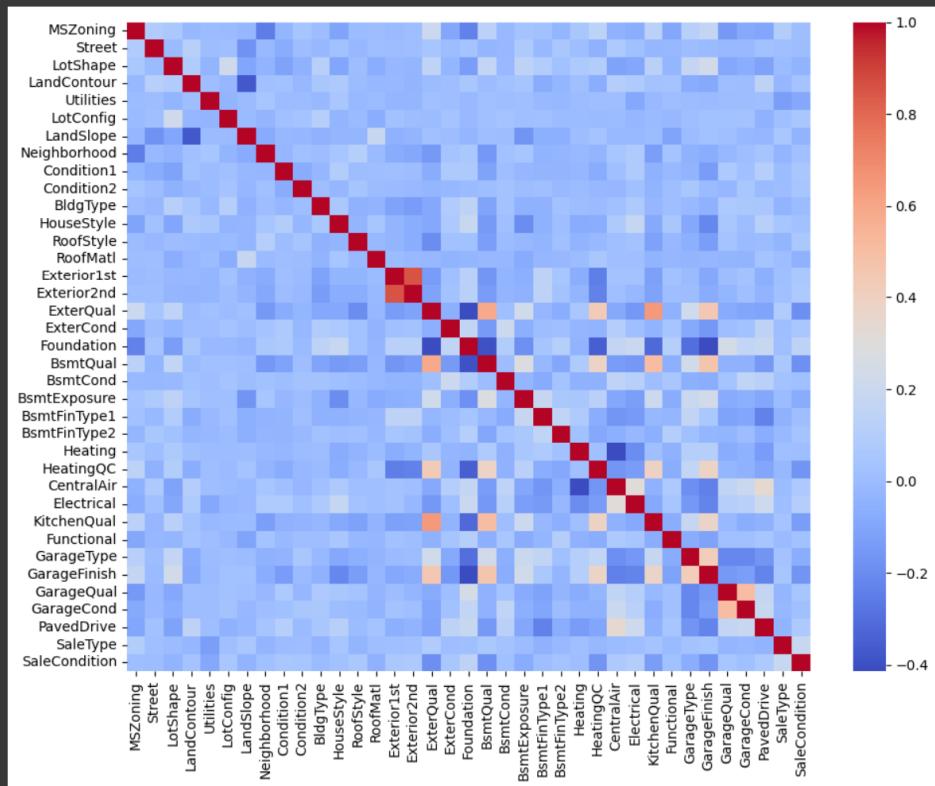


Converting Categorical features to numerical for model ready

```
[86] # for training set
for i in categorical_features:
    train[i] = le.fit_transform(train[i])
le.classes_
# for testing set
for i in test_categorical_features:
    test[i] = le.fit_transform(test[i])
le.classes_
array(['Abnorml', 'AdjLand', 'Alloca', 'Family', 'Normal', 'Partial'],
      dtype=object)
```

Correlation b/w categorical features

```
[87] plt.figure(figsize=(10,8))
sns.heatmap(train.loc[:,categorical_features].corr(),cmap='coolwarm')
plt.tight_layout()
plt.show()
```



using new Instances for training and testing set

```
[88] train_df = train.loc[:,categorical_features+numerical_features]
train_df.head()
```

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	Po...
0	3	1	3	3	0	4	0	5	2	2	...	0	61	0	0	0	0
1	3	1	3	3	0	2	0	24	1	2	...	298	0	0	0	0	0
2	3	1	0	3	0	4	0	5	2	2	...	0	42	0	0	0	0
3	3	1	0	3	0	0	0	6	2	2	...	0	35	272	0	0	0
4	3	1	0	3	0	2	0	15	2	2	...	192	84	0	0	0	0

5 rows × 75 columns

```
[89] test_df = test.loc[:,test_categorical_features+test_numerical_features]
test_df.head()
```

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	...	GarageArea	WoodDecksF	OpenPorchesF	EnclosedPorches	3SsnPorches	ScreenPorch	Po...
0	2	1	3	3	0	4	0	12	1	2	...	730.0	140	0	0	0	0	
1	3	1	0	3	0	0	0	12	2	2	...	312.0	393	36	0	0	0	
2	3	1	0	3	0	4	0	8	2	2	...	482.0	212	34	0	0	0	
3	3	1	0	3	0	4	0	8	2	2	...	470.0	360	36	0	0	0	
4	3	1	0	1	0	4	0	22	2	2	...	506.0	0	82	0	0	0	

5 rows × 74 columns

▼ Splitting features and target

```
[90] X = train_df.iloc[:, :-1]
Y = train_df['SalePrice']
```

```
[91] X.shape, Y.shape
```

((1460, 74), (1460,))

▼ RFE

```
[93] rf = RandomForestRegressor()
rfe = RFE(rf,n_features_to_select=20)
rfe.fit(X,Y)
```

```
rfe.support_
```

```
↳ array([False, False, False, False, False, False, True, False,
       False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,
       False, True, False, True, True, True, True, True, True,
       True, True, False, True, True, True, True, True, False, True,
       False, False, False, False, False, True, False, True, True,
       True, True, False, True, False, False, False, False, False,
       False, False])
```

```
[94] selected_features = [feature for feature,selected in zip(X.columns,rfe.support_) if selected]
selected_features
```

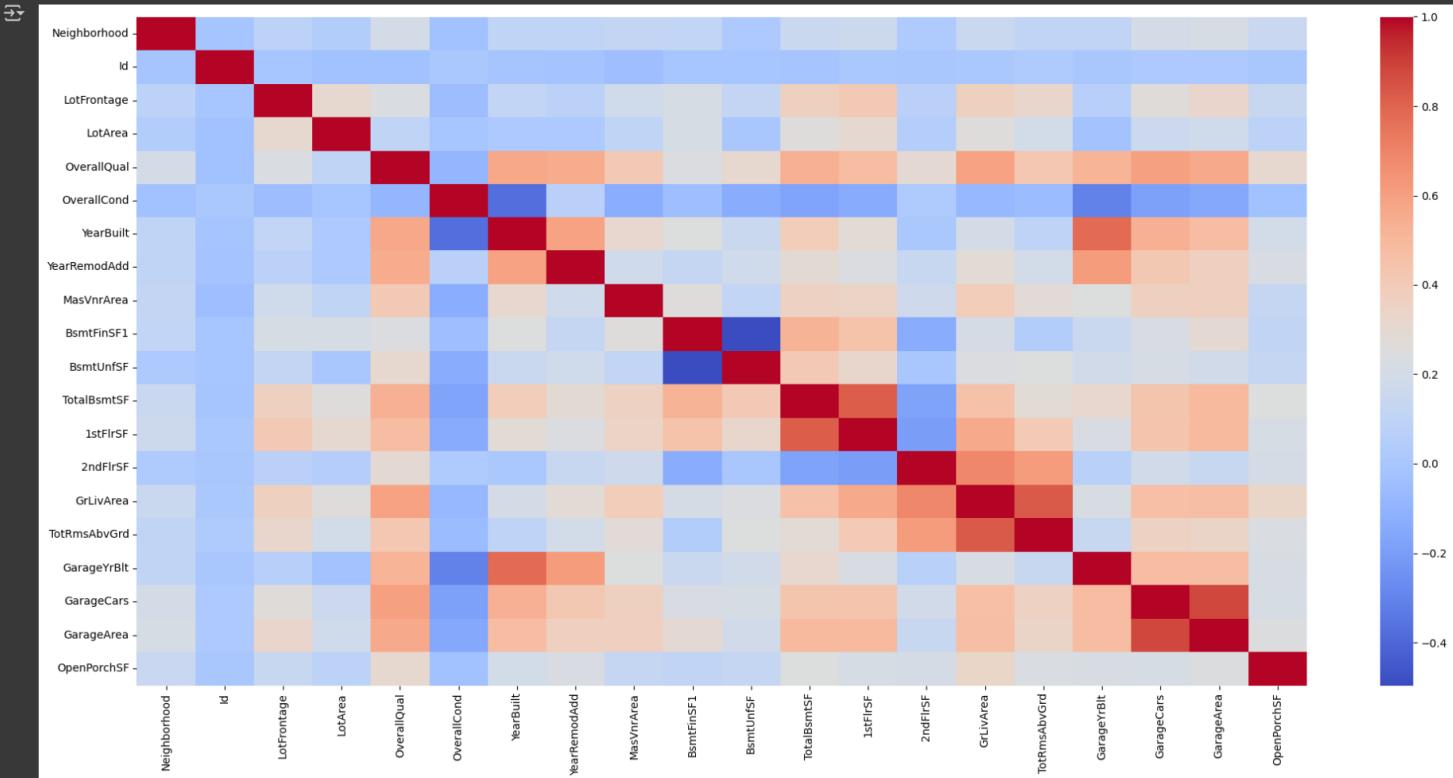
```
↳ ['Neighborhood',
 'Id',
 'LotFrontage',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'YearRemodAdd',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'GrLivArea',
 'TotRmsAbvGrd',
 'GarageYrBlt',
 'GarageCars',
 'GarageArea',
 'OpenPorchSF']
```

```
[95] x_train = X.loc[:,selected_features]
y_train = Y
```

```
x_train.shape, y_train.shape
```

```
↳ ((1460, 20), (1460,))
```

```
[96] plt.figure(figsize=(20,10))
sns.heatmap(x_train.corr(),cmap='coolwarm')
plt.tight_layout()
plt.show()
```



Model Deployment

```
[97] model = tf.keras.models.Sequential()
model.add(layers.Dense(128,input_shape=(x_train.shape[1]),activation='relu'))
model.add(layers.Dense(256,activation='relu'))
model.add(layers.Dropout(0.20))
model.add(layers.Dense(64,activation='relu'))
model.add(layers.Dense(1))
```

```
↳ /usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape` / `input_dim` argument to a layer. When using Sequential models, prefer `._init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
[98] model.compile(optimizer='adam',
                  loss='mse',
```

```

metrics=['mae'])

55s ⏎ history = model.fit(x_train,y_train,epochs=200,
                           batch_size = 64,
                           verbose = 1 , validation_split=0.20)

→ Epoch 1/200
19/19 2s 22ms/step - loss: 37749923840.0000 - mae: 176947.2344 - val_loss: 27301527552.0000 - val_mae: 145823.8750
Epoch 2/200
19/19 0s 8ms/step - loss: 21396987904.0000 - mae: 121807.9219 - val_loss: 7643740160.0000 - val_mae: 54383.3906
Epoch 3/200
19/19 0s 10ms/step - loss: 8169707008.0000 - mae: 51935.0859 - val_loss: 5810372608.0000 - val_mae: 46908.0273
Epoch 4/200
19/19 0s 10ms/step - loss: 5783134208.0000 - mae: 45985.5547 - val_loss: 4851748352.0000 - val_mae: 45536.3398
Epoch 5/200
19/19 0s 8ms/step - loss: 5193511424.0000 - mae: 44859.7969 - val_loss: 4530418176.0000 - val_mae: 45860.7539
Epoch 6/200
19/19 0s 10ms/step - loss: 4047542784.0000 - mae: 41736.9609 - val_loss: 4311923200.0000 - val_mae: 45494.5977
Epoch 7/200
19/19 0s 8ms/step - loss: 3807701248.0000 - mae: 42849.1641 - val_loss: 4130019584.0000 - val_mae: 44322.2734
Epoch 8/200
19/19 0s 9ms/step - loss: 3598571776.0000 - mae: 41892.5312 - val_loss: 3967644672.0000 - val_mae: 42579.2305
Epoch 9/200
19/19 0s 8ms/step - loss: 2874368768.0000 - mae: 38727.9219 - val_loss: 3826490880.0000 - val_mae: 40402.1367
Epoch 10/200
19/19 0s 10ms/step - loss: 3336378112.0000 - mae: 38853.5078 - val_loss: 3870662912.0000 - val_mae: 42148.8203
Epoch 11/200
19/19 0s 8ms/step - loss: 3026695424.0000 - mae: 39462.7188 - val_loss: 3629264384.0000 - val_mae: 38317.2734
Epoch 12/200
19/19 0s 10ms/step - loss: 2728735744.0000 - mae: 37125.0820 - val_loss: 3549077504.0000 - val_mae: 37177.3711
Epoch 13/200
19/19 0s 8ms/step - loss: 2617257216.0000 - mae: 36336.0781 - val_loss: 3471936512.0000 - val_mae: 35744.7383
Epoch 14/200
19/19 0s 8ms/step - loss: 2434131456.0000 - mae: 34110.3516 - val_loss: 34606666624.0000 - val_mae: 35432.2227
Epoch 15/200
19/19 0s 10ms/step - loss: 2315166720.0000 - mae: 33241.9414 - val_loss: 3406520064.0000 - val_mae: 34178.5430
Epoch 16/200
19/19 0s 8ms/step - loss: 2087222656.0000 - mae: 33021.4609 - val_loss: 335557120.0000 - val_mae: 32675.4824
Epoch 17/200
19/19 0s 9ms/step - loss: 1768164224.0000 - mae: 29660.1621 - val_loss: 3518031616.0000 - val_mae: 33554.2109
Epoch 18/200
19/19 0s 8ms/step - loss: 2098392320.0000 - mae: 31794.0273 - val_loss: 3401647360.0000 - val_mae: 31821.2285
Epoch 19/200
19/19 0s 8ms/step - loss: 2258603520.0000 - mae: 30976.7520 - val_loss: 3461771008.0000 - val_mae: 31682.7852
Epoch 20/200
19/19 0s 10ms/step - loss: 1793839360.0000 - mae: 29342.4492 - val_loss: 3520368640.0000 - val_mae: 32540.5859
Epoch 21/200
19/19 0s 8ms/step - loss: 1754913664.0000 - mae: 28727.0488 - val_loss: 3615269888.0000 - val_mae: 31498.9316
Epoch 22/200
19/19 0s 13ms/step - loss: 2112548096.0000 - mae: 29852.8242 - val_loss: 3686983168.0000 - val_mae: 31364.9648
Epoch 23/200
19/19 1s 14ms/step - loss: 1691754368.0000 - mae: 28241.4883 - val_loss: 3660957696.0000 - val_mae: 31371.3145
Epoch 24/200
19/19 0s 15ms/step - loss: 2086262016.0000 - mae: 29826.3926 - val_loss: 3684656640.0000 - val_mae: 31266.9355
Epoch 25/200
19/19 1s 15ms/step - loss: 1467636736.0000 - mae: 26898.6875 - val_loss: 3761703424.0000 - val_mae: 31991.2422
Epoch 26/200
19/19 1s 9ms/step - loss: 1895260160.0000 - mae: 28733.4336 - val_loss: 3867340032.0000 - val_mae: 31270.3672
Epoch 27/200
19/19 0s 8ms/step - loss: 1969007488.0000 - mae: 28344.1250 - val_loss: 3948715008.0000 - val_mae: 31348.9180
Epoch 28/200
19/19 0s 8ms/step - loss: 1693240064.0000 - mae: 28313.6133 - val_loss: 3887248128.0000 - val_mae: 31249.8223
Epoch 29/200
19/19 0s 9ms/step - loss: 1798059648.0000 - mae: 28154.9805 - val_loss: 3878919680.0000 - val_mae: 31199.0234

```

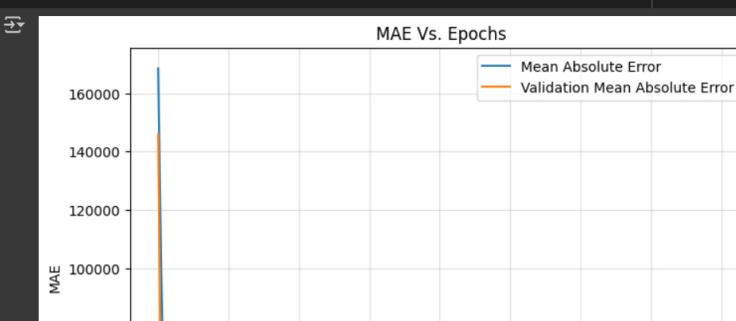
```
0s [100] model.summary()
```

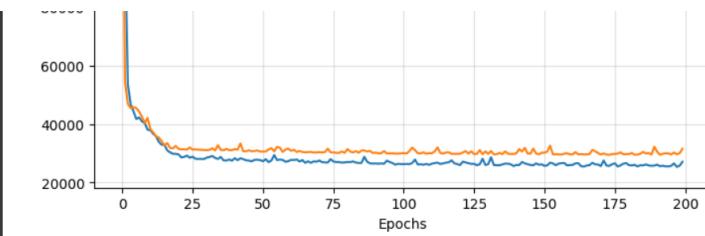
```
→ Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 128)	2,688
dense_13 (Dense)	(None, 256)	33,024
dropout_3 (Dropout)	(None, 256)	0
dense_14 (Dense)	(None, 64)	16,448
dense_15 (Dense)	(None, 1)	65

Total params: 156,077 (612.02 KB)
Trainable params: 52,225 (204.00 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 104,452 (408.02 KB)

```
0s ⏎ plt.figure(figsize=(8,6))
plt.plot(history.history['mae'], label="Mean Absolute Error")
plt.plot(history.history['val_mae'], label="Validation Mean Absolute Error")
plt.title(" MAE Vs. Epochs ")
plt.xlabel("Epochs")
plt.ylabel("MAE")
plt.legend()
plt.grid(alpha=0.40)
plt.show()
```





▼ Predictions by Model

```
[102] x_test = test_df.loc[:,selected_features]
      y_test = predicted['SalePrice']

[103] loss , mae = model.evaluate(x_test,y_test)
      print(f"\nLoss : {loss} , MAE : {mae}")

→ 46/46 ━━━━━━━━ 0s 4ms/step - loss: 3676492544.0000 - mae: 46344.0625
      Loss : 3797937920.0 , MAE : 47864.02734375
```

▼ Evaluation of Model

```
[104] pred = model.predict(x_test)
      score = r2_score(pred,y_test)
      rmse = root_mean_squared_error(pred,y_test)

      print(f"\nScore : {score:.3f}\n")
      print("RMSE : {rmse:.3f}")

→ 46/46 ━━━━━━━━ 0s 4ms/step
      Score : 0.082
      RMSE : 61627.415
```

▼ Fluctuations b/w Actual and Predicted Sale Prices

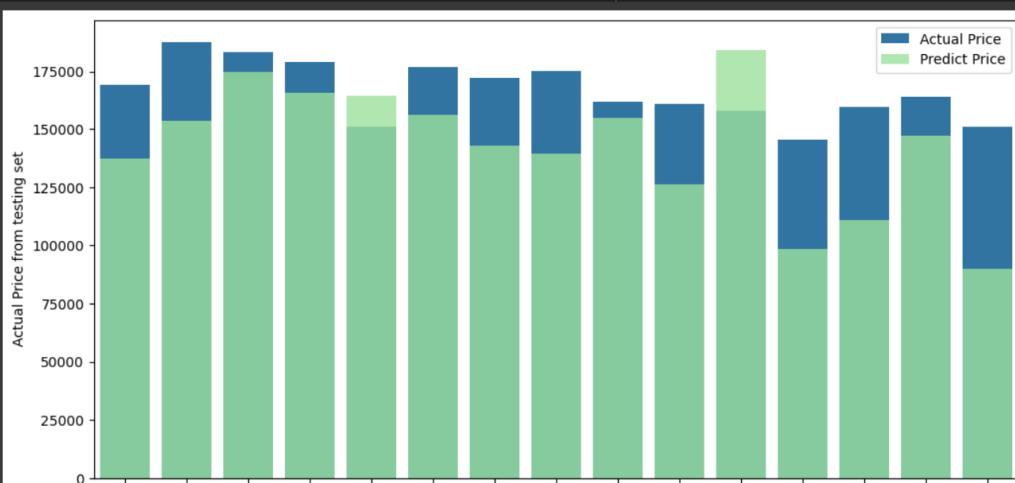
```
[105] Price_prediction = pd.DataFrame()
      Price_prediction['ID'] = predicted['Id']
      Price_prediction['Actual Price from testing set'] = predicted['SalePrice']
      Price_prediction['Predicted price by Model'] = pred
      Price_prediction['Error'] = Price_prediction['Actual Price from testing set'] - Price_prediction['Predicted price by Model']
      Price_prediction.head()
```

ID	Actual Price from testing set	Predicted price by Model	Error
0	1461	169277.0525	137436.046875 31841.005625
1	1462	187758.3940	153319.296875 34430.097125
2	1463	183583.6836	174842.062500 8741.621100
3	1464	179317.4775	165968.093750 13349.383750
4	1465	150730.0800	164413.812500 -13683.732500

Next steps: [Generate code with Price_prediction](#) [View recommended plots](#) [New interactive sheet](#)

▼ Visualization

```
[106] plt.figure(figsize=(12,6))
      sns.barplot(data=Price_prediction[:15], x='ID', y='Actual Price from testing set', label = 'Actual Price')
      sns.barplot(data=Price_prediction[:15], x='ID', y='Predicted price by Model', color='lightgreen',alpha=0.80, label = 'Predict Price')
      plt.show()
```



1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415
ID

0s [106]

Colab paid products - Cancel contracts here

Variables Terminal

✓ 22:41 Python 3