PROJECT 2 - AMES HOUSING DATA AND KAGGLE CHALLENGE

PROBLEM STATEMENT:

CREATE A REGRESSION MODEL BASED ON THE AMES HOUSING DATASET TO PREDICT THE PRICE OF A HOUSE AT SALE.

SAMAY SHAH – GENERAL ASSEMBLY 2021

OVERVIEW

- EDA (EXPLORATORY DATA ANALYSIS)
- BUILD REGRESSION MODELS
- COMPARE ERROR IN OUR MODEL USING RMSE (AS REQUIRED)
- CROSS VALIDATE
- TEST OUR MODEL ON THE TEST DATASET PROVIDED
- INSIGHTS
- WHAT ELSE COULD I HAVE DONE?
- EXPLORE PROJECT FILES
- QUESTIONS

EDA (EXPLORATORY DATA ANALYSIS)

- FOLLOW THE 5 STEP DATA ANALYSIS PROCESS
  1. Identify variable and data types
  2. Analyze basic metrics and perform imputation
  3. Univariate Analysis & fixing anomalies
  4. Bivariate/Relationship Analysis
  5. Export clean dataset

  AS SIMPLE AS THAT!

  More information on 5 Step EDA

STEP 1. IDENTIFY VARIABLE AND DATA TYPES

```
## Explore all features in the dataframe

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2051 entries, 0 to 2050
Data columns (total 81 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Id               2051 non-null   int64
 1   PID              2051 non-null   int64
 2   MS SubClass      2051 non-null   int64
 3   MS Zoning        2051 non-null   object
 4   Lot Frontage     1721 non-null   float64
 5   Lot Area         2051 non-null   int64
 6   Street           2051 non-null   object
 7   Alley            140 non-null    object
 8   Lot Shape        2051 non-null   object
 9   Land Contour     2051 non-null   object
 10  Utilities        2051 non-null   object
 11  Lot Config       2051 non-null   object
 12  Land Slope       2051 non-null   object
 13  Neighborhood     2051 non-null   object
 14  Condition 1      2051 non-null   object
 15  Condition 2      2051 non-null   object
 16  Bldg Type        2051 non-null   object
```

```
df.isnull().sum().sort_values(ascending = False).head(20)
```

```
Pool QC           2042
Misc Feature      1986
Alley             1911
Fence             1651
Fireplace Qu      1000
Lot Frontage       330
Garage Finish      114
Garage Cond        114
Garage Qual        114
Garage Yr Blt      114
Garage Type        113
Bsmt Exposure       58
BsmtFin Type 2      56
BsmtFin Type 1      55
Bsmt Cond           55
Bsmt Qual           55
Mas Vnr Type        22
Mas Vnr Area        22
Bsmt Half Bath       2
Bsmt Full Bath       2
```

STEP 2. ANALYZE BASIC METRICS AND PERFORM IMPUTATION

```python
## Explore all features in the dataframe

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2051 entries, 0 to 2050
Data columns (total 81 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Id              2051 non-null   int64
 1   PID             2051 non-null   int64
 2   MS SubClass     2051 non-null   int64
 3   MS Zoning       2051 non-null   object
 4   Lot Frontage    1721 non-null   float64
 5   Lot Area        2051 non-null   int64
 6   Street          2051 non-null   object
 7   Alley           140 non-null    object
 8   Lot Shape       2051 non-null   object
 9   Land Contour    2051 non-null   object
 10  Utilities       2051 non-null   object
 11  Lot Config      2051 non-null   object
 12  Land Slope      2051 non-null   object
 13  Neighborhood    2051 non-null   object
 14  Condition 1     2051 non-null   object
 15  Condition 2     2051 non-null   object
 16  Bldg Type       2051 non-null   object
```

```python
df.isnull().sum().sort_values(ascending = False).head(20)
```

```
Pool QC           2042
Misc Feature      1986
Alley             1911
Fence             1651
Fireplace Qu      1000
Lot Frontage       330
Garage Finish      114
Garage Cond        114
Garage Qual        114
Garage Yr Blt      114
Garage Type        113
Bsmt Exposure       58
BsmtFin Type 2      56
BsmtFin Type 1      55
Bsmt Cond           55
Bsmt Qual           55
Mas Vnr Type        22
Mas Vnr Area        22
Bsmt Half Bath       2
Bsmt Full Bath       2
```

EXAMPLE

```
##1 Lot Frontage Feature
##Replace missing values for 'Lot Frontage' with .median() values

for i in ['Lot Frontage']:
    df.loc[df.loc[:,i].isnull(),i]=df.loc[:,i].median()
```

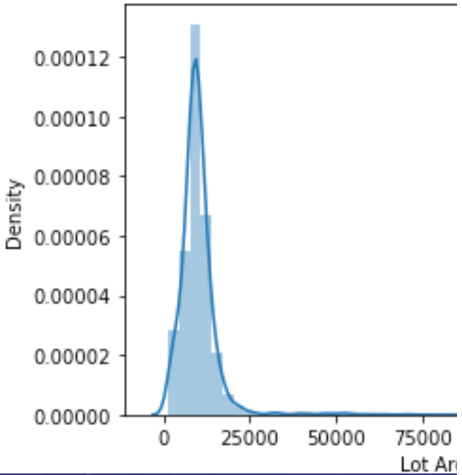
```
##Change the datatype to integer for easy calculations

df['Lot Frontage'] = df['Lot Frontage'].astype(int)
```

```
##22 BsmtFin SF 1, BsmtFin SF 2, mapped and typecast to int for consistency
## Grouped/Renamed - to dummify later

df['BsmtFin Type 1'] = df['BsmtFin Type 1'].map({'GLQ':5,'ALQ':4,'BLQ':3,'Rec':4,'LwQ':2,'Unf':1,'NA':0})
df['BsmtFin Type 2'] = df['BsmtFin Type 2'].map({'GLQ':5,'ALQ':4,'BLQ':3,'Rec':4,'LwQ':2,'Unf':1,'NA':0})
df['BsmtFin Type 1'] = df['BsmtFin Type 1'].astype(int)
df['BsmtFin Type 2'] = df['BsmtFin Type 2'].astype(int)
```

STEP 3: UNIVARIATE ANALYSIS & FIXING ANOMALIES

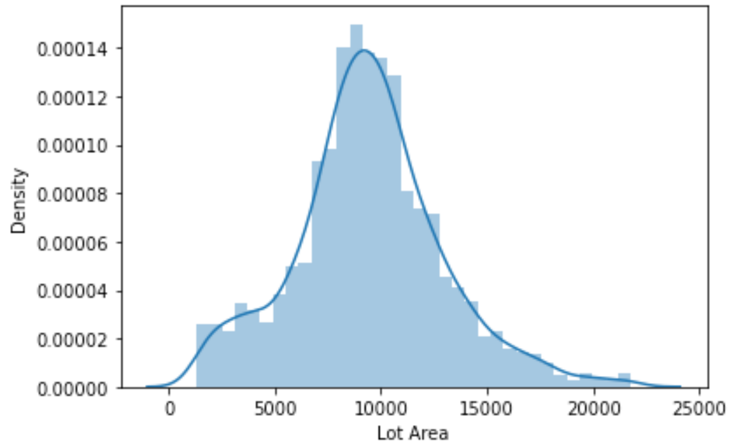
```
#1 Lot Area - Before
sns.distplot(df3['Lot Area']);
```

```
/Users/samay20/opt/anaconda3/lib/python3.8/site-pack
a deprecated function and will be removed in a futur
re-level function with similar
  warnings.warn(msg, FutureWar
```




```
#1 Lot Area - After
### Looks almost normally distributed now

sns.distplot(df4['Lot Area']);
```

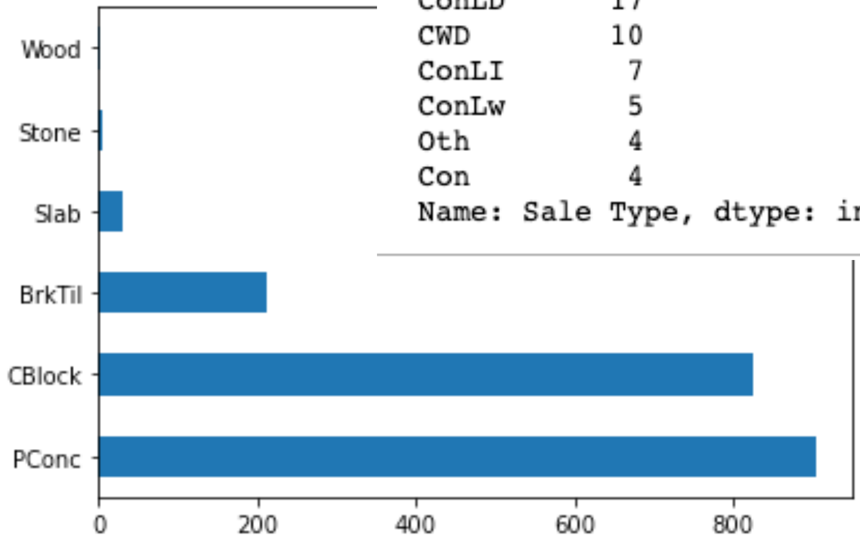
```
/Users/samay20/opt/anaconda3/lib/python3.8/site-pack
a deprecated function and will be removed in a futur
re-level function with similar flexibility) or `hist
  warnings.warn(msg, FutureWarning)
```




```
## Scale down the SalePrice using log transformatoin

df4['log_SalePrice'] = np.log(df4['SalePrice'])
```

```
#Categorial
df6['Foundation'].val
```

<AxesSubplot:>




```
## Group into fewer categories

df['Sale Type'].value_counts()
```

```
WD       1781
New       160
COD        63
ConLD      17
CWD        10
ConLI       7
ConLw       5
Oth         4
Con         4
Name: Sale Type, dtype: int64
```

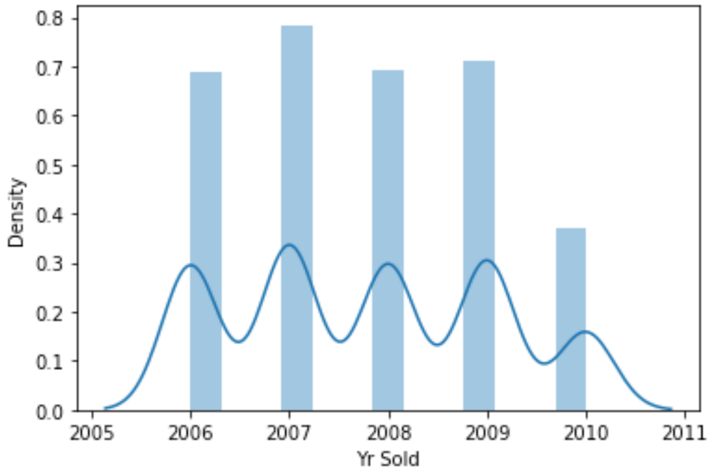
THERE IS ALWAYS ROOM FOR SOME ADDITIONAL INFORMATION

Although the housing market looked like it might have been on the road to stabilization in early 2010, the second half of the year dashed those hopes. Ultimately, housing prices ended the year down 4.1%, according to housing industry consulting firm Clear Capital. The company released its 2010 report today, which includes 2011 forecasts. In short, 2011 won't be much better than 2010.

Read Full Article

[106]:
```
#8 Yr Sold - Looks like the market was steady year over year, except 2010 (Categorical)
sns.distplot(df6['Yr Sold']);
```

```
/Users/samay20/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

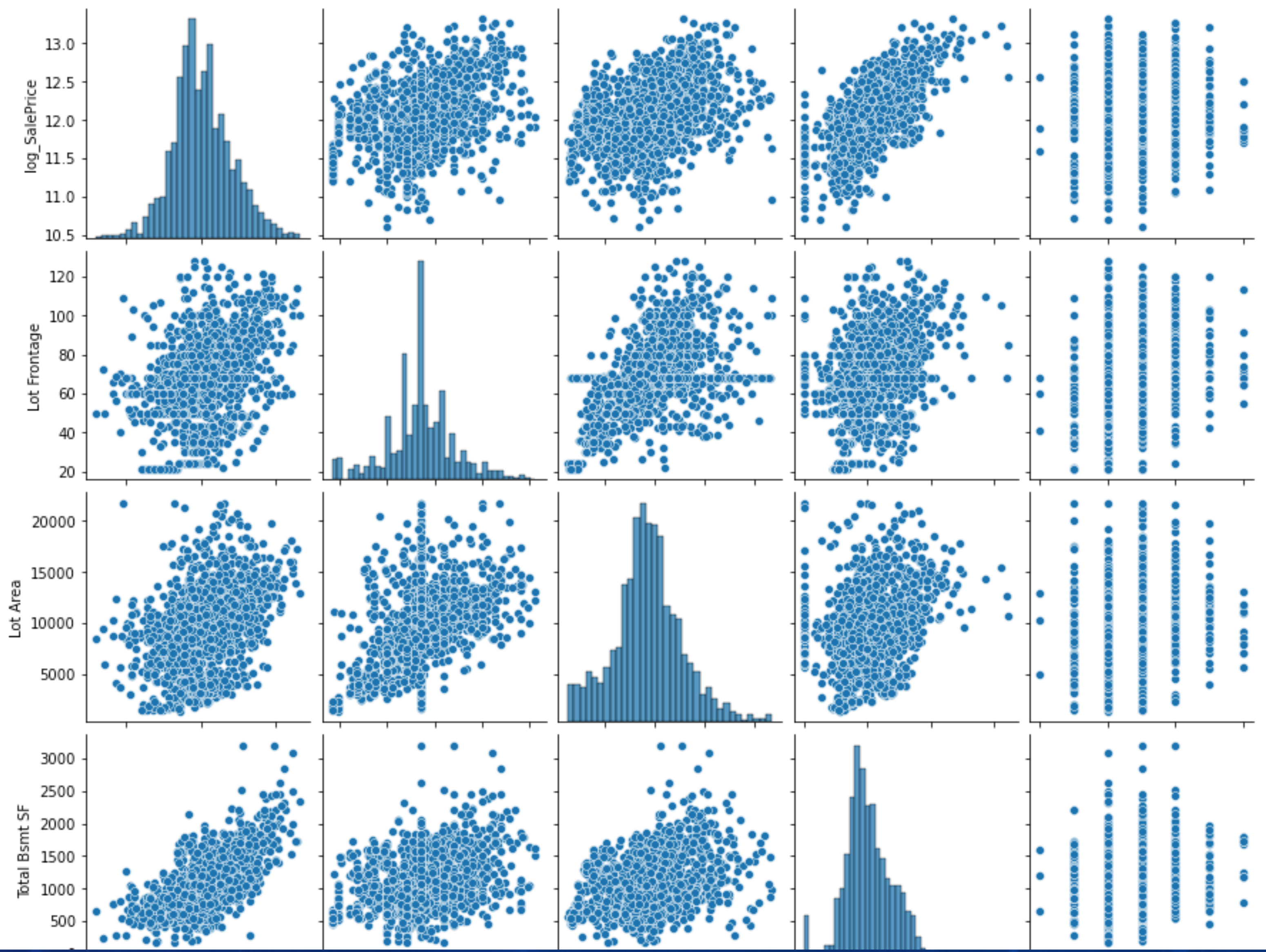


STEP 4. MULTIVARIATE/ RELATIONSHIP ANALYSIS

Seaborn library allows us to Pair Plots. Plot relationships across all numeric columns within a DataFrame.

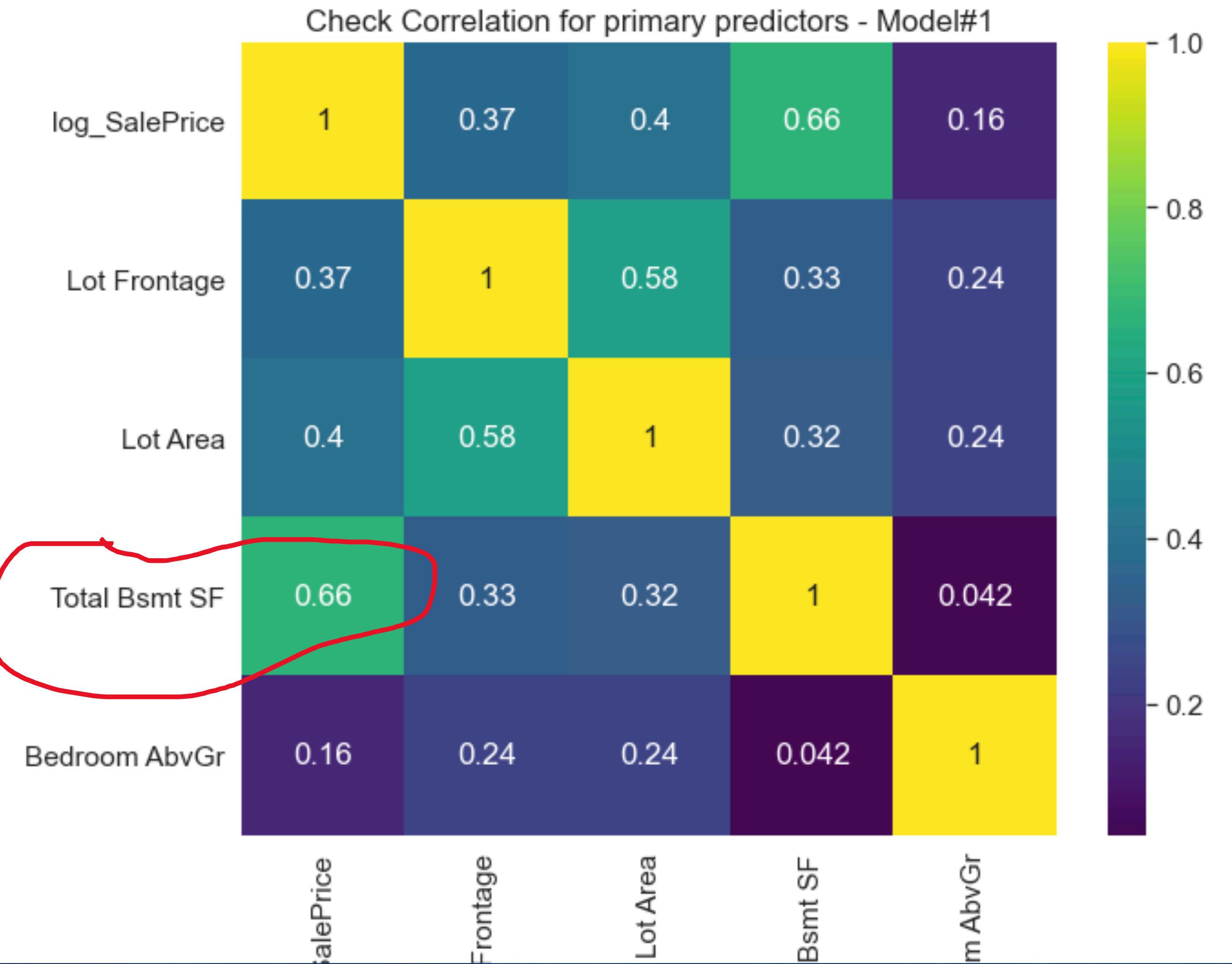
```
sns.pairplot(pairplot1)
```

```
<seaborn.axisgrid.PairGrid at 0x7fcf5a997ee0>
```



CHECK FOR CORRELATION

```python
plt.figure(figsize=(10,8))
sns.set(font_scale=1.4)
sns.heatmap(variables1.corr(), cmap='viridis', annot=True)
plt.title('Check Correlation for primary predictors - Model#1');
```



CREATE DUMMY VARIABLES FOR CATEGORICAL COLUMNS

Syntax: pd.getdummies(DataFrame, drop_first=True)

```
In [132]: createDummies = df7[['Neighborhood','Foundation','Sale Type','proximity_to','Density','Bldg_type','House_style','house_

In [133]: our_dummies1 = pd.get_dummies(createDummies,drop_first=True)
```

STEP 5: EXPORT CLEAN DATASET

Part1: Combine select dummified columns with our other main DataFrame using pd.concat()

Model#4 (Final) - with selected features using trial-and-error approach

```
143]: reg_sample_4_combined = pd.concat([df7,our_dummies1],axis=1)
```

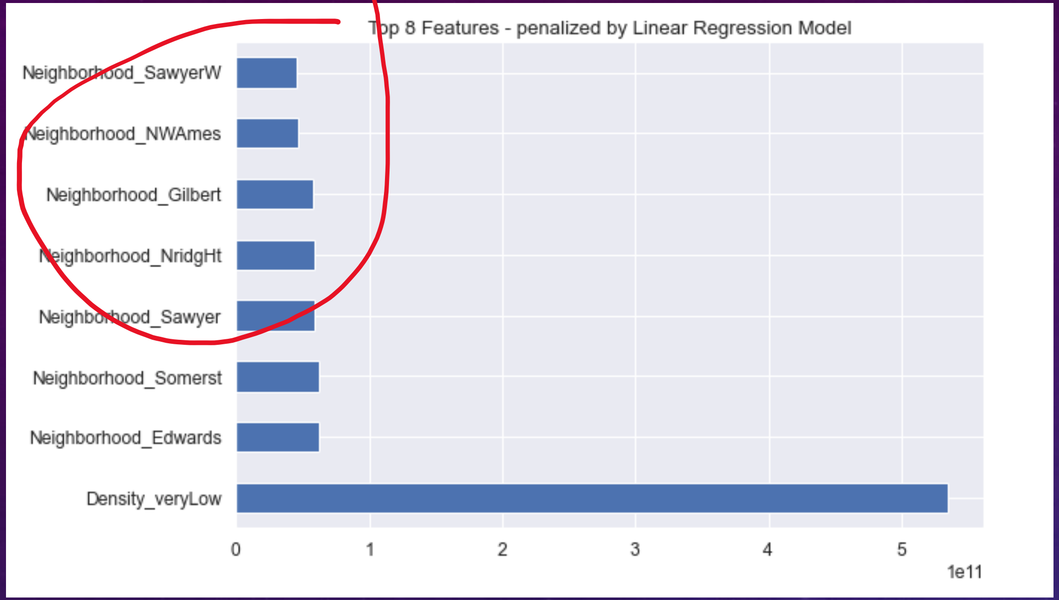
Part2: Export file using DataFrame.to_csv(filename, index=False)

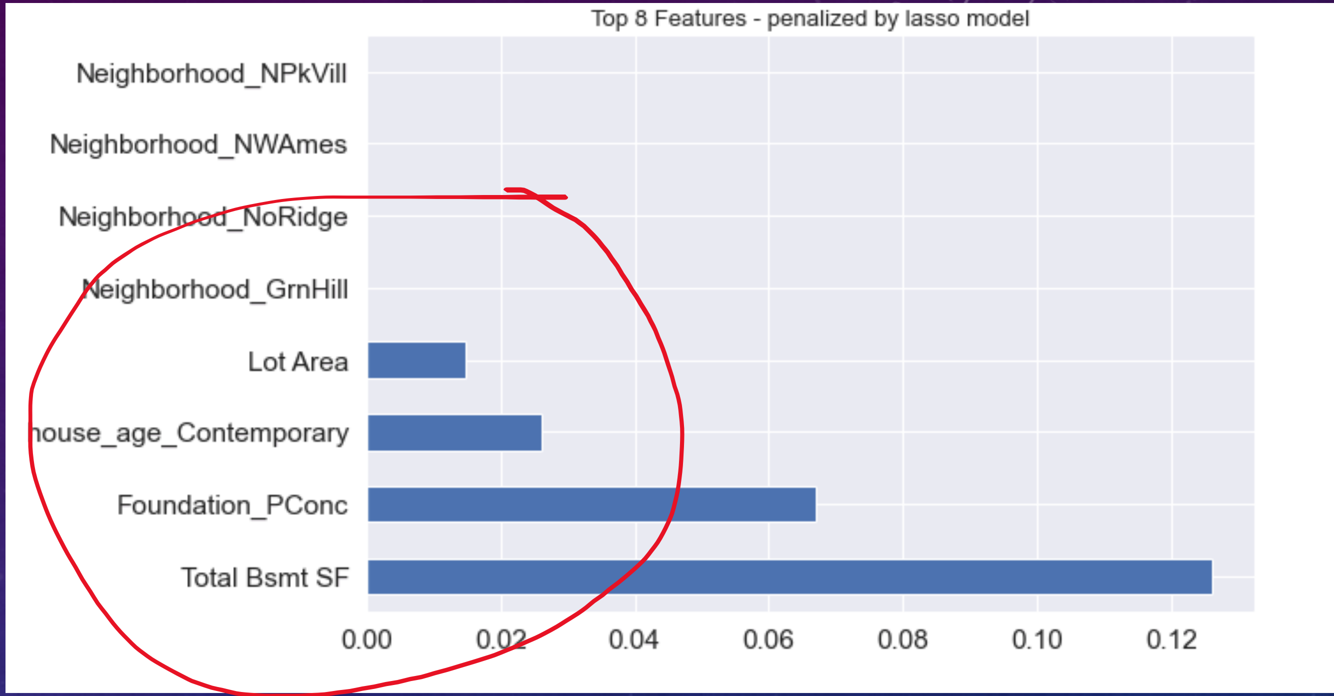
```
In [146]: reg_sample_4_withdummies.to_csv('datasets/model4_withdummies.csv',index=False)
```

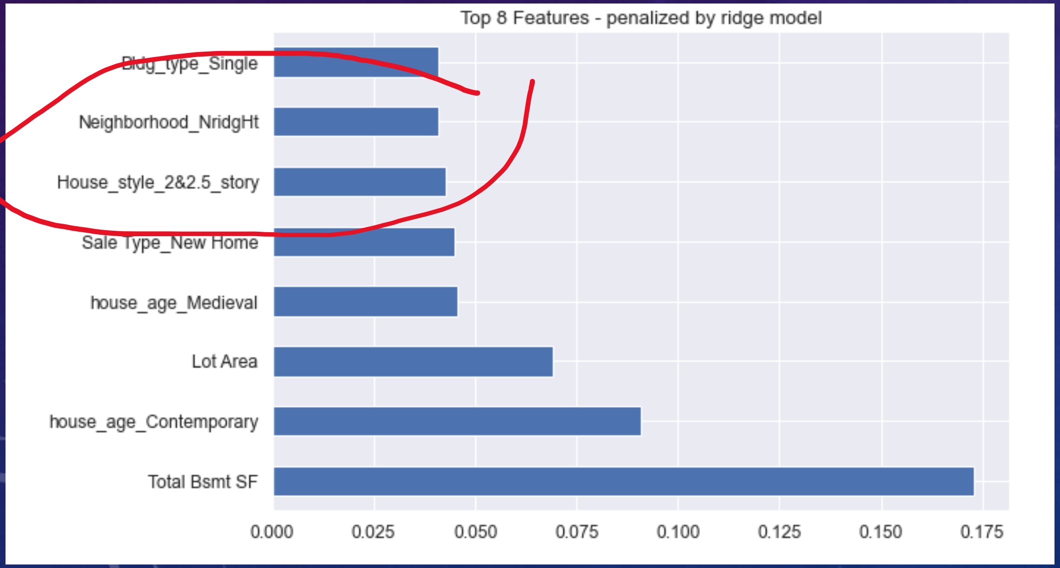
MULTIPLE LINEAR REGRESSION!

- Built 4 Models,
  - **Model#1**: Features that showed high correlation with our Sale Price feature along with few manually mapped categorical columns. Use stepwise regression using Ridge and Lasso Models to help narrow down the features.
    - Train R2: 0.83, Adj. R2: 0.82
    - Test R2: 0.77, Adj. 0.74

    Model's overfit. What can help? Yes. Check cross validation score using Ridge and Lasso Regression models. (trial-and-error for different values of alpha)







Penalized few features too much using alpha of 0.1 for Lasso model

```
]: print(lasso_model2.score(x_train_scaled,y_train))
   print(lasso_model2.score(x_test_scaled,y_test))

   ## significant difference
   ## Our results got better from train: 0.83 and Test:0.77 (overfit)

0.8129567728302759 TRAIN
0.7619687313752818 TEST
```

MULTIPLE LINEAR REGRESSION!

- Built 4 Models,
  - **Model#2**: Some features from Model1 along with the dummies that we created using pd.get_dummies(). Using Forward selection approach.
    - Train R2: 0.87, Adj. R2: 0.86
    - Test R2: 0.84, Adj. 0.80

    Model's overfit.

MULTIPLE LINEAR REGRESSION!

- Built 4 Models,
  - **Model#3**: Bring best features from Model#1 and Model#2 and select featues using trial-and-error method.
    - Train R2: 0.895, Adj. R2: 0.892
    - Test R2: 0.872, Adj. 0.864

  Model's still overfit. Lets build one more using a different approach.

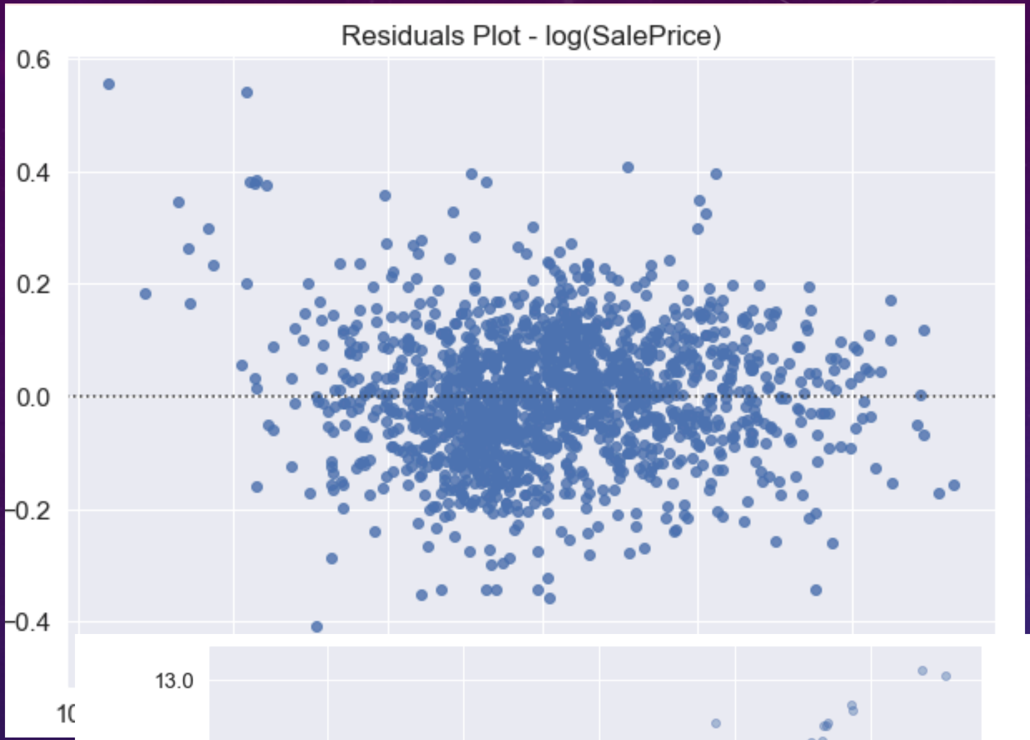
MULTIPLE LINEAR REGRESSION!

- Built 4 Models,
  - **Model#4**: Bring all possible features in and use backward selection approach.

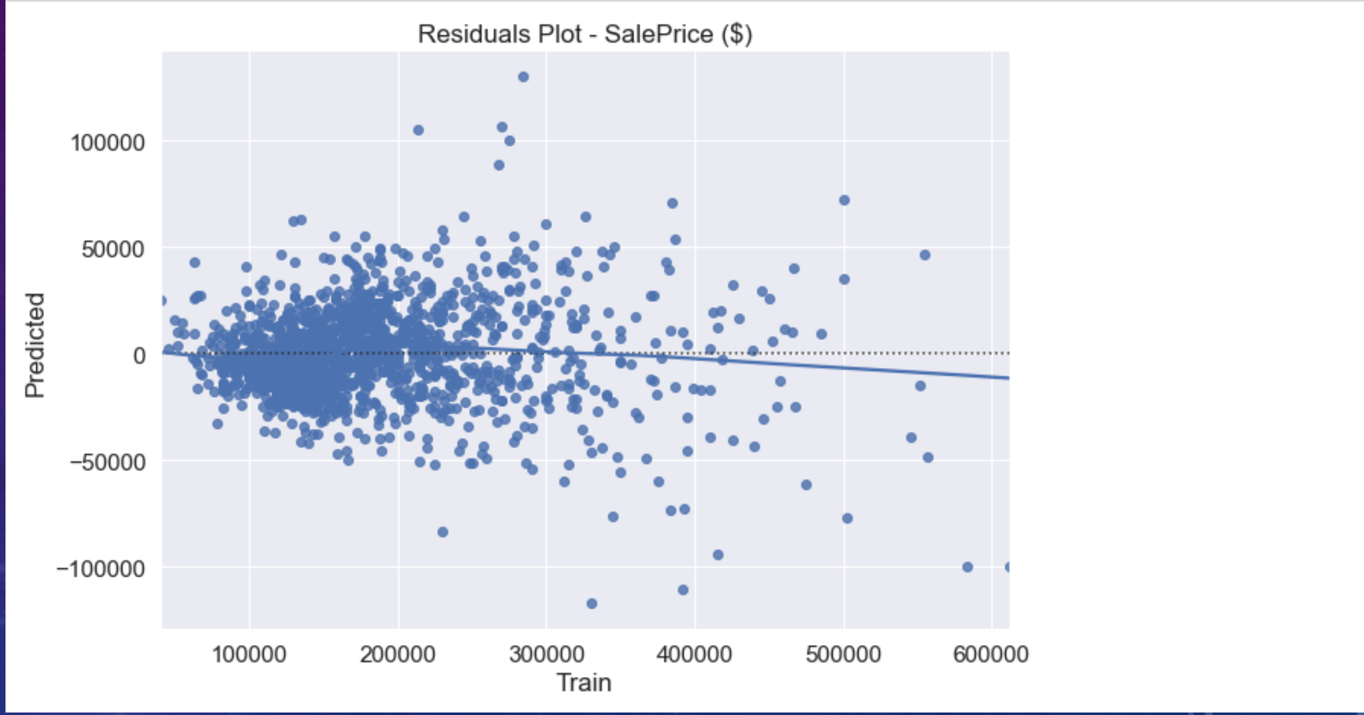
```
## Train Scores

lord_of_the_metrics(y_train, y_hat, inputs4.shape[1])

Mean squared error      =  0.014427692484988411
Root mean squared error =  0.12011532993331206
Median absolute error   =  0.0712557202935038
R^2                     =  0.906428391859603
Adjusted R^2            =  0.9043652046502817
```

```
## Test

lord_of_the_metrics(y_test, y_hat_test, inputs4.shape[1])

Mean squared error      =  0.014505610645402655
Root mean squared error =  0.1204392404717111
Median absolute error   =  0.06963519898172166
R^2                     =  0.907311874294196
Adjusted R^2            =  0.8985579957553145
```

- Almost close.




```
# -ve values for y_train-y_hat as SalePrices increase
# => Model gives higher predictions as SalesPrices increase as compared to houses with lower SalePrice
```



INSIGHTS FROM OUR TEST DATA SET

	Predictions	Target	Residual	Difference%
182	180041.38	180000.00	-41.38	0.02
145	167973.94	167900.00	-73.94	0.04
328	190650.67	190550.00	-100.67	0.05
183	148583.32	148500.00	-83.32	0.06
187	159118.58	159000.00	-118.58	0.07
...	...	...	...	...
250	367087.47	250000.00	-117087.47	46.83
336	121466.28	80000.00	-41466.28	51.83
357	79436.68	50138.00	-29298.68	58.44
219	101568.74	60000.00	-41568.74	69.28
393	288405.87	147000.00	-141405.87	96.19

395 rows × 4 columns

RESULT:
TEST DATASET

```
[46]: ## Final Predictions

      y_testpred = reg4.predict(tinput_scaled)

[47]: finalresult = pd.DataFrame(index=df4_testdata['Id'])

[48]: finalresult['SalePrice '] = np.exp(y_testpred)

[49]: finalresult
```

[49]:

Id	SalePrice
2658	135411.00
2718	156617.92
2414	254079.31
1989	104358.65
625	186612.68
...	...
1662	191745.64
1234	215992.11
1373	114764.44
1672	111031.33
1939	129363.88

843 rows × 1 columns

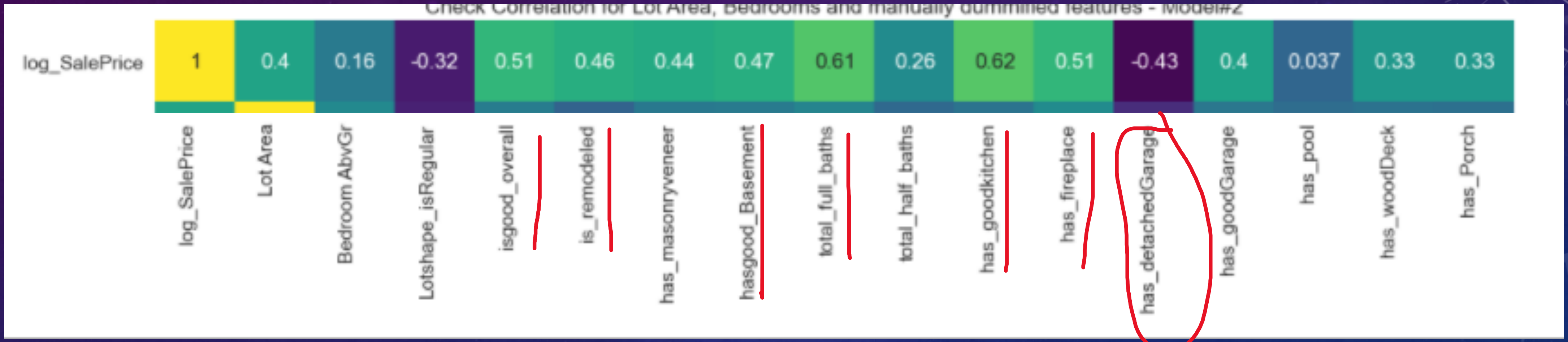
```
[50]: finalresult.to_csv('Model4_Predicted_SalePrices_Test_Dataset.csv')
```

COEFFICIENTS SUMMARY TABLE TO CREATE INFERENCES

```python
reg4_summary = pd.DataFrame(inputs4.columns.values, columns=['Features'])
reg4_summary['Coefficients'] = reg4.coef_

reg4_summary

## Helpful in creating inferences like
## Lot Area, Overall Condition, is_fullyfurnished, # of bathrooms, ishouse_contemporary (made after 1999) are
## more significant in predicting house prices than others.
```

	Features	Coefficients
12	Lotshape_isRegular	-0.01
13	isSlopeNormal	-0.01
14	is_remodeled	0.01
15	is_roofGable	-0.00
27	proximity_to_veryClose	-0.02
28	House_style_1_story	-0.04
29	House_style_Splitlevel	-0.02
30	house_age_Contemporary	0.04
31	house_age_Medieval	0.02
32	house_age_Stone-age	-0.01
40	Density_veryLow	5.351363e+11

COEFFICIENTS SUMMARY TABLE TO CREATE INFERENCES



CORRELATION VALUES

QUESTIONS? OR LETS LOOK AT THE PROJECT ☺