0 1 60 RL 65.0 1 2 20 RL 80.0 2 3 60 RL 68.0 3 4 70 RL 60.0 4 5 60 RL 84.0 1455 1456 60 RL 62.0	8450 Pave NaN Reg LvI AliPub 0 NaN NaN Normal 208500 9600 Pave NaN Reg LvI AliPub 0 NaN NaN NaN 0 5 2007 WD Normal 181500 11250 Pave NaN IR1 LvI AliPub 0 NaN
1456 1457 20 RL 85.0 1457 1458 70 RL 66.0 1458 1459 20 RL 68.0 1459 1460 20 RL 75.0 1460 rows × 81 columns	13175 Pave NaN Reg Lvi AliPub 0 NaN MnPrv NaN 0 2 2010 WD Normal 210000 9042 Pave NaN Reg Lvi AliPub 0 NaN GdPrv Shed 2500 5 2010 WD Normal 266500 9717 Pave NaN Reg Lvi AliPub 0 NaN NaN NaN NaN 0 4 2010 WD Normal 142125 9937 Pave NaN Reg Lvi AliPub 0 NaN NaN NaN NaN NaN 0 6 2008 WD Normal 147500
0 1 60 RL 65.0 8456 1 2 20 RL 80.0 9606 2 3 60 RL 68.0 11256 3 4 70 RL 60.0 9556	Pave NaN Reg Lvl AllPub 0 NaN NaN NaN 0 5 2007 WD Normal 181500 Pave NaN IR1 Lvl AllPub 0 NaN NaN NaN 0 9 2008 WD Normal 223500
<pre>In []: In [4]: df.isnull().sum() Out[4]: Id</pre>	
YrSold 0 SaleType 0 SaleCondition 0 SalePrice 0 Length: 81, dtype: int64 In [5]: df['LotFrontage']=df['LotFrontage'].fillna In [6]: df.drop(['Alley'] ,axis=1,inplace=True) In [7]: df['BsmtCond']=df['BsmtCond'].fillna(df['BsmtQual']=df['BsmtQual'].fillna(df['BsmtQual']).fi	smtCond'].mode()[0])
<pre>In [8]: df['FireplaceQu']=df['FireplaceQu'].fillna</pre>	<pre>[['GarageType'].mode()[0]) ide() ide(df['GarageFinish'].mode()[0]) if['GarageCond'].mode()[0]) if['GarageQual'].mode()[0]) ixis=1,inplace=True) if['MasVnrType'].mode()[0])</pre>
<pre>df['MasVnrArea']=df['MasVnrArea'].fillna(dr In [13]: sns.heatmap(df.isnull(),yticklabels=False, Out[13]: <axes:></axes:></pre>	['MasVnrArea'].mode()[0])
In [14]: Condition LotShape RearBuilt RearBuilt RearBuilt RearBuilt RearBuilt RearIng RearIng RearIng LotShape	In (df ['BsmtExposure'].mode()[0]) Share False, cmap='YlGnBu')
ld - LotShape - LotShape - LotConfig - Condition1 - HouseStyle - YearBuilt - RoofMatl - MasVnrType - ExterCond - BsmtCond - BsmtFinSF1 - HeatingQC - 1stFlrSF - GrLivArea - FullBath -	
<pre>In [15]: df['BsmtFinType2']=df['BsmtFinType2'].fille</pre>	na (df['BsmtFinType2'].mode()[0]) Street LotShape LandContour Utilities LotConfig EnclosedPorch 3SsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold SaleType SaleCondition SalePrice Pave Reg Lvl AllPub Inside 0 0 0 0 0 0 2 2008 WD Normal 208500
2 3 60 RL 68.0 11250 3 4 70 RL 60.0 9550 4 5 60 RL 84.0 14260 5 rows × 76 columns	Pave IR1 Lvl AllPub Inside 0 0 0 0 9 2008 WD Normal 223500 Pave IR1 Lvl AllPub Corner 272 0 0 0 0 Abnorml 140000
'RoofStyle', 'RoofMatl', 'Exterior1s' 'CentralAir', 'Electrical', 'KitchenQual', 'Funct' 'FireplaceQu', 'GarageType', 'Garage len(columns) Out[16]: 42 In [17]: def category_onehot_multcols(multcolumns): df_final=final_df	L','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinType2', L','Exterior2nd','MasVnrType','Heating','HeatingQC',
<pre>for fields in multcolumns: print(fields) df1=pd.get_dummies(final_df[fields]) final_df.drop([fields],axis=1,inplaif i==0: df_final=df1.copy() else: df_final=pd.concat([df_final,dri=i+1])</pre>	ace=True)
<pre>df_final=pd.concat([final_df,df_final] return df_final main_df=df.copy() ## Combine Test Data test_df=pd.read_csv('formulatedtest.csv') test_df.shape test_df.head() df.info()</pre>	axis=1)
<pre><class 'pandas.core.frame.dataframe'=""> Index: 1422 entries, 0 to 1459 Data columns (total 76 columns): # Column</class></pre>	
8 Utilities 1422 non-null object 9 LotConfig 1422 non-null object 10 LandSlope 1422 non-null object 11 Neighborhood 1422 non-null object 12 Condition1 1422 non-null object 13 Condition2 1422 non-null object 14 BldgType 1422 non-null object 15 HouseStyle 1422 non-null object 16 OverallQual 1422 non-null int64 17 OverallCond 1422 non-null int64 18 YearBuilt 1422 non-null int64 19 YearRemodAdd 1422 non-null int64 20 RoofStyle 1422 non-null object	
21 RoofMatl 1422 non-null object 22 Exterior1st 1422 non-null object 23 Exterior2nd 1422 non-null object 24 MasVnrType 1422 non-null object 25 MasVnrArea 1422 non-null float64 26 ExterQual 1422 non-null object 27 ExterCond 1422 non-null object 28 Foundation 1422 non-null object 29 BsmtQual 1422 non-null object 30 BsmtCond 1422 non-null object 31 BsmtExposure 1422 non-null object 32 BsmtFinType1 1422 non-null object 33 BsmtFinSF1 1422 non-null int64	
34 BsmtFinType2 1422 non-null object 35 BsmtFinSF2 1422 non-null int64 36 BsmtUnfSF 1422 non-null int64 37 TotalBsmtSF 1422 non-null int64 38 Heating 1422 non-null object 39 HeatingQC 1422 non-null object 40 CentralAir 1422 non-null object 41 Electrical 1422 non-null object 42 1stF1rSF 1422 non-null int64 43 2ndF1rSF 1422 non-null int64 44 LowQualFinSF 1422 non-null int64 45 GrLivArea 1422 non-null int64 46 BsmtFullBath 1422 non-null int64	
47 BsmtHalfBath 1422 non-null int64 48 FullBath 1422 non-null int64 49 HalfBath 1422 non-null int64 50 BedroomAbvGr 1422 non-null int64 51 KitchenAbvGr 1422 non-null int64 52 KitchenQual 1422 non-null object 53 TotRmsAbvGrd 1422 non-null int64 54 Functional 1422 non-null object 55 Fireplaces 1422 non-null int64 56 FireplaceQu 1422 non-null object 57 GarageType 1422 non-null object 58 GarageFinish 1422 non-null object 59 GarageCars 1422 non-null int64	
60 GarageArea 1422 non-null int64 61 GarageQual 1422 non-null object 62 GarageCond 1422 non-null object 63 PavedDrive 1422 non-null object 64 WoodDeckSF 1422 non-null int64 65 OpenPorchSF 1422 non-null int64 66 EnclosedPorch 1422 non-null int64 67 3SsnPorch 1422 non-null int64 68 ScreenPorch 1422 non-null int64 69 PoolArea 1422 non-null int64 70 MiscVal 1422 non-null int64 71 MoSold 1422 non-null int64 72 YrSold 1422 non-null int64	
73 SaleType 1422 non-null object 74 SaleCondition 1422 non-null object 75 SalePrice 1422 non-null int64 dtypes: float64(2), int64(35), object(39) memory usage: 855.4+ KB In [18]: final_df=pd.concat([df,test_df],axis=0,sor* In [19]: final_df['SalePrice'] Out[19]: 0 208500.0 1 181500.0 2 223500.0	=False)
3 140000.0 4 250000.0 1454 NaN 1455 NaN 1456 NaN 1457 NaN 1458 NaN Name: SalePrice, Length: 2881, dtype: floa	:64
1 9600 2 11250 3 9550 4 14260 1454 1936 1455 1894 1456 20000 1457 10441 1458 9627 Name: LotArea, Length: 2881, dtype: int64 In [21]: final_df=category_onehot_multcols(columns)	
MSZoning Street LotArea LandContour Utilities LotConfig LandSlope Neighborhood Condition2 BldgType Condition1 HouseStyle SaleType	
SaleCondition ExterCond ExterQual Foundation BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 RoofStyle RoofMatl Exterior1st Exterior2nd	
MasVnrType Heating HeatingQC CentralAir Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish GarageCond PavedDrive MasVnrTypr	
Alley LotShape In [22]: final_df.shape Out[22]: (2881, 2177) In [23]: final_df = final_df.loc[:,~final_df.columns In [24]: final_df.shape Out[24]: (2881, 2114)	<pre>duplicated()]</pre>
<pre>In [25]: final_df.shape Out[25]: (2881, 2114) In [26]: df.info()</pre>	
1 MSSubClass 1422 non-null int64 2 MSZoning 1422 non-null object 3 LotFrontage 1422 non-null float64 4 LotArea 1422 non-null int64 5 Street 1422 non-null object 6 LotShape 1422 non-null object 7 LandContour 1422 non-null object 8 Utilities 1422 non-null object 9 LotConfig 1422 non-null object 10 LandSlope 1422 non-null object 11 Neighborhood 1422 non-null object 12 Condition1 1422 non-null object 13 Condition2 1422 non-null object 14 BldgType 1422 non-null object	
15 HouseStyle 1422 non-null object 16 OverallQual 1422 non-null int64 17 OverallCond 1422 non-null int64 18 YearBuilt 1422 non-null int64 19 YearRemodAdd 1422 non-null int64 20 RoofStyle 1422 non-null object 21 RoofMatl 1422 non-null object 22 Exterior1st 1422 non-null object 23 Exterior2nd 1422 non-null object 24 MasVnrType 1422 non-null object 25 MasVnrArea 1422 non-null float64 26 ExterQual 1422 non-null object 27 ExterCond 1422 non-null object	
28 Foundation 1422 non-null object 29 BsmtQual 1422 non-null object 30 BsmtCond 1422 non-null object 31 BsmtExposure 1422 non-null object 32 BsmtFinType1 1422 non-null object 33 BsmtFinSF1 1422 non-null int64 34 BsmtFinType2 1422 non-null object 35 BsmtFinSF2 1422 non-null int64 36 BsmtUnfSF 1422 non-null int64 37 TotalBsmtSF 1422 non-null int64 38 Heating 1422 non-null object 39 HeatingQC 1422 non-null object 40 CentralAir 1422 non-null object	
41 Electrical 1422 non-null object 42 1stFlrSF 1422 non-null int64 43 2ndFlrSF 1422 non-null int64 44 LowQualFinSF 1422 non-null int64 45 GrLivArea 1422 non-null int64 46 BsmtFullBath 1422 non-null int64 47 BsmtHalfBath 1422 non-null int64 48 FullBath 1422 non-null int64 49 HalfBath 1422 non-null int64 50 BedroomAbvGr 1422 non-null int64 51 KitchenAbvGr 1422 non-null int64 52 KitchenQual 1422 non-null object 53 TotRmsAbvGrd 1422 non-null int64	
54 Functional 1422 non-null object 55 Fireplaces 1422 non-null int64 56 FireplaceQu 1422 non-null object 57 GarageType 1422 non-null object 58 GarageFinish 1422 non-null int64 59 GarageCars 1422 non-null int64 60 GarageArea 1422 non-null int64 61 GarageQual 1422 non-null object 62 GarageCond 1422 non-null object 63 PavedDrive 1422 non-null object 64 WoodDeckSF 1422 non-null int64 65 OpenPorchSF 1422 non-null int64 66 EnclosedPorch 1422 non-null int64	
67 3SsnPorch 1422 non-null int64 68 ScreenPorch 1422 non-null int64 69 PoolArea 1422 non-null int64 70 MiscVal 1422 non-null int64 71 MoSold 1422 non-null int64 72 YrSold 1422 non-null int64 73 SaleType 1422 non-null object 74 SaleCondition 1422 non-null object 75 SalePrice 1422 non-null int64 dtypes: float64(2), int64(35), object(39) memory usage: 855.4+ KB In [27]: final_df.info()	
<pre><class 'pandas.core.frame.dataframe'=""> Index: 2881 entries, 0 to 1458 Columns: 2114 entries, Id to Reg dtypes: bool(2078), float64(12), int64(24) memory usage: 6.5 MB In [28]: final_df.info() <class 'pandas.core.frame.dataframe'=""> Index: 2881 entries, 0 to 1458 Columns: 2114 entries, Id to Reg dtypes: bool(2078), float64(12), int64(24) memory usage: 6.5 MB</class></class></pre>	
0 1.0 60 65.0 7 1 2.0 20 80.0 6 2 3.0 60 68.0 7 3 4.0 70 60.0 7	valiCond YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 Attchd Basment BuiltIn CarPort Detchd RFn P IR2 IR3 Reg 5 2003 196.0 706.0 0.0 True False
<pre>4 5.0 60 84.0 8 5 rows × 2114 columns In [30]: final_df =final_df.loc[:,~final_df.columns In [31]: final_df.shape Out[31]: (2881, 2114) In [32]: final_df['SalePrice']</pre>	5 2000 2000 350.0 655.0 0.0 True False False False True False
Out[32]: 0	t64
<pre>In [33]: df_Train=final_df.iloc[:1422,:] df_Test=final_df.iloc[1422:,:] df_Train.head() df_Test['SalePrice'] Out[33]: 0</pre>	
1456 NaN 1457 NaN 1458 NaN Name: SalePrice, Length: 1459, dtype: floa In [34]: X_train=df_Train.drop(['SalePrice'], axis=1 y_train=df_Train['SalePrice'] In [35]: import xgboost classifier=xgboost.XGBRegressor() import xgboost regressor=xgboost.XGBRegressor()	
<pre>In [36]: booster=['gbtree', 'gblinear'] base_score=[0.25,0.5,0.75,1] In [37]: n_estimators = [100, 500, 900, 1100, 1500] max_depth = [2, 3, 5, 10, 15] booster=['gbtree', 'gblinear'] learning_rate=[0.05,0.1,0.15,0.20] min_child_weight=[1,2,3,4] # Define the grid of hyperparameters to see hyperparameter_grid = { 'n_estimators': n_estimators,</pre>	rch
<pre>'max_depth':max_depth, 'learning_rate':learning_rate, 'min_child_weight':min_child_weight, 'booster':booster, 'base_score':base_score } In [38]: regressor=xgboost.XGBRegressor(base_score=colsample_bytree=1, gamma=0, learning max_depth=2, min_child_weight=1, min_jobs=1, nthread=None, objective='score_alpha=0, reg_lambda=1, scale_possilent=True, subsample=1)</pre>	ssing= None , n_estimators=900, reg:linear', random_state=0,
<pre>In [39]: regressor.fit(X_train,y_train) C:\Users\Administrator\AppData\Local\Program inear is now deprecated in favor of reg:squa warnings.warn(smsg, UserWarning) C:\Users\Administrator\AppData\Local\Program Parameters: { "silent" } are not used. warnings.warn(smsg, UserWarning) Out[39]:</pre>	s\Python\Python311\Lib\site-packages\xgboost\core.py:160: UserWarning: [14:27:38] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0750514818a16474a-1\xgboost\xgboost\core.py:160: UserWarning: [14:27:38] WARNING: C:\buildkite-agent\builds\buildkite-windows-cpu-autoscaling-group-i-0750514818a16474a-1\xgboost\core.py:160: UserWarning: [14:27:38] WARNING: C:\buildkite-agent\buildkite-windows-cpu-autoscaling-group-i-0750514818a16474a-1\xgboost\core.py:160: UserWarning: [14:27:38] WARNING: C:\buildkite-agent\buildkite-agent\buildkite-windows-cpu-autoscaling-group-i
device=None, early_stopping eval_metric=None, feature_f importance_type=None, inter learning_rate=0.1, max_bin= max_cat_to_onehot=None, max max_leaves=None, min_child_ monotone_constraints=None,	ple_bynode=None, colsample_bytree=1, prounds=None, enable_categorical=False, prounds=None, gamma=0, grow_policy=None, prounds=None, gamma=0, grow_policy=None, prounds=None, gamma=0, grow_policy=None, prounds=None, prounds=None
<pre>In [55]: import pickle filename = 'finalized_model.pkl' pickle.dump(classifier, open(filename, 'wb TypeError</pre>	aceback (most recent call last) e, 'wb')) ype' object
C:\Users\Administrator\AppData\Local\Temp\ip A value is trying to be set on a copy of a s	ykernel_2036\708324064.py:1: SettingWithCopyWarning: lice from a DataFrame //pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy =True) pur XGBoost regressor
<pre>regressor.fit(X_train, y_train) # Now, make predictions on the test data y_pred = regressor.predict(df_Test) y_pred Out[42]: array([127911.22, 155760. , 177858.73, 235567.4], dtype=float32) In [61]: y_test=final_df['SalePrice'] In [62]: from sklearn.metrics import r2_score</pre>	., 169733.56, 112765.57,
<pre># Assuming you've predicted y_pred using you've predicted y_pred using you've predicted y_pred using you've predicted y_pred) print("R-squared score:", r2) ValueError</pre>	aceback (most recent call last)
File ~\AppData\Local\Programs\Python\Python3 208 try: 209 with config_context(210 skip_parameter_validation=(211 prefer_skip_nested_validation=(212 prefer_skip_nested_validation=(213 prefer_skip_nested_validation=(214 prefer_skip_nested_validation=(215 except_skip_nested_validation=(216 prefer_skip_nested_validation=(217 prefer_skip_nested_validation=(218 # when the function is just a with substitution is just a with substitut	ation or global_skip_validation apper around an estimator, we allow ation to the estimator, but we replace the name of the function in the error
<pre>219 # message to avoid confusion. 220 msg = re.sub(221</pre>	
<pre>869): 870 """:math:`R^2` (coefficient of coefficient of</pre>	<pre>put = _check_reg_targets(y_pred, sample_weight)</pre>
65 def _check_reg_targets(y_true, y_pred 66 """Check that y_true and y_pred 67 68 Parameters () 97 correct keyword. 98 """> 99 check_consistent_length(y_true, y_true = check_array(y_true, ens	belong to the same regression task. y_pred)

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\utils\validation.py:407, in check_consistent_length(*arrays)
405 uniques = np.unique(lengths)

"Found input variables with inconsistent numbers of samples: %r"

ValueError: Found input variables with inconsistent numbers of samples: [2881, 1459]

406 if len(uniques) > 1: --> 407 raise ValueError(

% [int(1) for 1 in lengths]

408

409

410)

In [1]: **import** numpy **as** np

Out[2]:

import pandas as pd

In [2]: df=pd.read_csv('train.csv')
 df

import seaborn as sns

import matplotlib.pyplot as plt

Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour Utilities ... PoolArea PoolQC Fence MiscFeature MiscVal MoSold YrSold SaleType SaleCondition SalePrice

