In [1]:
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import joblib
```

In [2]:
```python
train = pd.read_csv('/kaggle/input/house-prices-advanced-regression-techniques/train.csv')
test = pd.read_csv('/kaggle/input/house-prices-advanced-regression-techniques/test.csv')
```

In [3]:
```python
train.columns
```

Out[3]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [4]:
```python
test.columns
```

Out[4]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition'],
      dtype='object')
```

In [5]:
```python
train
```

Out[5]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	M
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

1460 rows × 81 columns

In [6]:
```python
train.isna().sum()
```

Out[6]:
```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
               ...
MoSold            0
YrSold            0
SaleType          0
SaleCondition     0
SalePrice         0
Length: 81, dtype: int64
```

In [7]:
```python
train.describe()
```

Out[7]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDe
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	...	1460.00
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	94.24
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	...	125.33
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.00
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.00
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	...	0.00
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712.250000	...	168.00
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.00

8 rows × 38 columns

In [8]:
```python
train = train.ffill()
test = test.ffill()
```

In [9]:
```python
train
```

Out[9]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	M
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	Pave	Reg	Lvl	AllPub	...	0	Gd	GdWo	TenC	0	
1456	1457	20	RL	85.0	13175	Pave	Pave	Reg	Lvl	AllPub	...	0	Gd	MnPrv	TenC	0	
1457	1458	70	RL	66.0	9042	Pave	Pave	Reg	Lvl	AllPub	...	0	Gd	GdPrv	Shed	2500	
1458	1459	20	RL	68.0	9717	Pave	Pave	Reg	Lvl	AllPub	...	0	Gd	GdPrv	Shed	0	
1459	1460	20	RL	75.0	9937	Pave	Pave	Reg	Lvl	AllPub	...	0	Gd	GdPrv	Shed	0	

1460 rows × 81 columns

In [10]:
```python
train.describe()
```

Out[10]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDe
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	...	1460.00
mean	730.500000	56.897260	70.104795	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.492466	443.639726	...	94.24
std	421.610009	42.300571	23.846996	9981.264932	1.382997	1.112799	30.202904	20.645407	180.795612	456.098091	...	125.33
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.00
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.00
50%	730.500000	50.000000	70.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	...	0.00
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	165.250000	712.250000	...	168.00
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.00

8 rows × 38 columns

In [11]:
```python
train.isna().sum()
```

Out[11]:
```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 81, dtype: int64
```

In [12]:
```python
train = train.drop("Id",axis=1)
```

In [13]:
```python
# train = pd.get_dummies(train)
# test = pd.get_dummies(test)
```

In [14]:
```python
train
```

Out[14]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVa
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	(
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	...	0	NaN	NaN	NaN	(
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	NaN	NaN	NaN	(
3	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	...	0	NaN	NaN	NaN	(
4	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	...	0	NaN	NaN	NaN	(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
1455	60	RL	62.0	7917	Pave	Pave	Reg	Lvl	AllPub	Inside	...	0	Gd	GdWo	TenC	(
1456	20	RL	85.0	13175	Pave	Pave	Reg	Lvl	AllPub	Inside	...	0	Gd	MnPrv	TenC	(
1457	70	RL	66.0	9042	Pave	Pave	Reg	Lvl	AllPub	Inside	...	0	Gd	GdPrv	Shed	250(
1458	20	RL	68.0	9717	Pave	Pave	Reg	Lvl	AllPub	Inside	...	0	Gd	GdPrv	Shed	(
1459	20	RL	75.0	9937	Pave	Pave	Reg	Lvl	AllPub	Inside	...	0	Gd	GdPrv	Shed	(

1460 rows × 80 columns

In [15]:
```python
columns_to_encode = ['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', 'GarageType', 'GarageFinish', 'GarageQual',
                     'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
                     'SaleType', 'SaleCondition']

# for column in columns_to_encode:
#     label_encoder = LabelEncoder()
#     train[column] = label_encoder.fit_transform(train[column])

train = train.drop(columns=columns_to_encode)
test = test.drop(columns=columns_to_encode)
```

In [16]:
```python
train
```

Out[16]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	...	WoodDeckSF	OpenPorcl
0	60	65.0	8450	7	5	2003	2003	196.0	706	0	...	0	
1	20	80.0	9600	6	8	1976	1976	0.0	978	0	...	298	
2	60	68.0	11250	7	5	2001	2002	162.0	486	0	...	0	
3	70	60.0	9550	7	5	1915	1970	0.0	216	0	...	0	
4	60	84.0	14260	8	5	2000	2000	350.0	655	0	...	192	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	6	5	1999	2000	0.0	0	0	...	0	
1456	20	85.0	13175	6	6	1978	1988	119.0	790	163	...	349	
1457	70	66.0	9042	7	9	1941	2006	0.0	275	0	...	0	
1458	20	68.0	9717	5	6	1950	1996	0.0	49	1029	...	366	
1459	20	75.0	9937	5	6	1965	1965	0.0	830	290	...	736	

1460 rows × 37 columns

In [17]:
```python
X=train.drop(columns='SalePrice')
y=train['SalePrice']
```

In [18]:
```python
X
```

Out[18]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	...	GarageArea	WoodDeckS
0	60	65.0	8450	7	5	2003	2003	196.0	706	0	...	548	
1	20	80.0	9600	6	8	1976	1976	0.0	978	0	...	460	29
2	60	68.0	11250	7	5	2001	2002	162.0	486	0	...	608	
3	70	60.0	9550	7	5	1915	1970	0.0	216	0	...	642	
4	60	84.0	14260	8	5	2000	2000	350.0	655	0	...	836	19
...	...	...	...	...	...	...	...	...	...	...	...	...	.
1455	60	62.0	7917	6	5	1999	2000	0.0	0	0	...	460	
1456	20	85.0	13175	6	6	1978	1988	119.0	790	163	...	500	34
1457	70	66.0	9042	7	9	1941	2006	0.0	275	0	...	252	
1458	20	68.0	9717	5	6	1950	1996	0.0	49	1029	...	240	36
1459	20	75.0	9937	5	6	1965	1965	0.0	830	290	...	276	73

1460 rows × 36 columns

In [19]:
```python
y
```

Out[19]:
```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

In [20]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [21]:
```python
model = LinearRegression()
model.fit(X_train,y_train)
```

Out[21]:
```
▾ LinearRegression
LinearRegression()
```

In [22]:
```python
y_pred=model.predict(X_train)
```

In [23]:
```python
model.score(X_train,y_train)
```

Out[23]:
```
0.8015281494106271
```

In [24]:
```python
model.score(X_test,y_test)
```

Out[24]:
```
0.8218984206447474
```

In [25]:
```python
model.intercept_
```

Out[25]:
```
-815908.849522751
```

In [26]:
```python
model.coef_
```

Out[26]:
```
array([-2.28758022e+02, -1.30274291e+02,  4.41475081e-01,  1.84916371e+04,
        3.17649210e+03,  2.69678017e+02,  1.81435648e+02,  2.11078675e+01,
        8.16044018e+00, -3.21945401e+00, -1.68887757e+00,  3.25210859e+00,
        1.06037439e+01,  1.19683935e+01,  9.75414631e+00,  3.23262838e+01,
        1.20663858e+04, -5.51389369e+02,  4.11521605e+03, -1.91986453e+03,
       -9.32378685e+03, -7.97990985e+03,  5.35009480e+03,  5.17524789e+03,
        8.99565969e+01,  1.53206678e+04, -9.25095806e+00,  1.90169178e+01,
       -1.94315073e+01,  2.43639078e+00,  4.41582379e+01,  7.16368010e+01,
       -1.19194250e+01, -2.38295259e+00,  3.22607017e+01, -1.49040988e+02])
```

In [27]:
```python
SalePrice = model.predict(test.drop(columns='Id'))
```

In [28]:
```python
submission_df = pd.DataFrame({"Id": test['Id'], "SalePrice": SalePrice})
```

In [29]:
```python
submission_df.to_csv("LinearRegression.csv", index=False)
```

In [30]:
```python
data = pd.read_csv("LinearRegression.csv")
data.head()
```

Out[30]:

	Id	SalePrice
0	1461	115699.581663
1	1462	123576.160107
2	1463	173714.288809
3	1464	199306.794830
4	1465	197913.200367

In [31]:
```
joblib.dump(model, 'House_price_linear_regression_model.pkl')
```

Out[31]:
```
['House_price_linear_regression_model.pkl']
```