house-price-prediction

September 4, 2023

0.0.1 Importing libraries

[37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[168]:
```
housing_prices_train = pd.read_csv("/kaggle/input/
  ↪house-prices-advanced-regression-techniques/train.csv")
```

[89]:
```
housing_prices_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  MasVnrType     1452 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
```

[90]:
```
housing_prices_train.sample(5)
```

[90]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
675	676	160	RL	24.0	2289	Pave	NaN	Reg	
1138	1139	20	RL	NaN	9819	Pave	NaN	IR1	
1083	1084	20	RL	80.0	8800	Pave	NaN	Reg	
437	438	45	RM	50.0	6000	Pave	NaN	Reg	
454	455	90	RL	63.0	9297	Pave	NaN	Reg	

	LandContour	Utilities	…	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	\
675	Lvl	AllPub	…	0	NaN	NaN	NaN	0	
1138	Lvl	AllPub	…	0	NaN	NaN	NaN	0	
1083	Lvl	AllPub	…	0	NaN	MnPrv	Shed	700	
437	Lvl	AllPub	…	0	NaN	NaN	NaN	0	
454	Lvl	AllPub	…	0	NaN	NaN	NaN	0	

	MoSold	YrSold	SaleType	SaleCondition	SalePrice
675	4	2009	WD	Normal	148500
1138	5	2009	WD	Normal	196000
1083	3	2006	WD	Normal	160000
437	1	2009	WD	Normal	119000
454	7	2006	WD	Family	188000

[5 rows x 81 columns]

[91]:
```
housing_prices_train.drop('Id',inplace=True,axis=1)
```

[92]:
```
num_features = [feature for feature in housing_prices_train.columns if␣
 ↪housing_prices_train[feature].dtype != 'object']
len(num_features)
```

[92]: 37

[93]:
```
cat_features = [feature for feature in housing_prices_train.columns if␣
 ↪housing_prices_train[feature].dtype == 'object']
len(cat_features)
```

[93]: 43

[94]:
```
housing_prices_train[cat_features].nunique().sort_values(ascending=False)
```

[94]:
```
Neighborhood     25
Exterior2nd      16
Exterior1st      15
SaleType          9
Condition1        9
Condition2        8
HouseStyle        8
RoofMatl          8
Functional        7
BsmtFinType2      6
Heating           6
RoofStyle         6
SaleCondition     6
BsmtFinType1      6
GarageType        6
Foundation        6
Electrical        5
FireplaceQu       5
HeatingQC         5
GarageQual        5
GarageCond        5
MSZoning          5
LotConfig         5
ExterCond         5
BldgType          5
BsmtExposure      4
MiscFeature       4
Fence             4
LotShape          4
LandContour       4
BsmtCond          4
KitchenQual       4
MasVnrType        4
ExterQual         4
BsmtQual          4
LandSlope         3
GarageFinish      3
```

```
PavedDrive        3
PoolQC            3
Utilities         2
CentralAir        2
Street            2
Alley             2
dtype: int64
```

[95]:
```
housing_prices_train.isna().sum().sort_values(ascending=False)[:10]
```

[95]:
```
PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
FireplaceQu      690
LotFrontage      259
GarageYrBlt       81
GarageCond        81
GarageType        81
GarageFinish      81
dtype: int64
```

[96]:
```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder

def preprocess_data(df):
    """
    Preprocesses the input DataFrame by imputing missing values and encoding␣
 ↪categorical features.

    Parameters:
    df (pd.DataFrame): Input DataFrame with missing values and categorical␣
 ↪features.

    Returns:
    pd.DataFrame: Preprocessed DataFrame with imputed values and encoded␣
 ↪categorical features.
    """
    # Separate numerical and categorical columns
    numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
    categorical_cols = df.select_dtypes(include=['object']).columns

    # Impute missing values in numerical columns with mean
    imputer = SimpleImputer(strategy='mean')
    df[numerical_cols] = imputer.fit_transform(df[numerical_cols])

    # Encode categorical columns using one-hot encoding
```

```
    encoder = OneHotEncoder(drop='first', sparse=False)
    encoded_categorical = encoder.fit_transform(df[categorical_cols])
    encoded_df = pd.DataFrame(encoded_categorical, columns=encoder.
↪get_feature_names(categorical_cols))

    # Combine numerical and encoded categorical features
    preprocessed_df = pd.concat([df[numerical_cols], encoded_df], axis=1)

    return preprocessed_df
```

[97]:
```
train = preprocess_data(housing_prices_train)
train
```

[97]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	\
0	60.0	65.0	8450.0	7.0	5.0	2003.0	
1	20.0	80.0	9600.0	6.0	8.0	1976.0	
2	60.0	68.0	11250.0	7.0	5.0	2001.0	
3	70.0	60.0	9550.0	7.0	5.0	1915.0	
4	60.0	84.0	14260.0	8.0	5.0	2000.0	
…	…	…	…	…	…	…	
1455	60.0	62.0	7917.0	6.0	5.0	1999.0	
1456	20.0	85.0	13175.0	6.0	6.0	1978.0	
1457	70.0	66.0	9042.0	7.0	9.0	1941.0	
1458	20.0	68.0	9717.0	5.0	6.0	1950.0	
1459	20.0	75.0	9937.0	5.0	6.0	1965.0	

	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	…	SaleType_ConLI	\
0	2003.0	196.0	706.0	0.0	…	0.0	
1	1976.0	0.0	978.0	0.0	…	0.0	
2	2002.0	162.0	486.0	0.0	…	0.0	
3	1970.0	0.0	216.0	0.0	…	0.0	
4	2000.0	350.0	655.0	0.0	…	0.0	
…	…	…	…	…	…	…	
1455	2000.0	0.0	0.0	0.0	…	0.0	
1456	1988.0	119.0	790.0	163.0	…	0.0	
1457	2006.0	0.0	275.0	0.0	…	0.0	
1458	1996.0	0.0	49.0	1029.0	…	0.0	
1459	1965.0	0.0	830.0	290.0	…	0.0	

	SaleType_ConLw	SaleType_New	SaleType_Oth	SaleType_WD	\
0	0.0	0.0	0.0	1.0	
1	0.0	0.0	0.0	1.0	
2	0.0	0.0	0.0	1.0	
3	0.0	0.0	0.0	1.0	
4	0.0	0.0	0.0	1.0	
…	…	…	…	…	
1455	0.0	0.0	0.0	1.0	

```
1456             0.0           0.0           0.0          1.0
1457             0.0           0.0           0.0          1.0
1458             0.0           0.0           0.0          1.0
1459             0.0           0.0           0.0          1.0

      SaleCondition_AdjLand  SaleCondition_Alloca  SaleCondition_Family  \
0                       0.0                   0.0                   0.0
1                       0.0                   0.0                   0.0
2                       0.0                   0.0                   0.0
3                       0.0                   0.0                   0.0
4                       0.0                   0.0                   0.0
…                       …                     …                     …
1455                    0.0                   0.0                   0.0
1456                    0.0                   0.0                   0.0
1457                    0.0                   0.0                   0.0
1458                    0.0                   0.0                   0.0
1459                    0.0                   0.0                   0.0

      SaleCondition_Normal  SaleCondition_Partial
0                      1.0                   0.0
1                      1.0                   0.0
2                      1.0                   0.0
3                      0.0                   0.0
4                      1.0                   0.0
…                      …                     …
1455                   1.0                   0.0
1456                   1.0                   0.0
1457                   1.0                   0.0
1458                   1.0                   0.0
1459                   1.0                   0.0

[1460 rows x 262 columns]
```

---

[12]:
```
plt.figure(figsize=(29,15))

for i, feature in enumerate(num_features):
    plt.subplot(7, 6, i+1)
    sns.histplot(housing_prices_train[feature])
    plt.tight_layout()
```

0.0.2 Simple Linear Regression Baseline

- Feature engineering upto modeling standards with no additional features and transformations.
- Establishing two baselines on:
- Non-Scaled data
- Scaled data

[98]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score,␣
 ↪mean_absolute_percentage_error
```

[99]:
```
y = train.pop('SalePrice')
X = train
```

[100]:
```
RANDOM_STATE = 42
TEST_SIZE = 0.3
```

[101]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y,␣
 ↪random_state=RANDOM_STATE, test_size= TEST_SIZE)
```

[102]:
```
lr = LinearRegression()
```

[103]:
```
lr.fit(X_train, y_train)
lr.get_params()
```

[103]:
```
{'copy_X': True,
 'fit_intercept': True,
```

```
 'n_jobs': None,
 'normalize': 'deprecated',
 'positive': False}
```

[104]:
```
print(f'Training : {lr.score(X_train, y_train)}')
print(f'Testing  : {lr.score(X_test, y_test)}')
```

```
Training : 0.9407472896375583
Testing  : 0.02433703501829121
```

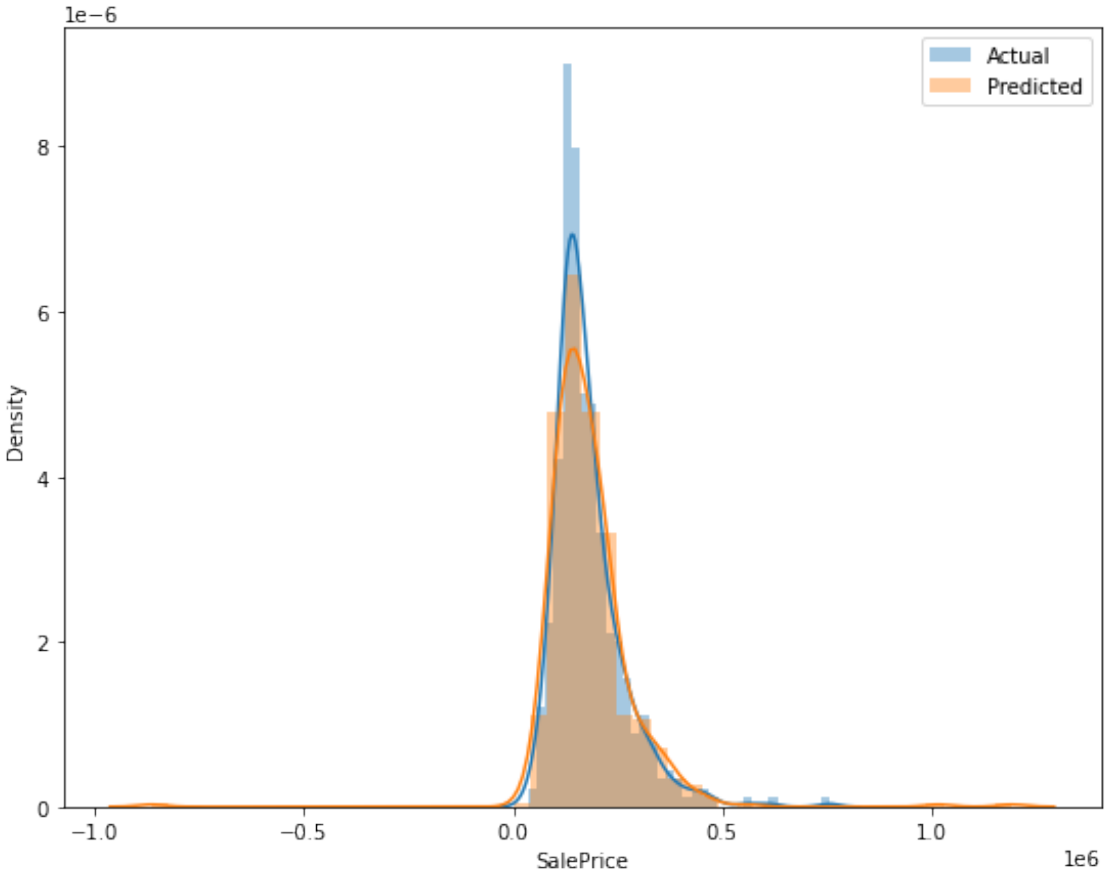
[105]:
```
lr_preds = lr.predict(X_test)
```

[106]:
```
def eval(y_test, preds):
    print(f'MAE : {mean_absolute_error(y_test, preds)}')
    print(f'MSE : {mean_squared_error(y_test, preds)}')
    print(f'R2  :{r2_score(y_test, preds)}')
    print(f'MAPE:{mean_absolute_percentage_error(y_test, preds)}\n')

    plt.figure(figsize=(20,7))
    plt.subplot(1,2,1)
    sns.distplot(y_test, kde=True, label="Actual")
    sns.distplot(preds, kde=True, label="Predicted")
    plt.legend()
```

[107]:
```
eval(y_test, lr_preds)
```

```
MAE : 23911.267430381165
MSE : 6808267648.718407
R2  :0.02433703501829121
MAPE:0.139007233078276
```



0.0.3 Feature Engineering, Modeling

[112]:
```
housing_prices_train[cat_features].nunique().sort_values(ascending=False)
```

[112]:
```
Neighborhood     25
Exterior2nd      16
Exterior1st      15
SaleType          9
Condition1        9
Condition2        8
HouseStyle        8
RoofMatl          8
Functional        7
BsmtFinType2      6
Heating           6
RoofStyle         6
SaleCondition     6
BsmtFinType1      6
GarageType        6
```

```
Foundation        6
Electrical        5
FireplaceQu       5
HeatingQC         5
GarageQual        5
GarageCond        5
MSZoning          5
LotConfig         5
ExterCond         5
BldgType          5
BsmtExposure      4
MiscFeature       4
Fence             4
LotShape          4
LandContour       4
BsmtCond          4
KitchenQual       4
MasVnrType        4
ExterQual         4
BsmtQual          4
LandSlope         3
GarageFinish      3
PavedDrive        3
PoolQC            3
Utilities         2
CentralAir        2
Street            2
Alley             2
dtype: int64
```

[113]:
```
housing_prices_train[['Neighborhood','Exterior1st','Exterior2nd']]
```

[113]:

	Neighborhood	Exterior1st	Exterior2nd
0	CollgCr	VinylSd	VinylSd
1	Veenker	MetalSd	MetalSd
2	CollgCr	VinylSd	VinylSd
3	Crawfor	Wd Sdng	Wd Shng
4	NoRidge	VinylSd	VinylSd
…	…	…	…
1455	Gilbert	VinylSd	VinylSd
1456	NWAmes	Plywood	Plywood
1457	Crawfor	CemntBd	CmentBd
1458	NAmes	MetalSd	MetalSd
1459	Edwards	HdBoard	HdBoard

[1460 rows x 3 columns]

[119]:
```
ex = housing_prices_train['Exterior1st'] +" "+␣
 ↪housing_prices_train['Exterior2nd']
ex
```

[119]:
```
0       VinylSd VinylSd
1       MetalSd MetalSd
2       VinylSd VinylSd
3       Wd Sdng Wd Shng
4       VinylSd VinylSd
             …
1455    VinylSd VinylSd
1456    Plywood Plywood
1457    CemntBd CmentBd
1458    MetalSd MetalSd
1459    HdBoard HdBoard
Length: 1460, dtype: object
```

[121]:
```
housing_prices_train['Exterior'] = housing_prices_train['Exterior1st'] +" "+␣
 ↪housing_prices_train['Exterior2nd']
```

[122]:
```
housing_prices_train
```

[122]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	60.0	RL	65.0	8450.0	Pave	NaN	Reg	
1	20.0	RL	80.0	9600.0	Pave	NaN	Reg	
2	60.0	RL	68.0	11250.0	Pave	NaN	IR1	
3	70.0	RL	60.0	9550.0	Pave	NaN	IR1	
4	60.0	RL	84.0	14260.0	Pave	NaN	IR1	
…	…	…	…	…	…	…	…	
1455	60.0	RL	62.0	7917.0	Pave	NaN	Reg	
1456	20.0	RL	85.0	13175.0	Pave	NaN	Reg	
1457	70.0	RL	66.0	9042.0	Pave	NaN	Reg	
1458	20.0	RL	68.0	9717.0	Pave	NaN	Reg	
1459	20.0	RL	75.0	9937.0	Pave	NaN	Reg	

	LandContour	Utilities	LotConfig	…	PoolQC	Fence	MiscFeature	MiscVal	\
0	Lvl	AllPub	Inside	…	NaN	NaN	NaN	0.0	
1	Lvl	AllPub	FR2	…	NaN	NaN	NaN	0.0	
2	Lvl	AllPub	Inside	…	NaN	NaN	NaN	0.0	
3	Lvl	AllPub	Corner	…	NaN	NaN	NaN	0.0	
4	Lvl	AllPub	FR2	…	NaN	NaN	NaN	0.0	
…	…	…	…	…	…	…	…	…	
1455	Lvl	AllPub	Inside	…	NaN	NaN	NaN	0.0	
1456	Lvl	AllPub	Inside	…	NaN	MnPrv	NaN	0.0	
1457	Lvl	AllPub	Inside	…	NaN	GdPrv	Shed	2500.0	
1458	Lvl	AllPub	Inside	…	NaN	NaN	NaN	0.0	
1459	Lvl	AllPub	Inside	…	NaN	NaN	NaN	0.0	

```
     MoSold  YrSold  SaleType  SaleCondition  SalePrice         Exterior
0       2.0  2008.0        WD         Normal   208500.0  VinylSd VinylSd
1       5.0  2007.0        WD         Normal   181500.0  MetalSd MetalSd
2       9.0  2008.0        WD         Normal   223500.0  VinylSd VinylSd
3       2.0  2006.0        WD        Abnorml   140000.0  Wd Sdng Wd Shng
4      12.0  2008.0        WD         Normal   250000.0  VinylSd VinylSd
…       …       …        …              …         …              …
1455    8.0  2007.0        WD         Normal   175000.0  VinylSd VinylSd
1456    2.0  2010.0        WD         Normal   210000.0  Plywood Plywood
1457    5.0  2010.0        WD         Normal   266500.0  CemntBd CmentBd
1458    4.0  2010.0        WD         Normal   142125.0  MetalSd MetalSd
1459    6.0  2008.0        WD         Normal   147500.0  HdBoard HdBoard

[1460 rows x 81 columns]
```

[120]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[123]:
```
housing_prices_train['Exterior'] = le.
 ↪fit_transform(housing_prices_train['Exterior'])
housing_prices_train['Neighborhood'] = le.
 ↪fit_transform(housing_prices_train['Neighborhood'])
```

[127]:
```
housing_prices_train.drop(['Exterior1st','Exterior2nd'], inplace=True, axis=1)
```

[130]:
```
y = housing_prices_train.pop('SalePrice')
```

[136]:
```
new_cols = housing_prices_train.select_dtypes(include='object').columns
```

[137]:
```
for feature in new_cols:
    housing_prices_train[feature] = le.
 ↪fit_transform(housing_prices_train[feature])
```

[139]:
```
housing_prices_train.head()
```

[139]:
```
   MSSubClass  MSZoning  LotFrontage  LotArea  Street  Alley  LotShape  \
0        60.0         3         65.0   8450.0       1      2         3
1        20.0         3         80.0   9600.0       1      2         3
2        60.0         3         68.0  11250.0       1      2         0
3        70.0         3         60.0   9550.0       1      2         0
4        60.0         3         84.0  14260.0       1      2         0

   LandContour  Utilities  LotConfig  …  PoolArea  PoolQC  Fence  \
0            3          0          4  …       0.0       3      4
1            3          0          2  …       0.0       3      4
2            3          0          4  …       0.0       3      4
```

```
3            3           0           0  …        0.0        3       4
4            3           0           2  …        0.0        3       4

   MiscFeature  MiscVal  MoSold  YrSold  SaleType  SaleCondition  Exterior
0            4      0.0     2.0  2008.0         8              4        48
1            4      0.0     5.0  2007.0         8              4        27
2            4      0.0     9.0  2008.0         8              4        48
3            4      0.0     2.0  2006.0         8              0        61
4            4      0.0    12.0  2008.0         8              4        48

[5 rows x 78 columns]
```

[143]:
```
train1 = housing_prices_train.copy()
```

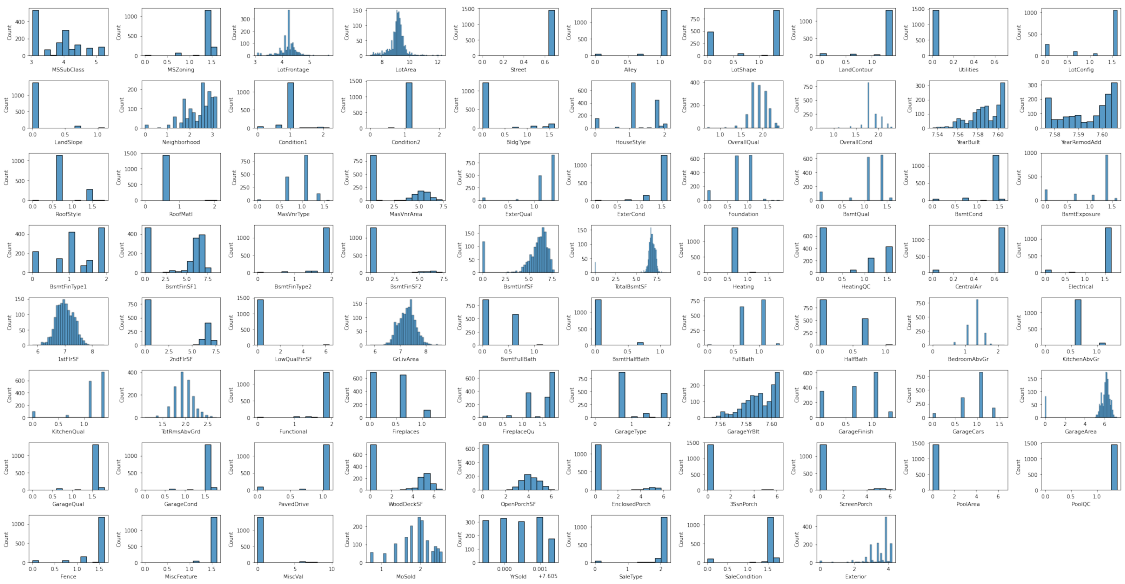
[147]:
```
for feature in housing_prices_train.columns:
    train1[feature] = np.log1p(train1[feature]+0.001)
```

[152]:
```
train_features = train1.columns
len(train_features)
```

[152]:
```
78
```

[153]:
```
plt.figure(figsize=(29,15))

for i, feature in enumerate(train_features):
    plt.subplot(8, 10, i+1)
    sns.histplot(train1[feature])
    plt.tight_layout()
```



0.0.4 Modeling

[154]:
```
X_train, X_test, y_train, y_test = train_test_split(train1, y,␣
↪random_state=RANDOM_STATE, test_size= TEST_SIZE)
```

[155]:
```
lr1 = LinearRegression()
```

[156]:
```
lr1.fit(X_train, y_train)
```

[156]:
```
LinearRegression()
```

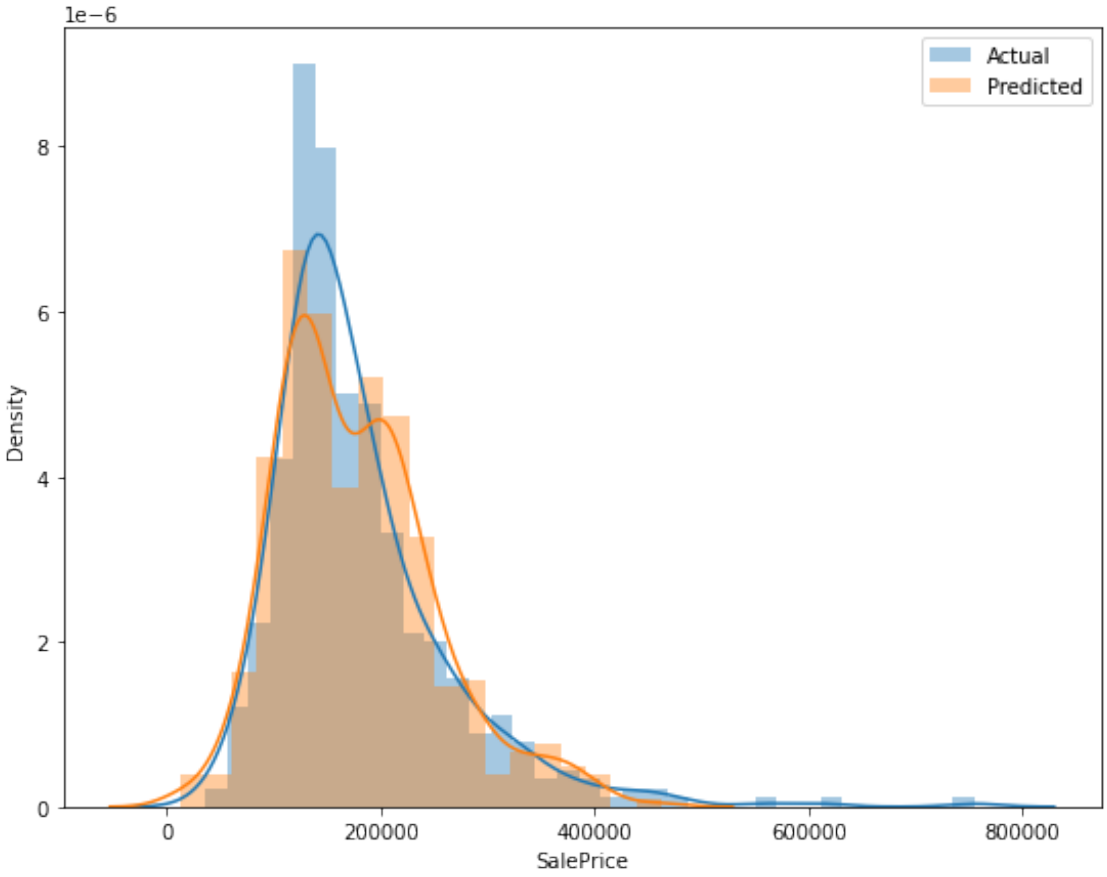
[157]:
```
lr1_preds = lr1.predict(X_test)
```

[160]:
```
print(f'Training : {lr1.score(X_train, y_train)}')
print(f'Testing  : {lr1.score(X_test, y_test)}')
```

```
Training : 0.8498195258132317
Testing  : 0.8277204449876873
```

[158]:
```
eval(y_test, lr1_preds)
```

```
MAE : 22055.137207967426
MSE : 1202182887.9688194
R2  :0.8277204449876873
MAPE:0.13120303210320716
```



Better Training and Testing score

[163]:
```
from catboost import CatBoostRegressor
cbr = CatBoostRegressor(verbose =200)
```

[164]:
```
cbr.fit(X_train, y_train)
```

```
Learning rate set to 0.041084
0:      learn: 75611.7193828    total: 60.3ms   remaining: 1m
200:    learn: 17209.2709686    total: 636ms    remaining: 2.53s
400:    learn: 11500.1887605    total: 1.2s     remaining: 1.79s
600:    learn: 8386.5716567     total: 1.77s    remaining: 1.18s
800:    learn: 6514.3767368     total: 2.34s    remaining: 582ms
999:    learn: 5219.5944478     total: 2.92s    remaining: 0us
```

[164]:
```
<catboost.core.CatBoostRegressor at 0x7f87da675d10>
```

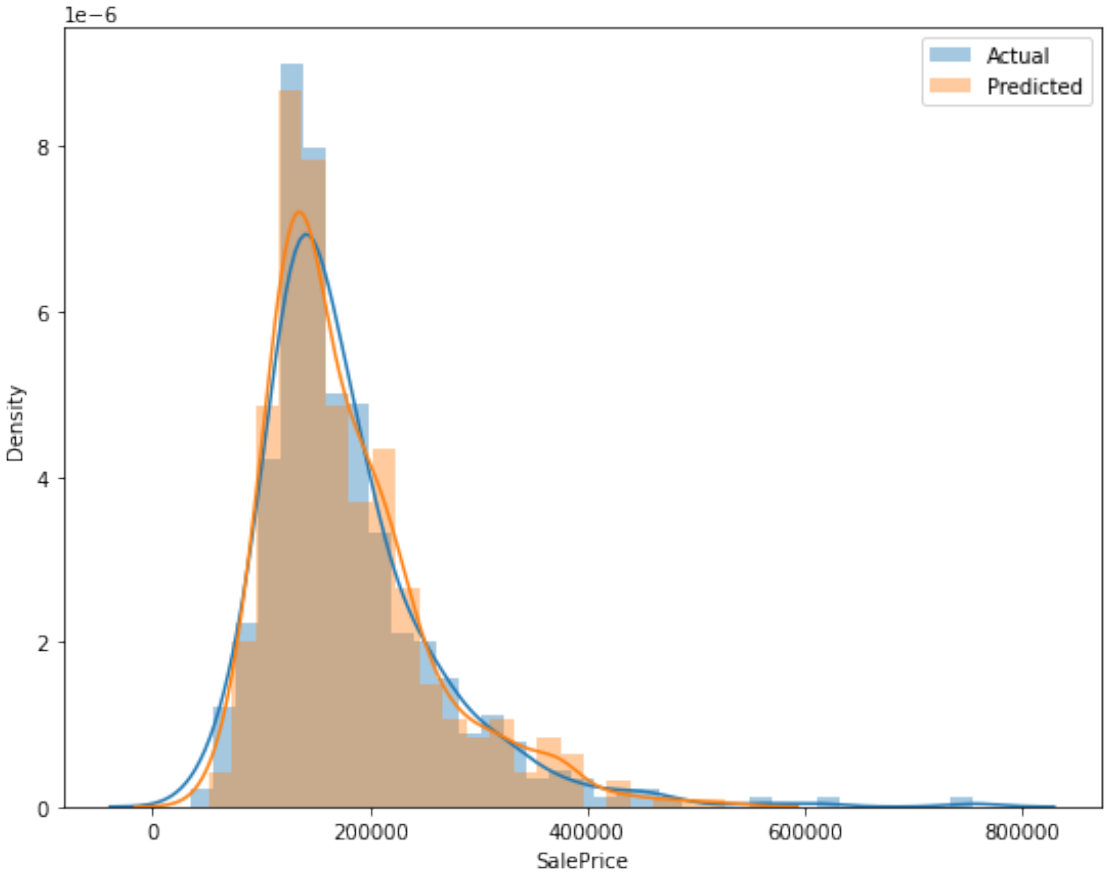
[165]:
```
cbr_pred = cbr.predict(X_test)
```

[166]:
```
print(f'Training : {cbr.score(X_train, y_train)}')
print(f'Testing  : {cbr.score(X_test, y_test)}')
```

```
Training : 0.9954733198337394
Testing  : 0.9119932294410434
```

[167]:
```
eval(y_test, cbr_pred)
```

```
MAE : 14819.4387589486
MSE : 614119496.557872
R2  :0.9119932294410434
MAPE:0.08801455544875962
```



0.0.5 Even better preds with CatBoost Regressor