

```
In [1]: #from google.colab import drive
#drive.mount('/content/drive')
```

```
In [500... import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import numpy as np
from sklearn.preprocessing import OneHotEncoder , StandardScaler, OrdinalEncoder
from sklearn.decomposition import PCA
from sklearn.feature_selection import RFE
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split, cross_validate, GridSearch
from sklearn.metrics import r2_score, mean_squared_error, make_scorer
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
import keras
from keras import models, layers
from keras.layers import Dense, Dropout, Activation
from keras.regularizers import l2
from keras.callbacks import ModelCheckpoint
from keras.wrappers.scikit_learn import KerasRegressor
%matplotlib inline
warnings.filterwarnings("ignore")
```

```
In [501... # Functions

def get_results(results):

    '''Prints the mean negative rmse from the model that had the lowest mean rms

    print('The best model parameters produce a mean rmse score on train data of:
    print(results.cv_results_['mean_train_neg_root_mean_squared_error'][results.
    print('')
    print('The best model parameters produce a mean R-squared score on train dat
    print(results.cv_results_['mean_train_r2'][results.best_index_])
    print('')
    print('The best model parameters produce a mean rmse score on test data of:'
    print(results.cv_results_['mean_test_neg_root_mean_squared_error'][results.b
    print('')
    print('The best model parameters produce a mean R-squared score on test data
    print(results.cv_results_['mean_test_r2'][results.best_index_])

def get_results_preds(y_train, y_train_preds, y_test, y_test_preds):

    # Getting R squared scores for training and testing data
    y_tr_r2 = r2_score(y_tr, y_tr_preds)
    y_val_r2 = r2_score(y_val, y_val_preds)

    print(f'R-Squared score for the training data: {y_tr_r2}')
    print(f'R-Squared score for the testing data: {y_val_r2}')
```

```

print('')
print('')

# Getting Mean Squared Error for training and testing data
y_tr_rmse = mean_squared_error(y_tr, y_tr_preds, squared = False)
y_val_rmse = mean_squared_error(y_val, y_val_preds, squared = False)

print(f'Root Mean Squared Error for the training data: {y_tr_rmse}')
print('')
print(f'Root Mean Squared Error for the testing data: {y_val_rmse}')


def replace_NaN(df, features, value):
    for feature in features:
        df[feature].replace(np.NaN, value, inplace=True)


categories = ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex']

scale_dict = dict(zip(categories, range(0,6)))

def scale_function(category):
    return scale_dict[category]


def map_function(df, features, function):
    '''maps certain categorical variables to numerical values'''
    for feature in features:
        df[feature] = df[feature].map(lambda x : function(str(x)))


def get_cv_results(cv):
    train_scores = cv['train_score']
    mean_train_score = cv['train_score'].mean()
    test_scores = cv['test_score']
    mean_test_score = cv['test_score'].mean()
    print(cv)
    print('')
    print(f'CV Train Scores: {train_scores}')
    print('')
    print(f'Mean CV Train Score: {mean_train_score}')
    print('')
    print(f'CV Test Scores: {test_scores}')
    print('')
    print(f'Mean CV Test Score: {mean_test_score}')


def train_error_distribution(y, y_hat):
    train_error = y - y_hat
    plt.scatter(y_hat, train_error)
    plt.axhline(y=0, color='r', linestyle='-')
    plt.title('Distribution Of Training Error')
    plt.ylabel('Error')
    plt.xlabel('Prediction');


def test_error_distribution(y, y_hat):
    test_error = y_val - y_hat
    plt.axhline(y=0.5, color='r', linestyle='-')

```

```

plt.title('Distribution Of Testing Error')
plt.ylabel('Error')
plt.xlabel('Prediction');

def visualize_nn(history, model, x_train, y_train, x_validation, y_validation):

    rmse = np.sqrt(history.history['mse'])
    val_rmse = np.sqrt(history.history['val_mse'])

    epochs = range(len(rmse))
    plt.plot(epochs, rmse, 'b-', label='Training RMSE');
    plt.plot(epochs, val_rmse, 'r-.', label='Validation RMSE');
    plt.title('Training And Validation MSE')
    plt.legend()
    plt.figure(figsize=(16,8));

    print('')
    print('Training Evaluation:')
    tr_eval = model.evaluate(x_train, y_train, batch_size=100)
    tr_eval
    print('')
    print('Validation Evaluation:')
    t_eval = model.evaluate(x_validation, y_validation, batch_size=50)
    t_eval
    print('')
    eval_rmse = np.sqrt(tr_eval[1])
    print(f'Train Evaluation RMSE: {eval_rmse}')
    print('')
    eval_val_rmse = np.sqrt(t_eval[1])
    print(f'Validation Evaluation RMSE: {eval_val_rmse}');

```

EDA

```

In [502... # Loading the data
train_df = pd.read_csv('data/train.csv')

```

```

In [503... train_df.head()

```

```

Out[503...

```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utili
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	All
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	All
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	All
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	All
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	All

5 rows × 81 columns

```

In [504... train_df.shape

```

```
In [505... train_df.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
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
56  Fireplaces          770 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

```

Predictors that need to be dropped, because of too many null values – ['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']

```
In [506... train_df.describe()
```

```

Out[506...
      Id  MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond  Ye
count  1460.000000  1460.000000  1201.000000  1460.000000  1460.000000  1460.000000  1460.000000
mean    730.500000    56.897260    70.049958  10516.828082    6.099315    5.575342  1971.250000
std    421.610009    42.300571    24.284752   9981.264932    1.382997    1.112799    30.250000
min     1.000000    20.000000    21.000000   1300.000000    1.000000    1.000000  1872.000000
25%    365.750000    20.000000    59.000000   7553.500000    5.000000    5.000000  1954.000000
50%    730.500000    50.000000    69.000000   9478.500000    6.000000    5.000000  1973.000000
75%   1095.250000    70.000000    80.000000  11601.500000    7.000000    6.000000  2000.000000
max   1460.000000   190.000000   313.000000  215245.000000   10.000000    9.000000  2010.000000

```

8 rows × 38 columns

```
In [507... target = train_df['SalePrice']
```

```
In [508... target.describe()
```

```

Out[508... count    1460.000000
mean    180921.195890
std     79442.502883
min     34900.000000
25%    129975.000000
50%    163000.000000

```

```
max      755000.000000  
Name: SalePrice, dtype: float64
```

By using a log transformation on the target variable, the distribution is now closer to normal. Hopefully this satisfies the assumption of normality. We will only truly know when we see the distribution of the errors of a baseline linear regression model.

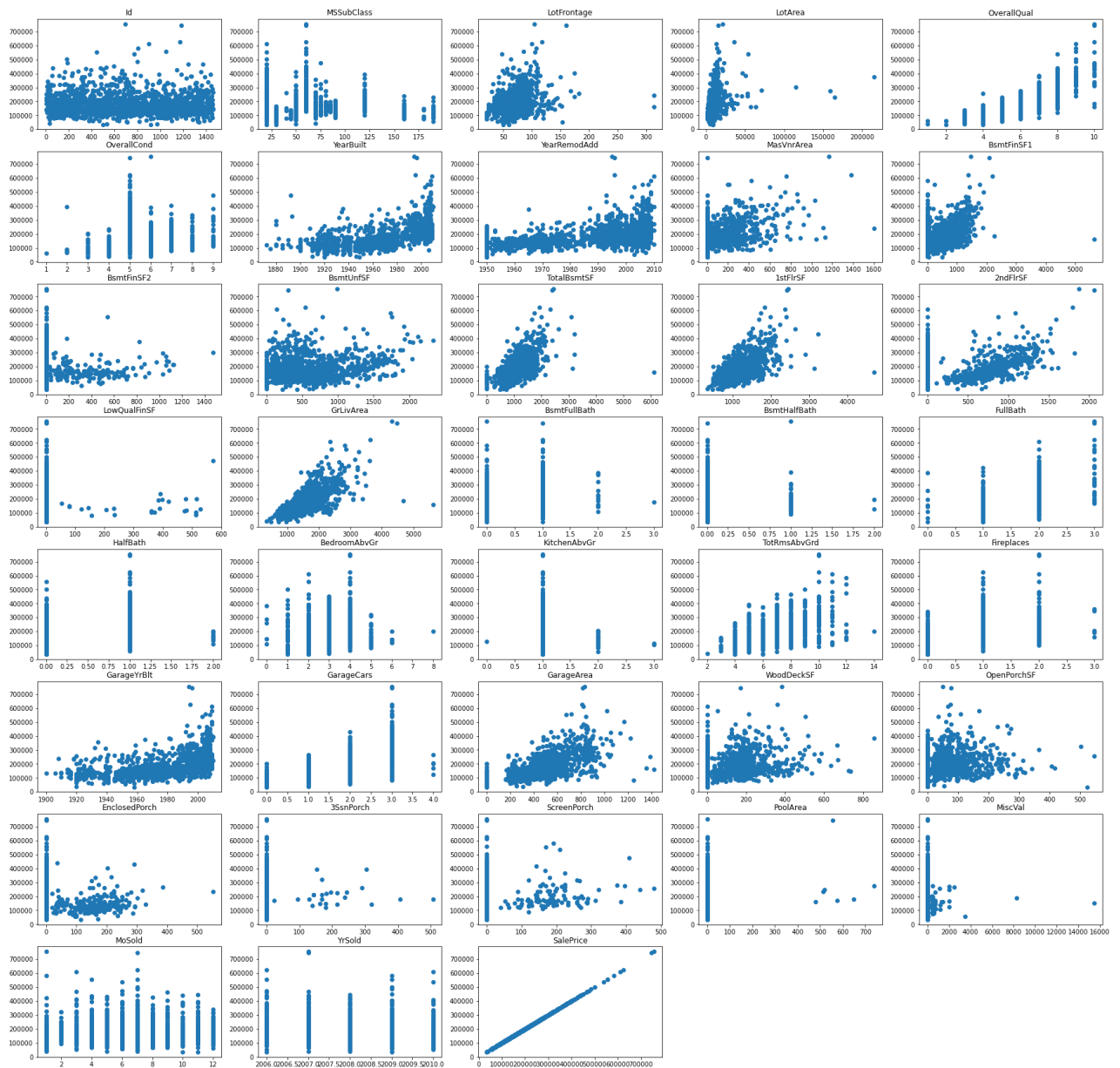
Checking Viability of Linear Model

Inspecting Linear Relationships Between Predictors And Target

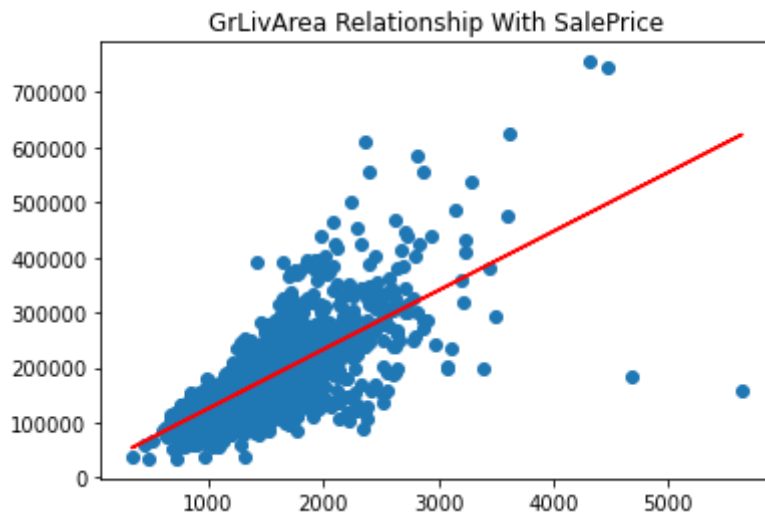
```
In [509... num_cols = list(train_df.select_dtypes(exclude=['object']))  
len(num_cols)
```

Out[509... 38

```
In [510... fig = plt.figure(1, figsize=(30,30))  
columns = 5  
rows = 8  
  
for i, col in enumerate(num_cols):  
    fig.add_subplot(rows, columns, i+1)  
    plt.scatter(train_df[col], train_df['SalePrice'])  
    plt.title(col);
```



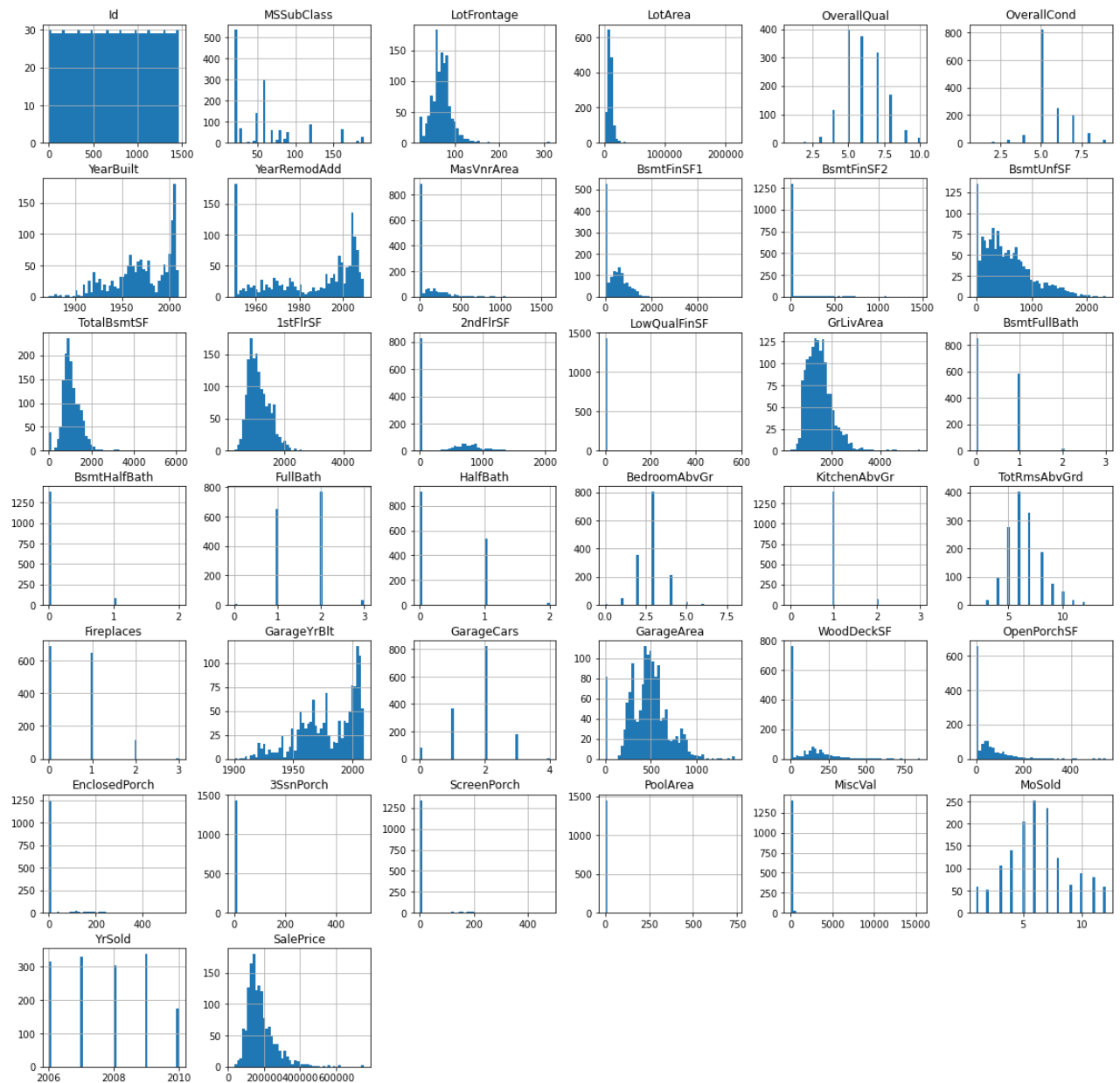
```
In [511... plt.scatter(train_df['GrLivArea'], train_df['SalePrice'])
theta = np.polyfit(train_df['GrLivArea'], train_df['SalePrice'], 1)
line = theta[1] + theta[0]*train_df['GrLivArea']
plt.plot(train_df['GrLivArea'],line, 'r')
plt.title('GrLivArea Relationship With SalePrice');
```



The independent variables that clearly do not have a linear relationship with the target are 'MSSubClass' and 'KitchensAbvGr'. I will consider dropping these variables during the feature selection stage.

Checking Multivariate Normality

```
In [512... train_df.hist(figsize=(20,20), bins=50);
```

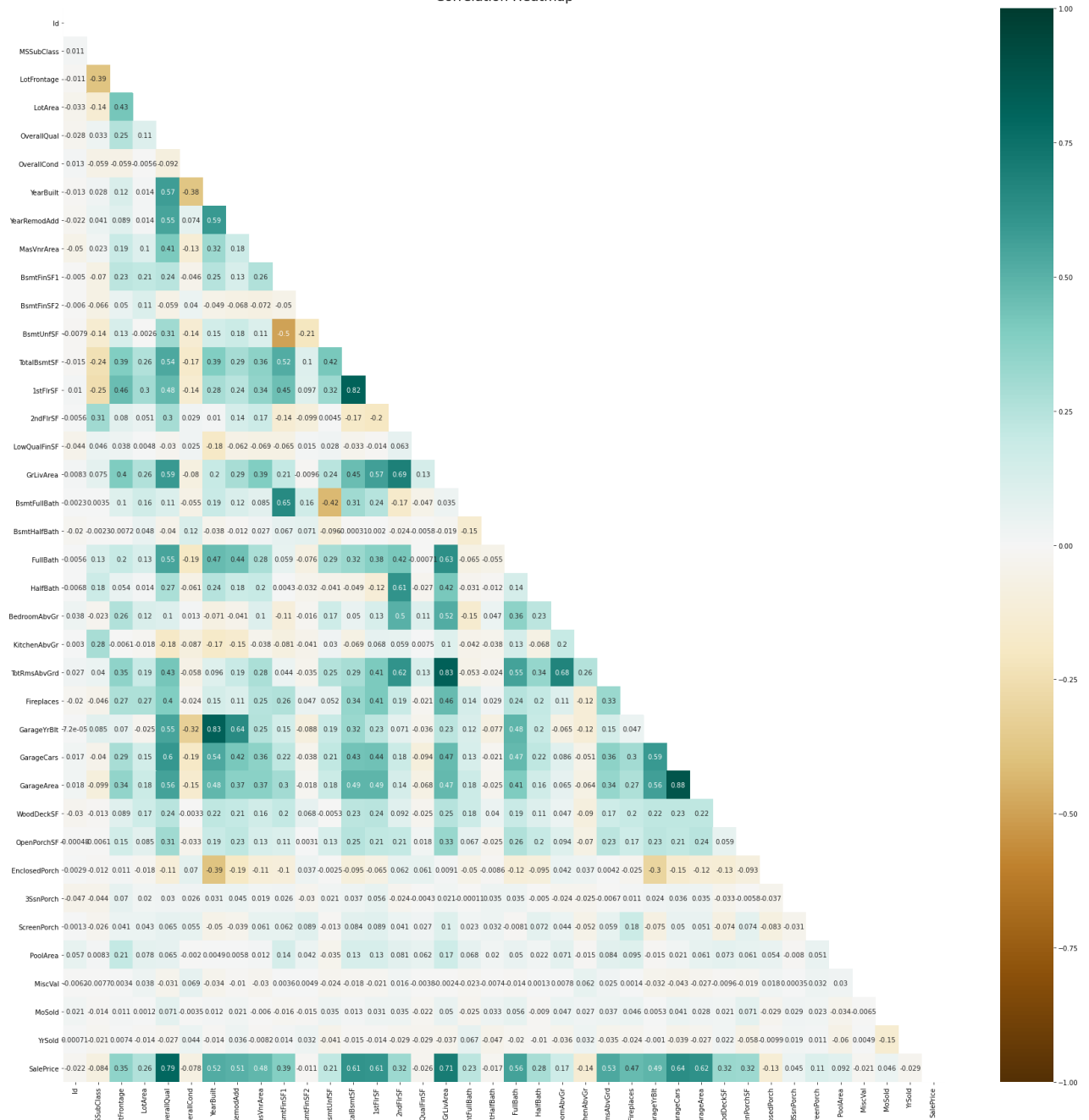



It appears that many variables in this dataset are slightly non-normally distributed. However, this is most likely an effect of the outliers, which I would like to keep for now in an effort to model the real world. And given that predictors with continuous values that are not in units of time, such as 'LotFrontage' and 'LotArea', are mostly normal, I am going to try training models on just a standard scaled raw dataset.

Checking For Multicollinearity

```
In [513... plt.figure(figsize=(30, 30))
mask = np.triu(np.ones_like(train_df.corr(), dtype=np.bool))
heatmap = sns.heatmap(train_df.corr(), mask=mask, vmin=-1, vmax=1, annot=True, c
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':20}, pad=12);
```

Correlation Heatmap

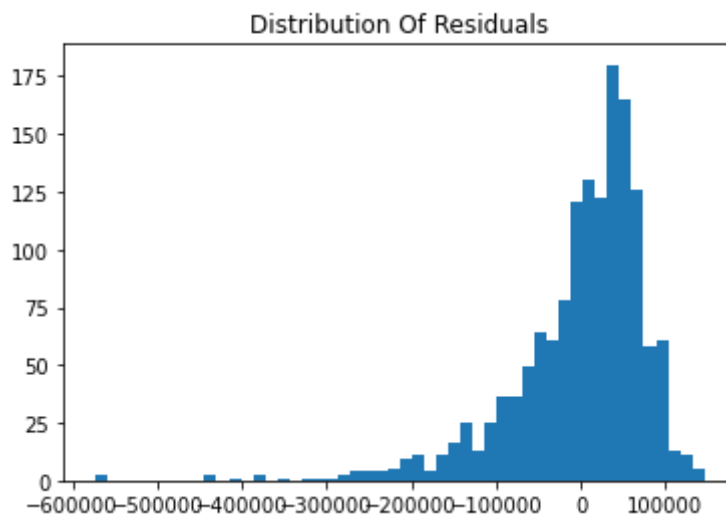


There are a few dark green points on the heatmap indicating a high correlation between two variables. When I start the process of data cleaning, I will take out multicollinear variables.

Checking Homoscedasticity of Target

In [514...]

```
# Using the mean of the target as a theoretical prediction for each observation
plt.hist((train_df['SalePrice'].mean() - train_df['SalePrice']), bins=50)
plt.title('Distribution Of Residuals');
```



It is clear that this distribution of baseline residuals skews left, however not terribly. Using linear regression is feasible, though multicollinearity needs to be reduced. I will check the residuals of the first basic linear regression for homoscedasticity.

Data Preprocessing

In [515... `train_df.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  Foundation           1460 non-null   object
29  PoolQual             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

```

```

In [516... # This dataset already contains very little data, and given the description, Lot
# So I am replacing null values of Lot Frontage with the variable mean of the ni
# to avoid dropping the 200+ rows of data or dropping the variable entirele
def impute_lot_frontage(row):
    if np.isnan(row['LotFrontage']):
        neighborhood = row['Neighborhood']
        lot_frontage_mean = train_df[train_df['Neighborhood'] == neighborhood]['
        row['LotFrontage'] = lot_frontage_mean
    return row

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
In [517... train_df = train_df.apply(impute_lot_frontage, axis=1)
```

```
In [518... # Replacing NaN values in certain garage features to 'U' for Unknown
replace_NaN(train_df, ['GarageType', 'GarageFinish', 'GarageQual', 'GarageCond'])
```

Feature Selection And Engineering

```
In [519... # Finding Multicollinear Predictors
# save absolute value of correlation matrix as a data frame
# converts all values to absolute value
# stacks the row:column pairs into a multindex
# reset the index to set the multindex to separate columns
# sort values. 0 is the column automatically generated by the stacking
train_df_mc = train_df.corr().abs().stack().reset_index().sort_values(0, ascending=False)

# rename correlation column as cc rather than 0
train_df_mc.columns = ['v1', 'v2', 'corr']

# drop duplicates. This could be dangerous if you have variables perfectly correlated
# for the sake of exercise, kept it in.
train_df_mc.drop_duplicates(inplace=True)

# reset index for future concatenation
train_df_mc.reset_index(inplace=True)
```

```
In [520... def corr_target(row):
    row['v1_y_corr'] = train_df.corr()['SalePrice'][row['v1']]
    row['v2_y_corr'] = train_df.corr()['SalePrice'][row['v2']]
    return row
```

```
In [521... train_df_mc = train_df_mc.apply(corr_target, axis=1)
```

```
In [522... train_df_mc['pairs'] = list(zip(train_df_mc.v1, train_df_mc.v2))

# drop level columns
train_df_mc.drop(columns=['index', 'v1', 'v2'], inplace=True)

# drop duplicates. This could be dangerous if you have variables perfectly correlated
# for the sake of exercise, kept it in.
train_df_mc.drop_duplicates('corr', inplace=True)

# set index to pairs
train_df_mc.set_index(['pairs'], inplace=True)
```

```
In [523... multicollinear = train_df_mc[(train_df_mc['corr'] > 0.7) & (train_df_mc['corr'] < 0.9)]
multicollinear
```

```
Out[523... corr v1_y_corr v2_y_corr
```

pairs

	corr	v1_y_corr	v2_y_corr
(GarageCars, GarageArea)	0.882475	0.640409	0.623431
(YearBuilt, GarageYrBlt)	0.825667	0.522897	0.486362
			0.533723

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	corr	v1_y_corr	v2_y_corr
pairs			
(1stFlrSF, TotalBsmtSF)	0.819530	0.605852	0.613581
(SalePrice, OverallQual)	0.790982	1.000000	0.790982
(SalePrice, GrLivArea)	0.708624	1.000000	0.708624

GaragesCars is more highly correlated with SalePrice than GarageArea, so GarageArea will be dropped.

YearBuilt is more highly correlated with SalePrice than GarageYrBlt, so GarageYrBlt will be dropped.

GrLivArea is more highly correlated with SalePrice than TotRmsAbvGrd, so TotRmsAbvGrd will be dropped.

TotBsmtSF is more highly correlated with SalePrice than 1stFlrSF, so 1stFlrSF will be dropped.

```
In [524... # Dropping 'Id' column
train_df.drop('Id', axis=1, inplace=True)

In [525... # Dropping predictors with too many null values
train_df.drop(['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature'], axis=1, inplace=True)

In [526... # Dropping multicollinear predictors
train_df.drop(['GarageArea', 'GarageYrBlt', 'TotRmsAbvGrd', '1stFlrSF'], axis=1, inplace=True)

In [527... # Dropping low variance predictors, because they will have little effect on the
var = train_df.var().sort_values(ascending=True)
low_var = var[var < .1]
low_var = list(low_var.index)
train_df.drop(low_var, axis=1, inplace=True)

In [528... # Dropping predictors not linearly related to the target
train_df.drop('MSSubClass', axis=1, inplace=True)

In [529... # Dropping remaining null values
train_df.dropna(inplace=True)

In [530... # Turning these predictors from categorical into numerical
map_function(train_df, ['ExterQual', 'ExterCond', 'BsmtQual', 'BsmtCond', 'Heati

In [531... # Combining the weaker mid_var predictors to try and give them more predictive p
train_df['Bath_Fireplaces'] = train_df['Fireplaces'] + train_df['FullBath']

In [532... train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1412 entries, 0 to 1459
Data columns (total 69 columns):
#   Column              Non-Null Count  Dtype
0   MSZoning             1412 non-null   object
```

1	LotFrontage	1412	non-null	float64
2	LotArea	1412	non-null	int64
3	Street	1412	non-null	object
4	LotShape	1412	non-null	object
5	LandContour	1412	non-null	object
6	Utilities	1412	non-null	object
7	LotConfig	1412	non-null	object
8	LandSlope	1412	non-null	object
9	Neighborhood	1412	non-null	object
10	Condition1	1412	non-null	object
11	Condition2	1412	non-null	object
12	BldgType	1412	non-null	object
13	HouseStyle	1412	non-null	object
14	OverallQual	1412	non-null	int64
15	OverallCond	1412	non-null	int64
16	YearBuilt	1412	non-null	int64
17	YearRemodAdd	1412	non-null	int64
18	RoofStyle	1412	non-null	object
19	RoofMatl	1412	non-null	object
20	Exterior1st	1412	non-null	object
21	Exterior2nd	1412	non-null	object
22	MasVnrType	1412	non-null	object
23	MasVnrArea	1412	non-null	float64
24	ExterQual	1412	non-null	int64
25	ExterCond	1412	non-null	int64
26	Foundation	1412	non-null	object
27	BsmtQual	1412	non-null	int64
28	BsmtCond	1412	non-null	int64
29	BsmtExposure	1412	non-null	object
30	BsmtFinType1	1412	non-null	object
31	BsmtFinSF1	1412	non-null	int64
32	BsmtFinType2	1412	non-null	object
33	BsmtFinSF2	1412	non-null	int64
34	BsmtUnfSF	1412	non-null	int64
35	TotalBsmtSF	1412	non-null	int64
36	Heating	1412	non-null	object
37	HeatingQC	1412	non-null	int64
38	CentralAir	1412	non-null	object
39	Electrical	1412	non-null	object
40	2ndFlrSF	1412	non-null	int64
41	LowQualFinSF	1412	non-null	int64
42	GrLivArea	1412	non-null	int64
43	BsmtFullBath	1412	non-null	int64
44	FullBath	1412	non-null	int64
45	HalfBath	1412	non-null	int64
46	BedroomAbvGr	1412	non-null	int64
47	KitchenQual	1412	non-null	int64
48	Functional	1412	non-null	object
49	Fireplaces	1412	non-null	int64
50	GarageType	1412	non-null	object
51	GarageFinish	1412	non-null	object
52	GarageCars	1412	non-null	int64
53	GarageQual	1412	non-null	object
54	GarageCond	1412	non-null	object
55	PavedDrive	1412	non-null	object
56	WoodDeckSF	1412	non-null	int64
57	OpenPorchSF	1412	non-null	int64
58	EnclosedPorch	1412	non-null	int64
59	3SsnPorch	1412	non-null	int64
60	ScreenPorch	1412	non-null	int64
61	PoolArea	1412	non-null	int64
62	MiscVal	1412	non-null	int64
63	MoSold	1412	non-null	int64
65	SaleType	1412	non-null	object

```

66 SaleCondition      1412 non-null    object
67 SalePrice          1412 non-null    int64
68 Bath_Fireplaces    1412 non-null    int64
dtypes: float64(2), int64(35), object(32)
memory usage: 772.2+ KB

```

```
In [533... train_df.head()
```

```

Out[533...
   MSZoning  LotFrontage  LotArea  Street  LotShape  LandContour  Utilities  LotConfig  LandSlo
0         RL          65.0     8450   Pave      Reg           Lvl      AllPub     Inside
1         RL          80.0     9600   Pave      Reg           Lvl      AllPub      FR2
2         RL          68.0    11250   Pave      IR1           Lvl      AllPub     Inside
3         RL          60.0     9550   Pave      IR1           Lvl      AllPub    Corner
4         RL          84.0    14260   Pave      IR1           Lvl      AllPub      FR2

```

5 rows × 69 columns

Modeling

Baseline Models

*disclaimer **

It is hard to say with any certainty how the values imputed to the lot frontage variable will effect the results of the models

Note: I am using recursive feature elimination prior to fitting

```
In [578... x_train = train_df.drop(['SalePrice'], axis=1)
           y_train = train_df['SalePrice']
```

```

In [579... categorical_cols = x_train.select_dtypes('object')
           numerical_cols = x_train.select_dtypes(exclude=['object'])

           numerical_transformer = Pipeline(steps=[('ss', StandardScaler())])
           categorical_transformer = Pipeline(steps=[('ohe', OneHotEncoder(handle_unknown='
           transformer = ColumnTransformer(transformers=[('numerical', numerical_transforme
                   ('categorical', categorical_transf
                   ])

           #cat_transformer = ColumnTransformer(transformers=[('categorical', categorical_t

```

```

In [580... x_tr, x_val, y_tr, y_val = train_test_split(x_train, y_train, test_size=0.25, ra

           # Transforming x_tr and x_val separately to avoid data leakage
           x_tr = transformer.fit_transform(x_tr)
           x_val = transformer.transform(x_val)

```



```

# Reducing Overfitting
rfe = RFE(LinearRegression(normalize=False), n_features_to_select=75)
x_tr = rfe.fit_transform(x_tr, y_tr)
x_val = rfe.transform(x_val)

# Instantiating and fitting model
linreg = LinearRegression(normalize=False)
results = linreg.fit(x_tr, y_tr)

# Getting Predictions
y_tr_preds = results.predict(x_tr)
y_val_preds = results.predict(x_val)

```

```
In [581... get_results_preds(y_tr, y_tr_preds, y_val, y_val_preds)
```

R-Squared score for the training data: 0.9011203723473886

R-Squared score for the testing data: 0.8108101837772864

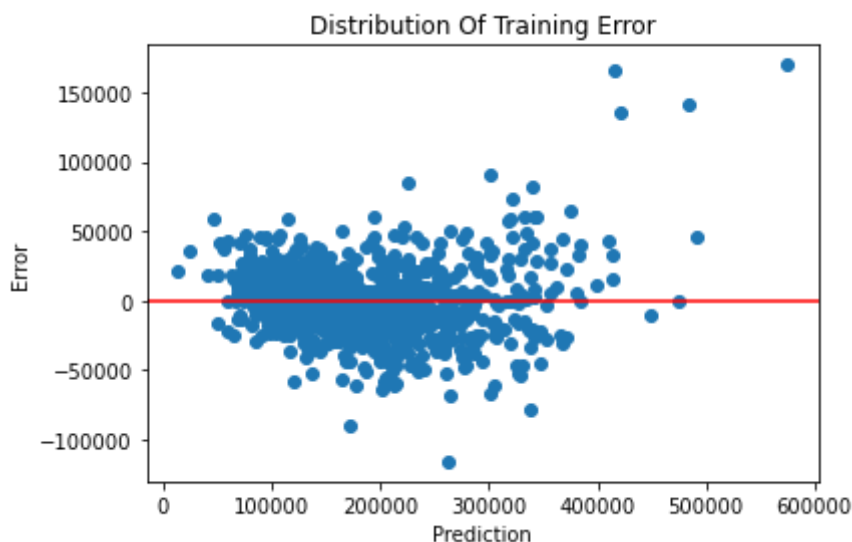
Root Mean Squared Error for the training data: 24378.368158032215

Root Mean Squared Error for the testing data: 36611.07878134237

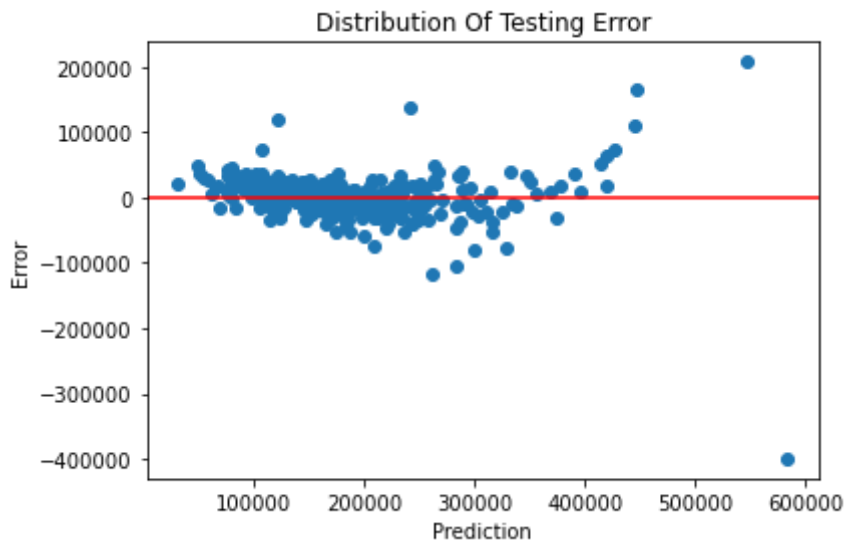
Analysis:

The baseline model using recursive feature elimination appears to have fit fairly well. I am going to check for homoskedasticity and then cross validate the model.

```
In [582... train_error_distribution(y_tr, y_tr_preds)
```



```
In [583... test_error_distribution(y_val, y_val_preds)
```



It appears that for the most part, the test residuals pass the check for homoscedasticity

```
In [584... rfe = RFE(LinearRegression(normalize=False), n_features_to_select=75)
x_train = transformer.fit_transform(x_train)
x_train = rfe.fit_transform(x_train, y_train)
cv = cross_validate(linreg, x_train, y_train, scoring='r2', cv=5, return_train_s
get_cv_results(cv)
```

```
{'fit_time': array([0.03184414, 0.04104424, 0.03518486, 0.03032088, 0.0344800
9]), 'score_time': array([0.00073886, 0.00060964, 0.00075006, 0.00127196, 0.0005
3477]), 'test_score': array([0.87173715, 0.82297199, 0.80113393, 0.86038515, 0.6
8988316]), 'train_score': array([0.88614528, 0.89213888, 0.89983384, 0.89158741,
0.89097417])}
```

CV Train Scores: [0.88614528 0.89213888 0.89983384 0.89158741 0.89097417]

Mean CV Train Score: 0.8921359185194845

CV Test Scores: [0.87173715 0.82297199 0.80113393 0.86038515 0.68988316]

Mean CV Test Score: 0.8092222755302014

Linear Regression With L2 Regularization

```
In [481... x_train = train_df.drop(['SalePrice'], axis=1)
y_train = train_df['SalePrice']
```

```
In [45]: ridge_pipeline = Pipeline(steps=[('trans', transformer),
                                          ('rfe', RFE(LinearRegression(normalize=False),
                                          ('model', Ridge(random_state=42)))]

ridge_grid = {'model__alpha': [10.0, 25.0, 50.0, 75.0],
              'model__normalize': [False],
              'model__max_iter': [500, 2000],
              'model__tol': [.001, .0001],
              'model__solver': ['auto']}

ridge_gs = GridSearchCV(ridge_pipeline,
                        ridge_grid,
                        scoring=['neg_root_mean_squared_error', 'r2'],
                        cv=5,
```

```
verbose=3,  
return_train_score=True)
```

```
In [46]: results = ridge_gs.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001  
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31217.429, test=-25914.247), r2=(train=0.849, test=0.879), total= 12.7s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001  
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 12.7s remaining: 0.0s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-30524.788, test=-33537.502), r2=(train=0.850, test=0.828), total= 13.5s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001  
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 26.3s remaining: 0.0s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-30825.116, test=-34914.931), r2=(train=0.840, test=0.839), total= 14.0s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31303.677, test=-28989.172), r2=(train=0.850, test=0.838), total= 12.0s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-27152.855, test=-47557.427), r2=(train=0.882, test=0.650), total= 11.6s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31203.493, test=-25956.402), r2=(train=0.849, test=0.878), total= 12.2s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-30502.807, test=-33546.749), r2=(train=0.850, test=0.828), total= 13.5s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-30816.955, test=-34922.224), r2=(train=0.840, test=0.839), total= 13.9s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31299.829, test=-28984.641), r2=(train=0.850, test=0.838), total= 14.3s  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001  
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-27137.097, test=-47567.727), r2=(train=0.882, test=0.650), total= 12.5s  
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001  
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31217.429, test
```

```

=-25914.247), r2=(train=0.849, test=0.879), total= 12.3s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-30524.788, test
=-33537.502), r2=(train=0.850, test=0.828), total= 14.5s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-30825.116, test
=-34914.931), r2=(train=0.840, test=0.839), total= 14.8s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31303.677, test
=-28989.172), r2=(train=0.850, test=0.838), total= 12.2s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-27152.855, test
=-47557.427), r2=(train=0.882, test=0.650), total= 11.5s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31203.493, tes
t=-25956.402), r2=(train=0.849, test=0.878), total= 12.8s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-30502.807, tes
t=-33546.749), r2=(train=0.850, test=0.828), total= 14.2s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-30816.955, tes
t=-34922.224), r2=(train=0.840, test=0.839), total= 15.1s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31299.829, tes
t=-28984.641), r2=(train=0.850, test=0.838), total= 12.6s
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-27137.097, tes
t=-47567.727), r2=(train=0.882, test=0.650), total= 13.2s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32620.709, test=
-26404.022), r2=(train=0.835, test=0.874), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31954.020, test=
-33696.401), r2=(train=0.836, test=0.827), total= 14.6s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32384.509, test=
-35787.114), r2=(train=0.824, test=0.831), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32874.977, test=

```

```

-29214.511), r2=(train=0.835, test=0.836), total= 12.2s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-28267.161, test=-47972.894), r2=(train=0.872, test=0.644), total= 13.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32612.837, test=-26385.103), r2=(train=0.835, test=0.874), total= 13.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31948.355, test=-33695.402), r2=(train=0.836, test=0.827), total= 13.7s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32378.537, test=-35810.237), r2=(train=0.824, test=0.831), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32870.346, test=-29208.105), r2=(train=0.835, test=0.836), total= 12.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-28258.292, test=-48009.875), r2=(train=0.872, test=0.643), total= 12.3s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32620.709, test=-26404.022), r2=(train=0.835, test=0.874), total= 12.4s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-31954.020, test=-33696.401), r2=(train=0.836, test=0.827), total= 13.6s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32384.509, test=-35787.114), r2=(train=0.824, test=0.831), total= 16.0s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-32874.977, test=-29214.511), r2=(train=0.835, test=0.836), total= 13.6s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-28267.161, test=-47972.894), r2=(train=0.872, test=0.644), total= 14.5s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32612.837, test=-26385.103), r2=(train=0.835, test=0.874), total= 14.4s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-31948.355, tes

```

```

t=-33695.402), r2=(train=0.836, test=0.827), total= 14.9s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32378.537, tes
t=-35810.237), r2=(train=0.824, test=0.831), total= 14.2s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-32870.346, tes
t=-29208.105), r2=(train=0.835, test=0.836), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-28258.292, tes
t=-48009.875), r2=(train=0.872, test=0.643), total= 11.7s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33840.555, test=
-27109.357), r2=(train=0.823, test=0.867), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33146.573, test=
-34070.566), r2=(train=0.823, test=0.823), total= 15.1s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33628.201, test=
-36957.953), r2=(train=0.810, test=0.820), total= 15.0s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34215.885, test=
-29744.047), r2=(train=0.821, test=0.830), total= 12.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-29298.070, test=
-48484.510), r2=(train=0.862, test=0.636), total= 12.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33832.205, test
=-27092.654), r2=(train=0.823, test=0.867), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33143.478, test
=-34074.794), r2=(train=0.823, test=0.823), total= 13.8s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33626.272, test
=-36955.917), r2=(train=0.810, test=0.820), total= 13.9s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34213.889, test
=-29739.909), r2=(train=0.821, test=0.830), total= 12.0s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-29289.614, test

```

```

=-48521.663), r2=(train=0.862, test=0.636), total= 11.5s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33840.555, test
=-27109.357), r2=(train=0.823, test=0.867), total= 12.7s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33146.573, test
=-34070.566), r2=(train=0.823, test=0.823), total= 14.7s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33628.201, test
=-36957.953), r2=(train=0.810, test=0.820), total= 14.1s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34215.885, test
=-29744.047), r2=(train=0.821, test=0.830), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-29298.070, test
=-48484.510), r2=(train=0.862, test=0.636), total= 11.5s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33832.205, tes
t=-27092.654), r2=(train=0.823, test=0.867), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33143.478, tes
t=-34074.794), r2=(train=0.823, test=0.823), total= 14.1s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33626.272, tes
t=-36955.917), r2=(train=0.810, test=0.820), total= 16.1s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34213.889, tes
t=-29739.909), r2=(train=0.821, test=0.830), total= 13.0s
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__sol
ver=auto, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=2000, model__normalize=False, model__so
lver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-29289.614, tes
t=-48521.663), r2=(train=0.862, test=0.636), total= 14.6s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34601.366, test=
-27671.529), r2=(train=0.815, test=0.862), total= 14.3s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33897.761, test=
-34410.237), r2=(train=0.815, test=0.819), total= 14.6s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solv
er=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__sol
ver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34388.950, test=

```

```

-37780.595), r2=(train=0.801, test=0.812), total= 13.9s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-35063.693, test=-30192.320), r2=(train=0.812, test=0.824), total= 12.1s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-29971.263, test=-48793.628), r2=(train=0.856, test=0.631), total= 11.6s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34599.893, test=-27668.287), r2=(train=0.815, test=0.862), total= 12.4s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33894.278, test=-34407.056), r2=(train=0.815, test=0.819), total= 13.7s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34387.190, test=-37773.169), r2=(train=0.801, test=0.812), total= 14.8s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-35062.161, test=-30198.001), r2=(train=0.812, test=0.824), total= 12.0s
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=500, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-29969.651, test=-48805.769), r2=(train=0.856, test=0.631), total= 11.6s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34601.366, test=-27671.529), r2=(train=0.815, test=0.862), total= 13.0s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-33897.761, test=-34410.237), r2=(train=0.815, test=0.819), total= 17.3s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-34388.950, test=-37780.595), r2=(train=0.801, test=0.812), total= 17.4s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-35063.693, test=-30192.320), r2=(train=0.812, test=0.824), total= 14.0s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.001, neg_root_mean_squared_error=(train=-29971.263, test=-48793.628), r2=(train=0.856, test=0.631), total= 11.5s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34599.893, tes

```



```
t=-27668.287), r2=(train=0.815, test=0.862), total= 14.3s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-33894.278, test=-34407.056), r2=(train=0.815, test=0.819), total= 13.7s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-34387.190, test=-37773.169), r2=(train=0.801, test=0.812), total= 14.3s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-35062.161, test=-30198.001), r2=(train=0.812, test=0.824), total= 14.1s
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001
[CV] model__alpha=75.0, model__max_iter=2000, model__normalize=False, model__solver=auto, model__tol=0.0001, neg_root_mean_squared_error=(train=-29969.651, test=-48805.769), r2=(train=0.856, test=0.631), total= 13.8s
[Parallel(n_jobs=1)]: Done 80 out of 80 | elapsed: 17.9min finished
```

```
In [47]: results.best_params_
```

```
Out[47]: {'model__alpha': 10.0,
          'model__max_iter': 500,
          'model__normalize': False,
          'model__solver': 'auto',
          'model__tol': 0.001}
```

```
In [48]: results.best_score_
```

```
Out[48]: -34182.65570655737
```

```
In [49]: get_results(results)
```

```
The best model parameters produce a mean rmse score on train data of:
-30204.77303842335
```

```
The best model parameters produce a mean R-squared score on train data of:
0.8542529575044672
```

```
The best model parameters produce a mean rmse score on test data of:
-34182.65570655737
```

```
The best model parameters produce a mean R-squared score on test data of:
0.8068734112651119
```

Analysis:

The best parameters for this Ridge regression(linear regression with l2 regularization) were an alpha value of 10.0, maximum iterations of 500, and .001 tol, the default tolerance value.

The best model is slightly overfit to the training data with relation to the validation data, but not too overfit in the broad scope.

Linear Regressions With l1 Regularization

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
```

```
x_train = train_df.drop(['SalePrice'], axis=1)
```

```
In [53]: y_train = train_df['SalePrice']
```

```
In [56]: lasso_pipeline = Pipeline(steps=[('trans', transformer),
                                           ('rfe', RFE(LinearRegression(normalize=False),
                                           ('model', Lasso(random_state=62)))]

lasso_grid = {'model__alpha': [10.0, 25.0, 50.0],
              'model__normalize': [False],
              'model__max_iter': [500, 1000],
              'model__tol': [.001, .0001],
              'model__selection': ['cyclic', 'random']}

lasso_gs = GridSearchCV(lasso_pipeline,
                        lasso_grid,
                        scoring=['neg_root_mean_squared_error', 'r2'],
                        refit='neg_root_mean_squared_error',
                        cv=5,
                        verbose=3,
                        return_train_score=True)
```

```
In [57]: lasso_results = lasso_gs.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26143.069, test=-26060.877), r2=(train=0.894, test=0.877), total= 12.4s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 12.4s remaining: 0.0s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25728.947, test=-35061.192), r2=(train=0.893, test=0.812), total= 13.7s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 26.2s remaining: 0.0s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-23926.139, test=-42020.850), r2=(train=0.904, test=0.767), total= 14.2s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26299.722, test=-29690.450), r2=(train=0.894, test=0.830), total= 12.1s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25098.740, test=-48333.786), r2=(train=0.899, test=0.638), total= 11.5s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26143.174, test=-26054.153), r2=(train=0.894, test=0.877), total= 12.4s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25728.605, test=0.812), total= 13.5s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25728.605, test=0.812), total= 13.5s
```

```

ction=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-23925.210,
test=-42044.563), r2=(train=0.904, test=0.767), total= 13.9s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26299.450,
test=-29683.108), r2=(train=0.894, test=0.830), total= 12.7s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25097.907,
test=-48338.065), r2=(train=0.899, test=0.638), total= 12.9s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26143.213,
test=-26060.646), r2=(train=0.894, test=0.877), total= 12.4s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25729.028,
test=-35060.984), r2=(train=0.893, test=0.812), total= 13.8s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-23926.372,
test=-42017.930), r2=(train=0.904, test=0.767), total= 14.3s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26298.771,
test=-29689.721), r2=(train=0.894, test=0.830), total= 12.2s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25098.509,
test=-48336.513), r2=(train=0.899, test=0.638), total= 13.1s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26143.205,
test=-26062.218), r2=(train=0.894, test=0.877), total= 12.8s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25728.609,
test=-35063.260), r2=(train=0.893, test=0.812), total= 14.0s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-23925.481,
test=-42038.228), r2=(train=0.904, test=0.767), total= 14.2s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26299.294,
test=-29682.467), r2=(train=0.894, test=0.830), total= 12.0s
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25097.888,
test=0.638), total= 11.7s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel

```

```

ection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26143.069, test=-26060.877), r2=(train=0.894, test=0.877), total= 12.9s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25728.947, test=-35061.192), r2=(train=0.893, test=0.812), total= 13.7s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-23926.139, test=-42020.850), r2=(train=0.904, test=0.767), total= 13.9s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26299.722, test=-29690.450), r2=(train=0.894, test=0.830), total= 11.9s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25098.740, test=-48333.786), r2=(train=0.899, test=0.638), total= 11.8s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26143.174, test=-26054.153), r2=(train=0.894, test=0.877), total= 12.4s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25728.605, test=-35063.254), r2=(train=0.893, test=0.812), total= 14.3s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-23925.210, test=-42044.563), r2=(train=0.904, test=0.767), total= 13.9s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26299.450, test=-29683.108), r2=(train=0.894, test=0.830), total= 12.0s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25097.907, test=-48338.065), r2=(train=0.899, test=0.638), total= 11.4s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26143.213, test=-26060.646), r2=(train=0.894, test=0.877), total= 12.6s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25729.028, test=-35060.984), r2=(train=0.893, test=0.812), total= 14.2s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-23926.372, test=0.767), total= 14.6s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__selection=

```

```

ection=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26298.771,
test=-29689.721), r2=(train=0.894, test=0.830), total= 16.1s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25098.509,
test=-48336.513), r2=(train=0.899, test=0.638), total= 13.4s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26143.20
5, test=-26062.218), r2=(train=0.894, test=0.877), total= 15.2s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25728.60
9, test=-35063.260), r2=(train=0.893, test=0.812), total= 15.7s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-23925.26
6, test=-42043.607), r2=(train=0.904, test=0.767), total= 14.4s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26299.29
4, test=-29682.467), r2=(train=0.894, test=0.830), total= 12.1s
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__sel
ection=random, model__tol=0.0001
[CV] model__alpha=10.0, model__max_iter=1000, model__normalize=False, model__se
lection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25097.88
8, test=-48338.366), r2=(train=0.899, test=0.638), total= 11.7s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26326.500,
test=-25852.840), r2=(train=0.893, test=0.879), total= 12.6s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25990.260,
test=-34785.669), r2=(train=0.891, test=0.815), total= 13.9s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-24263.358,
test=-40039.880), r2=(train=0.901, test=0.789), total= 14.5s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26523.096,
test=-29271.486), r2=(train=0.892, test=0.835), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25362.209,
test=-47987.312), r2=(train=0.897, test=0.644), total= 11.5s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26327.093,
test=0.879), total= 12.4s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele

```

```

ction=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25990.427,
test=-34787.336), r2=(train=0.891, test=0.815), total= 13.9s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-24258.532,
test=-40076.212), r2=(train=0.901, test=0.788), total= 14.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26522.662,
test=-29272.273), r2=(train=0.892, test=0.835), total= 12.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25359.573,
test=-47994.448), r2=(train=0.897, test=0.643), total= 11.5s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26327.311,
test=-25836.390), r2=(train=0.893, test=0.879), total= 12.4s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25992.837,
test=-34789.476), r2=(train=0.891, test=0.815), total= 14.0s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-24264.195,
test=-40035.464), r2=(train=0.901, test=0.789), total= 13.8s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26522.104,
test=-29270.425), r2=(train=0.892, test=0.835), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25362.586,
test=-47985.344), r2=(train=0.897, test=0.644), total= 11.8s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26327.227,
test=-25838.865), r2=(train=0.893, test=0.879), total= 13.4s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25991.635,
test=-34786.937), r2=(train=0.891, test=0.815), total= 14.3s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-24258.713,
test=-40075.207), r2=(train=0.901, test=0.788), total= 14.6s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26522.548,
test=0.835), total= 12.1s
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__sele

```

```

ction=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=500, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25359.605, test=-47994.151), r2=(train=0.897, test=0.643), total= 11.5s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26326.500, test=-25852.840), r2=(train=0.893, test=0.879), total= 12.6s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25990.260, test=-34785.669), r2=(train=0.891, test=0.815), total= 13.7s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-24263.358, test=-40039.880), r2=(train=0.901, test=0.789), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26523.096, test=-29271.486), r2=(train=0.892, test=0.835), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25362.209, test=-47987.312), r2=(train=0.897, test=0.644), total= 12.2s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26327.093, test=-25841.132), r2=(train=0.893, test=0.879), total= 12.9s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25990.427, test=-34787.336), r2=(train=0.891, test=0.815), total= 13.8s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-24258.532, test=-40076.212), r2=(train=0.901, test=0.788), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26522.662, test=-29272.273), r2=(train=0.892, test=0.835), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25359.573, test=-47994.448), r2=(train=0.897, test=0.643), total= 11.5s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26327.311, test=-25836.390), r2=(train=0.893, test=0.879), total= 12.3s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25992.837, test=0.815), total= 14.0s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25992.837, test=0.815), total= 14.0s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25992.837, test=0.815), total= 14.0s

```

```

ection=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-24264.195, test=-40035.464), r2=(train=0.901, test=0.789), total= 14.7s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26522.104, test=-29270.425), r2=(train=0.892, test=0.835), total= 11.9s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25362.586, test=-47985.344), r2=(train=0.897, test=0.644), total= 11.4s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26327.227, test=-25838.865), r2=(train=0.893, test=0.879), total= 12.5s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25991.635, test=-34786.937), r2=(train=0.891, test=0.815), total= 13.8s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-24258.713, test=-40075.207), r2=(train=0.901, test=0.788), total= 14.1s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26522.548, test=-29271.768), r2=(train=0.892, test=0.835), total= 12.0s
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=25.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25359.605, test=-47994.151), r2=(train=0.897, test=0.643), total= 11.7s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26744.739, test=-25533.631), r2=(train=0.889, test=0.882), total= 12.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26365.069, test=-34389.061), r2=(train=0.888, test=0.820), total= 14.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-24846.652, test=-38022.554), r2=(train=0.896, test=0.809), total= 14.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26956.076, test=-28929.463), r2=(train=0.889, test=0.839), total= 12.0s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.443, test=0.650), total= 11.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.443, test=0.650), total= 11.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.443, test=0.650), total= 11.5s

```



```

ction=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26747.370,
test=-25516.496), r2=(train=0.889, test=0.882), total= 12.6s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26365.171,
test=-34389.347), r2=(train=0.888, test=0.820), total= 13.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-24852.127,
test=-38001.387), r2=(train=0.896, test=0.810), total= 14.7s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26955.066,
test=-28931.009), r2=(train=0.889, test=0.839), total= 12.0s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25803.486,
test=-47535.761), r2=(train=0.893, test=0.650), total= 11.4s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26748.208,
test=-25510.998), r2=(train=0.889, test=0.882), total= 12.4s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26364.872,
test=-34391.098), r2=(train=0.888, test=0.819), total= 13.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-24851.077,
test=-38008.070), r2=(train=0.896, test=0.810), total= 14.1s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26955.758,
test=-28930.160), r2=(train=0.889, test=0.839), total= 11.9s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.035,
test=-47533.475), r2=(train=0.893, test=0.650), total= 11.5s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26747.841,
test=-25513.718), r2=(train=0.889, test=0.882), total= 15.1s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26365.082,
test=-34389.961), r2=(train=0.888, test=0.820), total= 18.3s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele
ction=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sel
ection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-24852.779,
test=0.810), total= 18.4s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__sele

```

```

ction=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26954.987, test=-28931.094), r2=(train=0.889, test=0.839), total= 16.2s
[CV] model__alpha=50.0, model__max_iter=500, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25803.306, test=-47535.539), r2=(train=0.893, test=0.650), total= 15.6s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26744.739, test=-25533.631), r2=(train=0.889, test=0.882), total= 13.4s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26365.069, test=-34389.061), r2=(train=0.888, test=0.820), total= 13.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-24846.652, test=-38022.554), r2=(train=0.896, test=0.809), total= 14.2s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-26956.076, test=-28929.463), r2=(train=0.889, test=0.839), total= 12.0s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.443, test=-47533.410), r2=(train=0.893, test=0.650), total= 11.4s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26747.370, test=-25516.496), r2=(train=0.889, test=0.882), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26365.171, test=-34389.347), r2=(train=0.888, test=0.820), total= 13.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-24852.127, test=-38001.387), r2=(train=0.896, test=0.810), total= 14.4s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-26955.066, test=-28931.009), r2=(train=0.889, test=0.839), total= 12.0s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=cyclic, model__tol=0.0001, neg_root_mean_squared_error=(train=-25803.486, test=-47535.761), r2=(train=0.893, test=0.650), total= 11.9s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26748.208, test=0.882), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26748.208, test=0.882), total= 12.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26748.208, test=0.882), total= 12.5s

```

```

lection=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26364.872, test=-34391.098), r2=(train=0.888, test=0.819), total= 13.7s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-24851.077, test=-38008.070), r2=(train=0.896, test=0.810), total= 13.9s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-26955.758, test=-28930.160), r2=(train=0.889, test=0.839), total= 12.2s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.001, neg_root_mean_squared_error=(train=-25803.035, test=-47533.475), r2=(train=0.893, test=0.650), total= 11.4s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26747.841, test=-25513.718), r2=(train=0.889, test=0.882), total= 12.7s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26365.082, test=-34389.961), r2=(train=0.888, test=0.820), total= 13.5s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-24852.779, test=-37999.384), r2=(train=0.896, test=0.810), total= 16.6s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-26954.987, test=-28931.094), r2=(train=0.889, test=0.839), total= 16.1s
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001
[CV] model__alpha=50.0, model__max_iter=1000, model__normalize=False, model__selection=random, model__tol=0.0001, neg_root_mean_squared_error=(train=-25803.306, test=-47535.539), r2=(train=0.893, test=0.650), total= 15.1s
[Parallel(n_jobs=1)]: Done 120 out of 120 | elapsed: 26.5min finished

```

```
In [58]: lasso_results.best_params_
```

```
Out[58]: {'model__alpha': 50.0,
          'model__max_iter': 500,
          'model__normalize': False,
          'model__selection': 'random',
          'model__tol': 0.0001}
```

```
In [59]: get_results(lasso_results)
```

The best model parameters produce a mean rmse score on train data of:
-26144.798791128233

The best model parameters produce a mean R-squared score on train data of:
0.8911157755585011

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js mean rmse score on test data of:

The best model parameters produce a mean R-squared score on test data of:
0.8001131523090838

Analysis:

The best parameters for this Lasso regression(linear regression with l1 regularization) were an alpha value of 50.0, maximum iterations of 500, and a tolerance of .0001 tol.

The best model is fairly overfit to the training data with relation to the validation data, though only slightly overfit broadly.

Note: I tested an elasticnet regression in a scratch notebook and it performed terribly, so I am not running it here. If you want to see the results, please reference the other notebook.

Decision Tree Regression

```
In [60]: x_train = train_df.drop(['SalePrice'], axis=1)
         y_train = train_df['SalePrice']
```

```
In [61]: dt_pipeline = Pipeline(steps=[('trans', transformer),
                                       ('rfe', RFE(LinearRegression(normalize=False), n_f
                                       ('model', DecisionTreeRegressor(random_state=92)))]

dt_grid = {'model__criterion': ['mse', 'mae'],
          'model__splitter': ['best'],
          'model__max_depth': [None, 10],
          'model__min_samples_split': [2, 12],
          'model__min_samples_leaf': [1, 3],
          'model__max_features': ['auto']}

dt_gs = GridSearchCV(dt_pipeline,
                    dt_grid,
                    scoring=['neg_root_mean_squared_error', 'r2'],
                    refit='neg_root_mean_squared_error',
                    cv=5,
                    verbose=3,
                    return_train_score=True)
```

```
In [62]: dt_results = dt_gs.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
[CV] model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-268.614, test=-39189.359), r2=(train=1.000, test=0.723), total= 13.9s
[CV] model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 13.9s remaining: 0.0s
[CV] model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-268.614, test=-51339.711), r2=(train=1.000, test=
```



```

el_min_samples_leaf=3, model_min_samples_split=2, model_splitter=best, neg_ro
ot_mean_squared_error=(train=-19029.962, test=-46838.185), r2=(train=0.939, test
=0.711), total= 14.3s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=2, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=2, model_splitter=best, neg_ro
ot_mean_squared_error=(train=-21792.433, test=-33652.965), r2=(train=0.927, test
=0.782), total= 11.8s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=2, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=2, model_splitter=best, neg_ro
ot_mean_squared_error=(train=-18500.033, test=-43924.474), r2=(train=0.945, test
=0.701), total= 12.3s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best, neg_r
oot_mean_squared_error=(train=-20120.453, test=-37626.285), r2=(train=0.937, tes
t=0.744), total= 12.5s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best, neg_r
oot_mean_squared_error=(train=-22534.753, test=-40769.342), r2=(train=0.918, tes
t=0.746), total= 13.7s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best, neg_r
oot_mean_squared_error=(train=-22839.347, test=-42528.074), r2=(train=0.912, tes
t=0.762), total= 14.2s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best, neg_r
oot_mean_squared_error=(train=-24728.612, test=-31939.485), r2=(train=0.906, tes
t=0.804), total= 12.1s
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mode
l_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best
[CV] model_criterion=mse, model_max_depth=None, model_max_features=auto, mod
el_min_samples_leaf=3, model_min_samples_split=12, model_splitter=best, neg_r
oot_mean_squared_error=(train=-22614.464, test=-40793.551), r2=(train=0.918, tes
t=0.742), total= 11.8s
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model_
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best, neg_root
_mean_squared_error=(train=-12654.920, test=-39350.907), r2=(train=0.975, test=
0.720), total= 12.5s
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model_
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best, neg_root
_mean_squared_error=(train=-11509.210, test=-34199.546), r2=(train=0.979, test=
0.821), total= 14.0s
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model_
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best, neg_root
_mean_squared_error=(train=-14161.032, test=-35240.820), r2=(train=0.966, test=
0.836), total= 14.2s
[CV] model_criterion=mse, model_max_depth=10, model_max_features=auto, model_
_min_samples_leaf=1, model_min_samples_split=2, model_splitter=best

```

```

[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-14538.939, test=-39911.464), r2=(train=0.968, test=0.693), total= 12.0s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-12270.395, test=-57138.510), r2=(train=0.976, test=0.495), total= 11.5s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-19206.010, test=-39133.680), r2=(train=0.943, test=0.724), total= 12.6s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-20349.445, test=-38210.319), r2=(train=0.933, test=0.777), total= 13.7s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-20058.445, test=-36595.657), r2=(train=0.932, test=0.823), total= 14.0s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-22556.534, test=-31208.700), r2=(train=0.922, test=0.812), total= 12.2s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-18707.109, test=-52859.165), r2=(train=0.944, test=0.567), total= 13.3s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-19248.384, test=-37119.892), r2=(train=0.943, test=0.751), total= 15.2s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-20474.230, test=-39082.633), r2=(train=0.932, test=0.767), total= 15.6s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-21212.785, test=-46116.268), r2=(train=0.924, test=0.720), total= 14.0s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-23230.077, test=-32776.497), r2=(train=0.917, test=

```

```

_min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-20612.698, test=-44143.395), r2=(train=0.932, test=
0.698), total= 11.4s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-21470.631, test=-37594.772), r2=(train=0.929, test=
0.745), total= 12.4s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-23072.598, test=-40556.881), r2=(train=0.914, test=
0.749), total= 14.1s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-23959.443, test=-42316.949), r2=(train=0.904, test=
0.764), total= 14.1s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-25359.081, test=-31742.432), r2=(train=0.902, test=
0.806), total= 11.9s
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mse, model__max_depth=10, model__max_features=auto, model
_min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-23555.594, test=-41047.435), r2=(train=0.911, test=
0.739), total= 11.7s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mode
l__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mod
el__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_ro
ot_mean_squared_error=(train=-268.614, test=-36905.484), r2=(train=1.000, test=
0.754), total= 12.8s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mode
l__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mod
el__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_ro
ot_mean_squared_error=(train=-685.680, test=-48507.701), r2=(train=1.000, test=
0.641), total= 14.1s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mode
l__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mod
el__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_ro
ot_mean_squared_error=(train=-374.662, test=-41113.049), r2=(train=1.000, test=
0.777), total= 14.4s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mode
l__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mod
el__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_ro
ot_mean_squared_error=(train=-655.318, test=-42771.555), r2=(train=1.000, test=
0.648), total= 12.0s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mode
l__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, mod
el__min samples leaf=1, model__min samples split=2, model__splitter=best, neg_ro
ot_mean_squared_error=(train=-49040.064), r2=(train=1.000, test=
0.628), total= 11.7s

```



```

[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-16200.312, test=-35114.747), r2=(train=0.959, test=0.777), total= 12.6s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-24722.516, test=-43457.551), r2=(train=0.902, test=0.712), total= 14.3s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-16303.975, test=-37601.338), r2=(train=0.955, test=0.814), total= 14.3s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-19335.905, test=-39594.299), r2=(train=0.943, test=0.698), total= 12.4s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-17547.685, test=-50793.669), r2=(train=0.951, test=0.601), total= 11.8s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-19122.664, test=-35359.731), r2=(train=0.943, test=0.774), total= 13.1s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-21858.345, test=-43346.192), r2=(train=0.923, test=0.713), total= 14.8s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-22921.096, test=-37193.963), r2=(train=0.912, test=0.818), total= 14.1s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-24559.197, test=-34501.102), r2=(train=0.908, test=0.771), total= 12.2s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-21878.217, test=-43961.956), r2=(train=0.923, test=0.701), total= 11.7s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-23099.149, test=-36077.442), r2=(train=0.917, test=0.777), total= 12.2s

```

```

t=0.765), total= 12.6s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-27217.420, test=-44428.154), r2=(train=0.881, test=0.699), total= 14.6s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-25492.010, test=-37569.146), r2=(train=0.891, test=0.814), total= 14.3s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-26987.509, test=-31767.958), r2=(train=0.889, test=0.806), total= 12.3s
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-24494.714, test=-43834.373), r2=(train=0.904, test=0.703), total= 11.6s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-13235.625, test=-35708.576), r2=(train=0.973, test=0.770), total= 12.6s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-13564.490, test=-47797.070), r2=(train=0.970, test=0.651), total= 13.7s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-13472.008, test=-41550.677), r2=(train=0.969, test=0.772), total= 14.1s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-15729.373, test=-39939.546), r2=(train=0.962, test=0.693), total= 12.8s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__splitter=best, neg_root_mean_squared_error=(train=-11336.987, test=-53198.489), r2=(train=0.979, test=0.562), total= 11.6s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_root_mean_squared_error=(train=-18260.636, test=-35074.244), r2=(train=0.948, test=0.778), total= 13.2s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js x_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_roo

```

```

t_mean_squared_error=(train=-25846.097, test=-43945.715), r2=(train=0.892, test=
0.705), total= 13.8s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-18261.467, test=-40078.888), r2=(train=0.944, test=
0.788), total= 14.7s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-21494.413, test=-29255.943), r2=(train=0.929, test=
0.835), total= 12.1s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=1, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-18957.222, test=-45161.272), r2=(train=0.942, test=
0.684), total= 11.6s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-20818.099, test=-35243.752), r2=(train=0.933, test=
0.776), total= 13.1s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-22896.704, test=-40233.315), r2=(train=0.916, test=
0.753), total= 13.7s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-21673.612, test=-42089.457), r2=(train=0.921, test=
0.766), total= 14.2s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-25722.951, test=-33871.515), r2=(train=0.899, test=
0.779), total= 12.4s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=2, model__splitter=best, neg_roo
t_mean_squared_error=(train=-22390.049, test=-43851.070), r2=(train=0.920, test=
0.702), total= 11.8s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-24253.818, test=-36529.481), r2=(train=0.909, test=
0.759), total= 12.6s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-27661.317, test=-44063.981), r2=(train=0.877, test=
0.704), total= 13.9s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_

```

```

__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-23805.672, test=-42068.093), r2=(train=0.905, test=
0.767), total= 14.1s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-27556.247, test=-31039.179), r2=(train=0.884, test=
0.814), total= 12.1s
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best
[CV] model__criterion=mae, model__max_depth=10, model__max_features=auto, model_
__min_samples_leaf=3, model__min_samples_split=12, model__splitter=best, neg_roo
t_mean_squared_error=(train=-24761.479, test=-43766.967), r2=(train=0.902, test=
0.703), total= 11.5s
[Parallel(n_jobs=1)]: Done 80 out of 80 | elapsed: 17.4min finished

```

```
In [63]: dt_results.best_params_
```

```
Out[63]: {'model__criterion': 'mse',
          'model__max_depth': 10,
          'model__max_features': 'auto',
          'model__min_samples_leaf': 3,
          'model__min_samples_split': 12,
          'model__splitter': 'best'}
```

```
In [64]: get_results(dt_results)
```

The best model parameters produce a mean rmse score on train data of:
-23483.469430541732

The best model parameters produce a mean R-squared score on train data of:
0.9118110885672543

The best model parameters produce a mean rmse score on test data of:
-38651.694073440085

The best model parameters produce a mean R-squared score on test data of:
0.760553933795492

Analysis:

The best parameters for the decision tree regressor were the absolute error for split criterion, a max model depth set to 10 (could be interesting to play with this value in the future), a min samples leaf value of 3, a min samples split value of 12, and the 'best' splitter method.

The best model in the grid is considerably overfit to the training data both relative to the validation data as well as to the train set.

Random Forest Regression

```
In [65]: x_train = train_df.drop(['SalePrice'], axis=1)
         y_train = train_df['SalePrice']
```

```
In [66]: rf_pipeline = Pipeline(steps=[('trans', transformer),
                                       ('rfe', RFE(LinearRegression(normalize=False), n_f
                                       ('model', RandomForestRegressor(random_state=100))
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
rf_grid = {'model__n_estimators': [50, 250],
```

```

'model__criterion': ['mse', 'mae'],
'model__max_depth': [None, 10],
'model__min_samples_split': [2],
'model__min_samples_leaf': [1, 3],
'model__max_features': ['auto'],
'model__ccp_alpha': [0.0]}

```

```

rf_gs = GridSearchCV(rf_pipeline,
                    rf_grid,
                    scoring=['neg_root_mean_squared_error', 'r2'],
                    refit='neg_root_mean_squared_error',
                    cv=5,
                    verbose=3,
                    return_train_score=True)

```

In [67]: `rf_results = rf_gs.fit(x_train, y_train)`

```

Fitting 5 folds for each of 16 candidates, totalling 80 fits
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-13053.776, test=-28794.846), r2=(train=0.974, test=0.850), total= 13.3s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 13.3s remaining: 0.0s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-12898.673, test=-34218.859), r2=(train=0.973, test=0.821), total= 15.2s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 28.6s remaining: 0.0s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-12090.817, test=-32258.990), r2=(train=0.975, test=0.863), total= 14.8s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-13196.930, test=-28953.461), r2=(train=0.973, test=0.839), total= 12.7s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-12035.625, test=-34972.127), r2=(train=0.977, test=0.811), total= 12.3s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-11933.347, test=-28135.09

```



```

ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250, neg_root_mean_squared_error=(train=-19029.074, test=-28505.68
4), r2=(train=0.944, test=0.853), total= 15.1s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250, neg_root_mean_squared_error=(train=-20568.004, test=-31943.14
0), r2=(train=0.932, test=0.844), total= 16.1s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250, neg_root_mean_squared_error=(train=-20158.626, test=-34650.24
0), r2=(train=0.932, test=0.842), total= 16.8s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250, neg_root_mean_squared_error=(train=-21788.077, test=-27464.63
4), r2=(train=0.927, test=0.855), total= 14.2s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
n_estimators=250, neg_root_mean_squared_error=(train=-18973.704, test=-37809.84
2), r2=(train=0.942, test=0.779), total= 14.1s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-14494.685, test=-28821.967),
r2=(train=0.968, test=0.850), total= 14.1s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-15066.044, test=-33871.321),
r2=(train=0.963, test=0.825), total= 14.7s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-14781.597, test=-32132.680),
r2=(train=0.963, test=0.864), total= 14.8s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-16649.886, test=-27858.450),
r2=(train=0.958, test=0.851), total= 12.8s
[CV] model__ccp_alpha=0.0, model__criterion=mse, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50

```

```

[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=50, neg_root_mean_squared_error=(train=-13750.722, test=-35050.225), r2=(train=0.970, test=0.810), total= 12.2s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250, neg_root_mean_squared_error=(train=-13903.088, test=-28282.794), r2=(train=0.970, test=0.856), total= 15.4s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250, neg_root_mean_squared_error=(train=-14469.399, test=-33573.905), r2=(train=0.966, test=0.828), total= 16.3s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250, neg_root_mean_squared_error=(train=-14495.528, test=-32509.986), r2=(train=0.965, test=0.861), total= 17.5s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250, neg_root_mean_squared_error=(train=-16374.884, test=-27794.861), r2=(train=0.959, test=0.851), total= 14.4s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model_n_estimators=250, neg_root_mean_squared_error=(train=-13473.241, test=-36334.982), r2=(train=0.971, test=0.796), total= 14.1s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50, neg_root_mean_squared_error=(train=-20013.641, test=-28796.974), r2=(train=0.938, test=0.850), total= 13.0s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50, neg_root_mean_squared_error=(train=-21481.747, test=-32548.265), r2=(train=0.926, test=0.838), total= 14.7s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50, neg_root_mean_squared_error=(train=-20879.407, test=-34765.291), r2=(train=0.927, test=0.841), total= 14.7s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n_estimators=50

```



```

_estimators=50, neg_root_mean_squared_error=(train=-22327.635, test=-28514.286),
r2=(train=0.924, test=0.843), total= 12.6s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=50, neg_root_mean_squared_error=(train=-19589.528, test=-37879.484),
r2=(train=0.938, test=0.778), total= 13.7s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250, neg_root_mean_squared_error=(train=-19520.491, test=-28573.18
4), r2=(train=0.941, test=0.853), total= 18.9s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250, neg_root_mean_squared_error=(train=-21080.181, test=-31978.23
2), r2=(train=0.928, test=0.844), total= 17.2s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250, neg_root_mean_squared_error=(train=-20694.568, test=-34644.05
1), r2=(train=0.928, test=0.842), total= 16.9s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250, neg_root_mean_squared_error=(train=-22418.412, test=-27515.14
9), r2=(train=0.923, test=0.854), total= 13.9s
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_max
_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250
[CV] model_ccp_alpha=0.0, model_criterion=mse, model_max_depth=10, model_ma
x_features=auto, model_min_samples_leaf=3, model_min_samples_split=2, model_n
_estimators=250, neg_root_mean_squared_error=(train=-19503.741, test=-37885.66
4), r2=(train=0.939, test=0.778), total= 13.6s
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model_m
ax_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model__
max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
n_estimators=50, neg_root_mean_squared_error=(train=-12581.325, test=-27555.37
2), r2=(train=0.976, test=0.863), total= 17.9s
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model_m
ax_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model__
max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
n_estimators=50, neg_root_mean_squared_error=(train=-13225.974, test=-34658.94
5), r2=(train=0.972, test=0.817), total= 19.0s
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model_m
ax_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
n_estimators=50
[CV] model_ccp_alpha=0.0, model_criterion=mae, model_max_depth=None, model__
max_features=auto, model_min_samples_leaf=1, model_min_samples_split=2, model__
neg_root_mean_squared_error=(train=-11855.276, test=-32928.24
8), r2=(train=0.976, test=0.857), total= 19.2s

```

```

[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-13252.168, test=-27311.858), r2=(train=0.973, test=0.856), total= 16.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-12188.972, test=-35133.049), r2=(train=0.976, test=0.809), total= 17.4s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-11834.073, test=-27496.952), r2=(train=0.978, test=0.863), total= 41.5s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-12097.399, test=-34659.498), r2=(train=0.976, test=0.817), total= 38.1s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-11663.391, test=-32850.088), r2=(train=0.977, test=0.858), total= 40.5s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-12938.387, test=-27254.850), r2=(train=0.974, test=0.857), total= 36.4s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-11444.003, test=-35465.480), r2=(train=0.979, test=0.805), total= 39.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-21428.185, test=-28129.940), r2=(train=0.929, test=0.857), total= 16.1s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-23047.210, test=-33425.379), r2=(train=0.914, test=0.829), total= 16.9s

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50

```

```

n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=50, neg_root_mean_squared_error=(train=-22246.370, test=-35421.25
8), r2=(train=0.917, test=0.835), total= 17.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=50, neg_root_mean_squared_error=(train=-23312.237, test=-27807.58
2), r2=(train=0.917, test=0.851), total= 15.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=50, neg_root_mean_squared_error=(train=-21491.761, test=-39003.89
0), r2=(train=0.926, test=0.764), total= 15.5s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250, neg_root_mean_squared_error=(train=-21170.936, test=-28208.87
5), r2=(train=0.931, test=0.856), total= 31.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250, neg_root_mean_squared_error=(train=-22554.080, test=-33328.07
8), r2=(train=0.918, test=0.830), total= 31.3s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250, neg_root_mean_squared_error=(train=-22206.331, test=-35031.05
7), r2=(train=0.917, test=0.838), total= 36.6s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250, neg_root_mean_squared_error=(train=-23643.043, test=-27481.48
1), r2=(train=0.914, test=0.855), total= 30.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__m
ax_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=None, model__
max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__
_n_estimators=250, neg_root_mean_squared_error=(train=-21331.989, test=-38797.07
4), r2=(train=0.927, test=0.767), total= 31.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-15185.236, test=-27337.380),
r2=(train=0.964, test=0.865), total= 17.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma

```

```

x_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-15849.084, test=-36270.354), r2=(train=0.960, test=0.799), total= 18.2s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-15308.987, test=-33488.724), r2=(train=0.961, test=0.852), total= 19.2s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-17272.544, test=-27039.323), r2=(train=0.954, test=0.859), total= 16.2s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-14615.569, test=-35021.067), r2=(train=0.966, test=0.810), total= 17.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-14551.582, test=-27639.036), r2=(train=0.967, test=0.862), total= 37.6s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-15127.339, test=-35652.357), r2=(train=0.963, test=0.806), total= 36.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-15101.177, test=-32914.745), r2=(train=0.962, test=0.857), total= 38.2s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-17073.712, test=-27274.566), r2=(train=0.955, test=0.857), total= 34.1s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=1, model__min_samples_split=2, model__n_estimators=250, neg_root_mean_squared_error=(train=-14129.350, test=-34652.670), r2=(train=0.968, test=0.814), total= 37.0s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n_estimators=50, neg_root_mean_squared_error=(train=-21837.966, test=-27921.260),

```

```

r2=(train=0.926, test=0.859), total= 16.2s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-23680.137, test=-33531.146),
r2=(train=0.910, test=0.828), total= 17.1s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-22867.524, test=-36052.744),
r2=(train=0.912, test=0.829), total= 17.7s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-23842.483, test=-27552.987),
r2=(train=0.913, test=0.854), total= 15.3s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=50, neg_root_mean_squared_error=(train=-21829.599, test=-38593.023),
r2=(train=0.924, test=0.769), total= 15.4s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250, neg_root_mean_squared_error=(train=-21444.929, test=-28187.94
3), r2=(train=0.929, test=0.857), total= 30.5s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250, neg_root_mean_squared_error=(train=-22959.647, test=-33472.41
5), r2=(train=0.915, test=0.829), total= 29.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250, neg_root_mean_squared_error=(train=-22711.235, test=-35283.68
2), r2=(train=0.913, test=0.836), total= 31.4s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250, neg_root_mean_squared_error=(train=-24296.121, test=-27403.76
5), r2=(train=0.910, test=0.855), total= 26.8s
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__max
_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250
[CV] model__ccp_alpha=0.0, model__criterion=mae, model__max_depth=10, model__ma
x_features=auto, model__min_samples_leaf=3, model__min_samples_split=2, model__n
_estimators=250, neg_root_mean_squared_error=(train=-21681.142, test=-38695.62

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js al= 30.9s

[Parallel(n_jobs=1)]: Done 80 out of 80 | elapsed: 27.1min finished

It seems that as the model iterates through more candidates, it is beginning to overfit less.

However, it appears that there is at least one fold per candidate that returns an r-squared value below .08. This could potentially be caused by model overfitting on the outliers.

The 'poisson' criteria seems to be resulting in significant overfitting as the train r-squared scores are ~.96 while the validation r-squared scores are between 0.8 and 0.6 for the most part.

```
In [70]: rf_results.best_params_
```

```
Out[70]: {'model__ccp_alpha': 0.0,
          'model__criterion': 'mse',
          'model__max_depth': None,
          'model__max_features': 'auto',
          'model__min_samples_leaf': 1,
          'model__min_samples_split': 2,
          'model__n_estimators': 250}
```

```
In [71]: get_results(rf_results)
```

```
The best model parameters produce a mean rmse score on train data of:
-12115.373235799616
```

```
The best model parameters produce a mean R-squared score on train data of:
0.9765943742612325
```

```
The best model parameters produce a mean rmse score on test data of:
-31515.990302817292
```

```
The best model parameters produce a mean R-squared score on test data of:
0.8402221513439431
```

Analysis:

The best parameters for the random forest regressor were the absolute error for split criterion, a max model depth of none, a min samples leaf value of 1, a min samples split value of 2, and n_estimators 150.

The best model in the grid is significantly overfit to the training data both relative to the validation data, and more broadly.

XGBoost Regressor

```
In [149... x_train = train_df.drop(['SalePrice'], axis=1)
            y_train = train_df['SalePrice']
```

```
In [150... xgb_pipeline = Pipeline(steps=[('trans', transformer),
                                   ('rfe', RFE(LinearRegression(normalize=False), n_
                                   ('model', XGBRegressor(random_state=42))])

xgb_grid = {'model__n_estimators': [250, ],
            'model__max_depth': [None, 10],
            'model__booster': ['gbtree', 'gblinear', 'dart'],
            'model__reg_alpha': [.01, 1.0],
            'model__reg_lambda': [5.0, 100.0]}
```

```
xgb_gs = GridSearchCV(xgb_pipeline,
                      xgb_grid,
                      scoring=['neg_root_mean_squared_error', 'r2'],
                      refit='neg_root_mean_squared_error',
                      cv=5,
                      verbose=3,
                      return_train_score=True)
```

In [151... `xgb_results = xgb_gs.fit(x_train, y_train)`

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-8806.488, test=-27474.891), r2=(train=0.988, test=0.864), total= 13.0s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 13.0s remaining: 0.0s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10309.400, test=-36896.248), r2=(train=0.983, test=0.792), total= 13.8s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 26.8s remaining: 0.0s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10069.150, test=-32738.183), r2=(train=0.983, test=0.859), total= 14.8s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-12213.353, test=-31401.668), r2=(train=0.977, test=0.810), total= 12.2s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-9046.607, test=-33581.869), r2=(train=0.987, test=0.825), total= 11.8s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-9604.448, test=-27768.220), r2=(train=0.986, test=0.861), total= 13.1s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11147.286, test=-37431.024), r2=(train=0.980, test=0.786), total= 13.8s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11250.766, test=-34035.292), r2=(train=0.979, test=0.847), total= 14.3s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-13326.933, test=-32262.225), r2=(train=0.973, test=0.800), total= 12.3s
[CV] model__booster=gbtrees, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js |ax_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-
```

```

-10188.210, test=-32835.886), r2=(train=0.983, test=0.833), total= 11.8s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-8806.520, test=-27474.893), r2=(train=0.988, test=0.864), total= 12.7s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10309.434, test=-36896.251), r2=(train=0.983, test=0.792), total= 13.9s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10049.473, test=-32755.754), r2=(train=0.983, test=0.859), total= 14.2s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-12163.796, test=-31403.873), r2=(train=0.977, test=0.810), total= 12.2s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-9046.643, test=-33581.872), r2=(train=0.987, test=0.825), total= 12.3s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-9667.991, test=-27676.161), r2=(train=0.986, test=0.862), total= 12.7s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11264.316, test=-37340.020), r2=(train=0.980, test=0.787), total= 13.9s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11250.799, test=-34035.294), r2=(train=0.979, test=0.847), total= 14.2s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-13326.964, test=-32262.237), r2=(train=0.973, test=0.800), total= 12.1s
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-10188.240, test=-32835.885), r2=(train=0.983, test=0.833), total= 11.8s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-4318.153, test=-28053.307), r2=(train=0.997, test=0.858), total= 14.8s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-4765.918, test=-37704.731), r2=(train=0.996, test=0.783), total= 14.3s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js jax_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-53

```



```

55.634, test=-34379.027), r2=(train=0.995, test=0.844), total= 14.5s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-71
45.747, test=-30769.288), r2=(train=0.992, test=0.818), total= 13.5s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-45
42.508, test=-32082.633), r2=(train=0.997, test=0.841), total= 12.1s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-5
234.111, test=-27447.337), r2=(train=0.996, test=0.864), total= 12.9s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-6
013.058, test=-38313.744), r2=(train=0.994, test=0.776), total= 14.1s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-6
340.638, test=-34658.426), r2=(train=0.993, test=0.842), total= 14.3s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-8
481.706, test=-31231.379), r2=(train=0.989, test=0.812), total= 12.4s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-5
797.885, test=-32289.007), r2=(train=0.995, test=0.839), total= 12.0s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-428
1.361, test=-28064.510), r2=(train=0.997, test=0.858), total= 12.9s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-455
9.824, test=-37990.685), r2=(train=0.997, test=0.780), total= 14.2s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-535
4.173, test=-34380.065), r2=(train=0.995, test=0.844), total= 14.5s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-714
5.813, test=-30769.292), r2=(train=0.992, test=0.818), total= 12.4s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-455
6.403, test=-32100.514), r2=(train=0.997, test=0.840), total= 11.9s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js |ax_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-52

```

```

72.246, test=-27450.336), r2=(train=0.996, test=0.864), total= 13.1s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-60
51.036, test=-38317.130), r2=(train=0.994, test=0.776), total= 13.9s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-63
40.688, test=-34658.416), r2=(train=0.993, test=0.842), total= 14.5s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-84
81.761, test=-31231.382), r2=(train=0.989, test=0.812), total= 12.8s
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gbtree, model__max_depth=10, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-57
99.663, test=-32294.582), r2=(train=0.995, test=0.839), total= 12.0s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train
=-77886.814, test=-71955.583), r2=(train=0.062, test=0.065), total= 12.6s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train
=-77196.715, test=-79267.032), r2=(train=0.040, test=0.041), total= 13.9s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train
=-75352.127, test=-85392.357), r2=(train=0.046, test=0.039), total= 14.1s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train
=-79444.148, test=-71301.893), r2=(train=0.034, test=0.021), total= 12.0s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train
=-76833.239, test=-78274.477), r2=(train=0.053, test=0.052), total= 11.6s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(trai
n=-78493.561, test=-72554.669), r2=(train=0.047, test=0.050), total= 12.9s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(trai
n=-77588.811, test=-79686.156), r2=(train=0.031, test=0.031), total= 13.8s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(trai
n=-75785.023, test=-85836.512), r2=(train=0.035, test=0.029), total= 14.0s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, mo
    del__reg_alpha=0.01, model__reg_lambda=100.0
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
    __max_depth=None, model__n_estimators=250, m
    odel__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(trai

```

```

n=-79785.446, test=-71624.083), r2=(train=0.026, test=0.012), total= 12.6s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77349.343, test=-78782.240), r2=(train=0.040, test=0.039), total= 12.2s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77886.879, test=-71955.647), r2=(train=0.062, test=0.065), total= 12.8s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77196.757, test=-79267.077), r2=(train=0.040, test=0.041), total= 13.9s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-75352.175, test=-85392.406), r2=(train=0.046, test=0.039), total= 14.1s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-79444.182, test=-71301.924), r2=(train=0.034, test=0.021), total= 12.1s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-76833.294, test=-78274.529), r2=(train=0.053, test=0.052), total= 12.2s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-78493.610, test=-72554.714), r2=(train=0.047, test=0.050), total= 12.4s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77588.850, test=-79686.196), r2=(train=0.031, test=0.031), total= 13.7s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-75785.061, test=-85836.549), r2=(train=0.035, test=0.029), total= 14.0s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-79785.470, test=-71624.104), r2=(train=0.026, test=0.012), total= 12.0s
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=gblinear, model__max_depth=None, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77349.384, test=-78782.276), r2=(train=0.040, test=0.039), total= 11.8s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[01:44:38] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.

```

This may not be accurate due to some parameters are only used in language bind

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

passed down to XGBoost core. Or some parameters are not used but slip through

this

verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77886.814, test=-71955.583), r2=(train=0.062, test=0.065), total= 12.8s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[01:44:51] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77196.715, test=-79267.032), r2=(train=0.040, test=0.041), total= 13.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[01:45:05] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-75352.127, test=-85392.357), r2=(train=0.046, test=0.039), total= 14.0s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[01:45:18] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-79444.148, test=-71301.893), r2=(train=0.034, test=0.021), total= 12.2s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[01:45:29] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js some parameters are not used but slip through this

verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-76833.239, test=-78274.477), r2=(train=0.053, test=0.052), total= 11.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[01:45:42] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-78493.561, test=-72554.669), r2=(train=0.047, test=0.050), total= 12.8s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[01:45:56] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77588.811, test=-79686.156), r2=(train=0.031, test=0.031), total= 13.9s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[01:46:10] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-75785.023, test=-85836.512), r2=(train=0.035, test=0.029), total= 14.1s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[01:46:22] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-79785.446, test=-71624.083), r2=(train=0.026, test=0.012), total= 12.1s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0
[01:46:34] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77349.343, test=-78782.240), r2=(train=0.040, test=0.039), total= 11.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[01:46:47] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77886.879, test=-71955.647), r2=(train=0.062, test=0.065), total= 12.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[01:47:01] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-77196.757, test=-79267.077), r2=(train=0.040, test=0.041), total= 14.7s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[01:47:17] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-75352.175, test=-85392.406), r2=(train=0.046, test=0.039), total= 16.1s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[01:47:30] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-79444.182, test=-71301.924), r2=(train=0.034, test=0.021), total= 12.0s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0
[01:47:41] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-76833.294, test=-78274.529), r2=(train=0.053, test=0.052), total= 11.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[01:47:54] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-78493.610, test=-72554.714), r2=(train=0.047, test=0.050), total= 12.7s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[01:48:07] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77588.850, test=-79686.196), r2=(train=0.031, test=0.031), total= 13.5s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[01:48:22] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-75785.061, test=-85836.549), r2=(train=0.035, test=0.029), total= 14.2s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[01:48:34] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-79785.470, test=-71624.104), r2=(train=0.026, test=0.012), total= 12.6s
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0
[01:48:46] WARNING: /Users/runner/miniforge3/conda-bld/xgboost_1598185652448/work/src/learner.cc:516:
Parameters: { max_depth } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[CV] model__booster=gblinear, model__max_depth=10, model__n_estimators=250, model__reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-77349.384, test=-78782.276), r2=(train=0.040, test=0.039), total= 12.1s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-8806.488, test=-27474.891), r2=(train=0.988, test=0.864), total= 13.4s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10309.400, test=-36896.248), r2=(train=0.983, test=0.792), total= 14.5s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0
```

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js }_depth=None, model__n_estimators=250, model__reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-10
```



```

069.150, test=-32738.183), r2=(train=0.983, test=0.859), total= 15.0s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-12
    213.353, test=-31401.668), r2=(train=0.977, test=0.810), total= 13.1s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-90
    46.607, test=-33581.870), r2=(train=0.987, test=0.825), total= 12.4s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-9
    604.448, test=-27768.220), r2=(train=0.986, test=0.861), total= 13.1s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-1
    1147.286, test=-37431.024), r2=(train=0.980, test=0.786), total= 14.5s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-1
    1250.766, test=-34035.292), r2=(train=0.979, test=0.847), total= 17.8s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-1
    3326.933, test=-32262.225), r2=(train=0.973, test=0.800), total= 13.1s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-1
    0188.210, test=-32835.886), r2=(train=0.983, test=0.833), total= 12.6s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-877
    8.349, test=-27510.387), r2=(train=0.988, test=0.863), total= 13.3s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-103
    09.434, test=-36896.251), r2=(train=0.983, test=0.792), total= 14.5s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-100
    49.473, test=-32755.754), r2=(train=0.983, test=0.859), total= 14.9s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-121
    63.796, test=-31403.873), r2=(train=0.977, test=0.810), total= 13.4s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-904
    6.643, test=-33581.872), r2=(train=0.987, test=0.825), total= 12.5s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js }__depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-96

```

```

67.991, test=-27676.161), r2=(train=0.986, test=0.862), total= 13.5s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11
264.316, test=-37340.020), r2=(train=0.980, test=0.787), total= 14.7s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-11
250.799, test=-34035.294), r2=(train=0.979, test=0.847), total= 16.6s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-13
326.964, test=-32262.238), r2=(train=0.973, test=0.800), total= 13.0s
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=None, model__n_estimators=250, model__
    __reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-10
188.240, test=-32835.885), r2=(train=0.983, test=0.833), total= 12.4s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-431
8.153, test=-28053.307), r2=(train=0.997, test=0.858), total= 14.2s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-476
5.918, test=-37704.731), r2=(train=0.996, test=0.783), total= 15.1s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-535
5.634, test=-34379.027), r2=(train=0.995, test=0.844), total= 16.2s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-714
5.747, test=-30769.288), r2=(train=0.992, test=0.818), total= 14.0s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-454
2.508, test=-32082.633), r2=(train=0.997, test=0.841), total= 13.3s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-523
4.111, test=-27447.337), r2=(train=0.996, test=0.864), total= 14.0s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-601
3.058, test=-38313.744), r2=(train=0.994, test=0.776), total= 15.4s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-634
0.638, test=-34658.426), r2=(train=0.993, test=0.842), total= 15.6s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-848

```

```

1.706, test=-31231.379), r2=(train=0.989, test=0.812), total= 13.7s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=0.01, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=0.01, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-579
7.885, test=-32289.007), r2=(train=0.995, test=0.839), total= 13.5s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-4281.
361, test=-28064.510), r2=(train=0.997, test=0.858), total= 14.3s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-4559.
824, test=-37990.685), r2=(train=0.997, test=0.780), total= 15.2s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-5354.
173, test=-34380.065), r2=(train=0.995, test=0.844), total= 15.8s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-7145.
813, test=-30769.292), r2=(train=0.992, test=0.818), total= 13.6s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=75.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=75.0, neg_root_mean_squared_error=(train=-4556.
403, test=-32100.514), r2=(train=0.997, test=0.840), total= 14.2s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-527
2.245, test=-27450.335), r2=(train=0.996, test=0.864), total= 14.3s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-605
1.036, test=-38317.130), r2=(train=0.994, test=0.776), total= 15.2s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-634
0.688, test=-34658.416), r2=(train=0.993, test=0.842), total= 15.7s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-848
1.761, test=-31231.382), r2=(train=0.989, test=0.812), total= 13.8s
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__r
eg_alpha=1.0, model__reg_lambda=100.0
[CV] model__booster=dart, model__max_depth=10, model__n_estimators=250, model__
reg_alpha=1.0, model__reg_lambda=100.0, neg_root_mean_squared_error=(train=-579
9.663, test=-32294.582), r2=(train=0.995, test=0.839), total= 13.3s
[Parallel(n_jobs=1)]: Done 120 out of 120 | elapsed: 27.0min finished

```

In [152... xgb_results.best_params_

Out[152... {'model__booster': 'gbtree',
 'model__max_depth': None,

```
'model__reg_alpha': 0.01,  
'model__reg_lambda': 75.0}
```

```
In [153... get_results(xgb_results)
```

```
The best model parameters produce a mean rmse score on train data of:  
-10088.99973543373
```

```
The best model parameters produce a mean R-squared score on train data of:  
0.9835781039433739
```

```
The best model parameters produce a mean rmse score on test data of:  
-32418.571807187243
```

```
The best model parameters produce a mean R-squared score on test data of:  
0.830023355678742
```

Analysis:

The best parameters for the XGboost regressor were 'dart' for the booster parameter, a max model depth of none, 250 n_estimators, a reg_alpha of 1.0, and a reg_lambda of 50.0

The best model in the grid is more overfit to the training data relative to the validation data and the whole set than any of the other regressions I have fit thus far.

Generally, it does not appear that recursive feature elimination was that helpful in reducing overfitting of the models.

Of the four models I trained using GridSearchCV, the Ridge Regression appeared to have the best fit in regard to the bias-variance trade-off, so I am going to train another using the best features from the grid search and messing around with the n_features_to_select parameter of the recursive feature elimination object.

Experimenting With Neural Networks

```
In [437... x_train = train_df.drop(['SalePrice'], axis=1)  
y_train = train_df['SalePrice']
```

```
In [438... categorical_cols = x_train.select_dtypes('object')  
numerical_cols = x_train.select_dtypes(exclude=['object'])  
  
numerical_transformer = Pipeline(steps=[('ss', StandardScaler())])  
ordinal_transformer = Pipeline(steps=[('oe', OrdinalEncoder())])  
  
oe_transformer = ColumnTransformer(transformers=[('categorical', ordinal_transfo  
num_transformer = ColumnTransformer(transformers=[('numerical', numerical_transf
```

```
In [454... x_train = train_df.drop(['SalePrice'], axis=1)  
y_train = train_df['SalePrice']  
  
# Transforming categorical variables  
x_train = oe_transformer.fit_transform(x_train)
```

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
```

```
x_tr, x_val, y_tr, y_val = train_test_split(x_train, y_train, test_size=0.25, ra
```

```

# Selecting features using recursive feature elimination
rfe = RFE(LinearRegression(), n_features_to_select=40)
x_tr = rfe.fit_transform(x_tr, y_tr)
x_val = rfe.transform(x_val)

# Building the model
model = models.Sequential()

model.add(layers.Dense(40, activation='tanh', input_shape=(40,)))

model.add(layers.Dense(80, activation='tanh', kernel_regularizer=l2(l2=0.001)))

model.add(Dropout(0.5))

model.add(layers.Dense(160, activation='tanh', kernel_regularizer=l2(l2=0.001)))

model.add(Dropout(0.5))

model.add(layers.Dense(40, activation='tanh', kernel_regularizer=l2(l2=0.001)))

model.add(Dropout(0.25))

model.add(layers.Dense(10, activation='tanh', kernel_regularizer=l2(l2=0.001)))

model.add(Dropout(0.25))

model.add(layers.Dense(5, activation='tanh', kernel_regularizer=l2(l2=0.01)))

model.add(Dropout(0.2))

model.add(layers.Dense(1, activation='linear'))

# Compiling the model
model.compile(optimizer='SGD',
              loss='mse',
              metrics=['mse'])

# Fitting The Model
history = model.fit(x_tr,
                    y_tr,
                    batch_size=50,
                    epochs=100,
                    steps_per_epoch=20,
                    verbose=2,
                    validation_data=(x_val, y_val),
                    validation_steps=7,
                    validation_batch_size=50
                    )

```

```

Epoch 1/100
20/20 - 0s - loss: 26188328960.0000 - mse: 26187814912.0000 - val_loss: 10096229
376.0000 - val_mse: 10095594496.0000
Epoch 2/100
20/20 - 0s - loss: 15509465088.0000 - mse: 15508811776.0000 - val_loss: 81803683
84.0000 - val_mse: 8179692544.0000
Epoch 3/100
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js · mse: 11890400256.0000 - val_loss: 68646369
28.0000 - val_mse: 6863880192.0000

```

Epoch 4/100
20/20 - 0s - loss: 10040206336.0000 - mse: 10039446528.0000 - val_loss: 64428702
72.0000 - val_mse: 6442105856.0000

Epoch 5/100
20/20 - 0s - loss: 7692273152.0000 - mse: 7691509760.0000 - val_loss: 618827776
0.0000 - val_mse: 6187516416.0000

Epoch 6/100
20/20 - 0s - loss: 7615449600.0000 - mse: 7050882048.0000 - val_loss: 751194368
0.0000 - val_mse: 5968963072.0000

Epoch 7/100
20/20 - 0s - loss: 8353430016.0000 - mse: 6816254464.0000 - val_loss: 752938240
0.0000 - val_mse: 5998694912.0000

Epoch 8/100
20/20 - 0s - loss: 8565298176.0000 - mse: 7040318976.0000 - val_loss: 748888780
8.0000 - val_mse: 5970394624.0000

Epoch 9/100
20/20 - 0s - loss: 7885812736.0000 - mse: 6372931584.0000 - val_loss: 746390681
6.0000 - val_mse: 5957510144.0000

Epoch 10/100
20/20 - 0s - loss: 8180767232.0000 - mse: 6678356480.0000 - val_loss: 746505830
4.0000 - val_mse: 5965921280.0000

Epoch 11/100
20/20 - 0s - loss: 8167787008.0000 - mse: 6634845184.0000 - val_loss: 748715673
6.0000 - val_mse: 5947059200.0000

Epoch 12/100
20/20 - 0s - loss: 8124568064.0000 - mse: 6590308864.0000 - val_loss: 747505612
8.0000 - val_mse: 5947229696.0000

Epoch 13/100
20/20 - 0s - loss: 8184435200.0000 - mse: 6662620160.0000 - val_loss: 750015232
0.0000 - val_mse: 5984497152.0000

Epoch 14/100
20/20 - 0s - loss: 7859954176.0000 - mse: 6350211072.0000 - val_loss: 745282304
0.0000 - val_mse: 5949243392.0000

Epoch 15/100
20/20 - 0s - loss: 7453519872.0000 - mse: 5955752448.0000 - val_loss: 743942656
0.0000 - val_mse: 5947825152.0000

Epoch 16/100
20/20 - 0s - loss: 8274885120.0000 - mse: 6789000192.0000 - val_loss: 742720153
6.0000 - val_mse: 5947483648.0000

Epoch 17/100
20/20 - 0s - loss: 7911601664.0000 - mse: 6437503488.0000 - val_loss: 741499596
8.0000 - val_mse: 5947066880.0000

Epoch 18/100
20/20 - 0s - loss: 7211947520.0000 - mse: 5749543424.0000 - val_loss: 741179033
6.0000 - val_mse: 5955556352.0000

Epoch 19/100
20/20 - 0s - loss: 8643514368.0000 - mse: 7192710656.0000 - val_loss: 739222272
0.0000 - val_mse: 5947590144.0000

Epoch 20/100
20/20 - 0s - loss: 7724259840.0000 - mse: 6284965888.0000 - val_loss: 738035353
6.0000 - val_mse: 5947228160.0000

Epoch 21/100
20/20 - 0s - loss: 7828993024.0000 - mse: 6401115136.0000 - val_loss: 736892979
2.0000 - val_mse: 5947222016.0000

Epoch 22/100
20/20 - 0s - loss: 7775602176.0000 - mse: 6359054336.0000 - val_loss: 741342105
6.0000 - val_mse: 6003039744.0000

Epoch 23/100
20/20 - 0s - loss: 7707425280.0000 - mse: 6302205440.0000 - val_loss: 736582860
8.0000 - val_mse: 5966074368.0000

Epoch 24/100
20/20 - 0s - loss: 8188403712.0000 - mse: 6794154496.0000 - val_loss: 733552691
2.0000 - val_mse: 5946923520.0000

20/20 - 0s - loss: 6961206784.0000 - mse: 5578019328.0000 - val_loss: 733373593

```

6.0000 - val_mse: 5956194304.0000
Epoch 26/100
20/20 - 0s - loss: 7450252288.0000 - mse: 6078037504.0000 - val_loss: 733619712
0.0000 - val_mse: 5969630208.0000
Epoch 27/100
20/20 - 0s - loss: 8455199232.0000 - mse: 7093867520.0000 - val_loss: 730856550
4.0000 - val_mse: 5952885248.0000
Epoch 28/100
20/20 - 0s - loss: 7584183808.0000 - mse: 6233652736.0000 - val_loss: 729377638
4.0000 - val_mse: 5948896768.0000
Epoch 29/100
20/20 - 0s - loss: 7916024320.0000 - mse: 6576206336.0000 - val_loss: 728514048
0.0000 - val_mse: 5950974976.0000
Epoch 30/100
20/20 - 0s - loss: 7779212800.0000 - mse: 6450022912.0000 - val_loss: 727197132
8.0000 - val_mse: 5948434432.0000
Epoch 31/100
20/20 - 0s - loss: 7803129344.0000 - mse: 6484484608.0000 - val_loss: 726683392
0.0000 - val_mse: 5953841152.0000
Epoch 32/100
20/20 - 0s - loss: 7830039552.0000 - mse: 6521862656.0000 - val_loss: 725115494
4.0000 - val_mse: 5948631040.0000
Epoch 33/100
20/20 - 0s - loss: 7745680896.0000 - mse: 6447884800.0000 - val_loss: 724364083
2.0000 - val_mse: 5951495168.0000
Epoch 34/100
20/20 - 0s - loss: 7574787584.0000 - mse: 6231597056.0000 - val_loss: 734066739
2.0000 - val_mse: 5947177984.0000
Epoch 35/100
20/20 - 0s - loss: 8225057792.0000 - mse: 6837051392.0000 - val_loss: 733694412
8.0000 - val_mse: 5954556416.0000
Epoch 36/100
20/20 - 0s - loss: 6910556672.0000 - mse: 5533560320.0000 - val_loss: 732272128
0.0000 - val_mse: 5951347200.0000
Epoch 37/100
20/20 - 0s - loss: 8766458880.0000 - mse: 7400384512.0000 - val_loss: 731705446
4.0000 - val_mse: 5956604416.0000
Epoch 38/100
20/20 - 0s - loss: 7433861120.0000 - mse: 6078626304.0000 - val_loss: 729651916
8.0000 - val_mse: 5946907136.0000
Epoch 39/100
20/20 - 0s - loss: 8123907072.0000 - mse: 6779421184.0000 - val_loss: 728578304
0.0000 - val_mse: 5946924032.0000
Epoch 40/100
20/20 - 0s - loss: 7369131008.0000 - mse: 6035310592.0000 - val_loss: 729695795
2.0000 - val_mse: 5968765440.0000
Epoch 41/100
20/20 - 0s - loss: 7951337472.0000 - mse: 6627951104.0000 - val_loss: 727731200
0.0000 - val_mse: 5958281216.0000
Epoch 42/100
20/20 - 0s - loss: 7674714112.0000 - mse: 6360557568.0000 - val_loss: 725676390
4.0000 - val_mse: 5948240384.0000
Epoch 43/100
20/20 - 0s - loss: 7380514816.0000 - mse: 6076774400.0000 - val_loss: 727313868
8.0000 - val_mse: 5975029760.0000
Epoch 44/100
20/20 - 0s - loss: 7934078976.0000 - mse: 6640679936.0000 - val_loss: 723736832
0.0000 - val_mse: 5949600768.0000
Epoch 45/100
20/20 - 0s - loss: 7746375168.0000 - mse: 6463487488.0000 - val_loss: 722481664
0.0000 - val_mse: 5947309056.0000
Epoch 46/100
20/20 - 0s - loss: 7398269440.0000 - mse: 6125786624.0000 - val_loss: 721486182

```

20/20 - 0s - loss: 7771188736.0000 - mse: 6508801536.0000 - val_loss: 720423219
 2.0000 - val_mse: 5946997248.0000
 Epoch 48/100
 20/20 - 0s - loss: 8040433152.0000 - mse: 6788059648.0000 - val_loss: 719411046
 4.0000 - val_mse: 5946891776.0000
 Epoch 49/100
 20/20 - 0s - loss: 7799008256.0000 - mse: 6556568064.0000 - val_loss: 718763110
 4.0000 - val_mse: 5950348288.0000
 Epoch 50/100
 20/20 - 0s - loss: 6789713408.0000 - mse: 5557129728.0000 - val_loss: 718834124
 8.0000 - val_mse: 5960915456.0000
 Epoch 51/100
 20/20 - 0s - loss: 8638150656.0000 - mse: 7415344128.0000 - val_loss: 716987187
 2.0000 - val_mse: 5952224256.0000
 Epoch 52/100
 20/20 - 0s - loss: 7518482432.0000 - mse: 6305369600.0000 - val_loss: 716874547
 2.0000 - val_mse: 5960777216.0000
 Epoch 53/100
 20/20 - 0s - loss: 7619158016.0000 - mse: 6415654400.0000 - val_loss: 714757785
 6.0000 - val_mse: 5949232128.0000
 Epoch 54/100
 20/20 - 0s - loss: 7115548160.0000 - mse: 5921592320.0000 - val_loss: 713780889
 6.0000 - val_mse: 5949010432.0000
 Epoch 55/100
 20/20 - 0s - loss: 7737314304.0000 - mse: 6552828928.0000 - val_loss: 713842585
 6.0000 - val_mse: 5959098368.0000
 Epoch 56/100
 20/20 - 0s - loss: 7752824832.0000 - mse: 6577965568.0000 - val_loss: 711890380
 8.0000 - val_mse: 5948970496.0000
 Epoch 57/100
 20/20 - 0s - loss: 6827664384.0000 - mse: 5662334464.0000 - val_loss: 711072921
 6.0000 - val_mse: 5950116864.0000
 Epoch 58/100
 20/20 - 0s - loss: 7868745728.0000 - mse: 6712658944.0000 - val_loss: 709830912
 0.0000 - val_mse: 5946941440.0000
 Epoch 59/100
 20/20 - 0s - loss: 7694989824.0000 - mse: 6548073984.0000 - val_loss: 708953036
 8.0000 - val_mse: 5947335168.0000
 Epoch 60/100
 20/20 - 0s - loss: 7888617472.0000 - mse: 6750798848.0000 - val_loss: 708241356
 8.0000 - val_mse: 5949317632.0000
 Epoch 61/100
 20/20 - 0s - loss: 7577272832.0000 - mse: 6448479232.0000 - val_loss: 708034457
 6.0000 - val_mse: 5956275200.0000
 Epoch 62/100
 20/20 - 0s - loss: 7607144960.0000 - mse: 6487306240.0000 - val_loss: 706200678
 4.0000 - val_mse: 5946892288.0000
 Epoch 63/100
 20/20 - 0s - loss: 7285133312.0000 - mse: 6174176256.0000 - val_loss: 705524684
 8.0000 - val_mse: 5949015552.0000
 Epoch 64/100
 20/20 - 0s - loss: 7563953152.0000 - mse: 6461809152.0000 - val_loss: 704459520
 0.0000 - val_mse: 5947175424.0000
 Epoch 65/100
 20/20 - 0s - loss: 7256065536.0000 - mse: 6162664960.0000 - val_loss: 703857561
 6.0000 - val_mse: 5949898240.0000
 Epoch 66/100
 20/20 - 0s - loss: 7699049984.0000 - mse: 6614322688.0000 - val_loss: 702875238
 4.0000 - val_mse: 5948748800.0000
 Epoch 67/100
 20/20 - 0s - loss: 7461968896.0000 - mse: 6386058752.0000 - val_loss: 703003545
 6.0000 - val_mse: 5958635008.0000
 Epoch 68/100
 Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js mse: 6684379136.0000 - val_loss: 702504140
 8.0000 - val_mse: 5962175488.0000

Epoch 69/100
20/20 - 0s - loss: 7048855040.0000 - mse: 5986472960.0000 - val_loss: 7019062784.0000 - val_mse: 5947124736.0000
Epoch 70/100
20/20 - 0s - loss: 7632611840.0000 - mse: 6564819456.0000 - val_loss: 7013072384.0000 - val_mse: 5949673984.0000
Epoch 71/100
20/20 - 0s - loss: 7523956736.0000 - mse: 6464632320.0000 - val_loss: 7003872768.0000 - val_mse: 5948945408.0000
Epoch 72/100
20/20 - 0s - loss: 6928394240.0000 - mse: 5877460992.0000 - val_loss: 7061898752.0000 - val_mse: 6015158272.0000
Epoch 73/100
20/20 - 0s - loss: 7915719680.0000 - mse: 6872370176.0000 - val_loss: 6986348032.0000 - val_mse: 5947149312.0000
Epoch 74/100
20/20 - 0s - loss: 7246109184.0000 - mse: 6210787328.0000 - val_loss: 6984709120.0000 - val_mse: 5953789440.0000
Epoch 75/100
20/20 - 0s - loss: 7279534080.0000 - mse: 6252423680.0000 - val_loss: 6972344832.0000 - val_mse: 5949637120.0000
Epoch 76/100
20/20 - 0s - loss: 7945454592.0000 - mse: 6926490624.0000 - val_loss: 6968816640.0000 - val_mse: 5954256384.0000
Epoch 77/100
20/20 - 0s - loss: 7364380160.0000 - mse: 6353499648.0000 - val_loss: 6954884608.0000 - val_mse: 5948405248.0000
Epoch 78/100
20/20 - 0s - loss: 7563870720.0000 - mse: 6561189376.0000 - val_loss: 6948780032.0000 - val_mse: 5950292480.0000
Epoch 79/100
20/20 - 0s - loss: 7189685760.0000 - mse: 6195126784.0000 - val_loss: 6939579392.0000 - val_mse: 5949045760.0000
Epoch 80/100
20/20 - 0s - loss: 7603304448.0000 - mse: 6616634368.0000 - val_loss: 6939217408.0000 - val_mse: 5956574720.0000
Epoch 81/100
20/20 - 0s - loss: 7862440448.0000 - mse: 6883596800.0000 - val_loss: 6931918848.0000 - val_mse: 5957104128.0000
Epoch 82/100
20/20 - 0s - loss: 6949983744.0000 - mse: 5978904576.0000 - val_loss: 6947010560.0000 - val_mse: 5979960832.0000
Epoch 83/100
20/20 - 0s - loss: 6962670080.0000 - mse: 5999291904.0000 - val_loss: 6931483648.0000 - val_mse: 5972137472.0000
Epoch 84/100
20/20 - 0s - loss: 7782282240.0000 - mse: 6826545664.0000 - val_loss: 6903334400.0000 - val_mse: 5951629824.0000
Epoch 85/100
20/20 - 0s - loss: 7155589120.0000 - mse: 6207434752.0000 - val_loss: 6891149312.0000 - val_mse: 5947025920.0000
Epoch 86/100
20/20 - 0s - loss: 7448748544.0000 - mse: 6508114944.0000 - val_loss: 6884016640.0000 - val_mse: 5947413504.0000
Epoch 87/100
20/20 - 0s - loss: 8228761088.0000 - mse: 7295587328.0000 - val_loss: 6913410048.0000 - val_mse: 5984267776.0000
Epoch 88/100
20/20 - 0s - loss: 6443811840.0000 - mse: 5518040576.0000 - val_loss: 6922072064.0000 - val_mse: 6000331776.0000
Epoch 89/100
20/20 - 0s - loss: 7580481536.0000 - mse: 6662232576.0000 - val_loss: 6866941440.0000 - val_mse: 5952543744.0000
20/20 - 0s - loss: 7359698944.0000 - mse: 6448898048.0000 - val_loss: 6865293824.0000 - val_mse: 5952543744.0000

```

4.0000 - val_mse: 5958179840.0000
Epoch 91/100
20/20 - 0s - loss: 7152232960.0000 - mse: 6248655872.0000 - val_loss: 685215078
4.0000 - val_mse: 5952261632.0000
Epoch 92/100
20/20 - 0s - loss: 7108498944.0000 - mse: 6212088320.0000 - val_loss: 686622412
8.0000 - val_mse: 5973504000.0000
Epoch 93/100
20/20 - 0s - loss: 7532124160.0000 - mse: 6642824704.0000 - val_loss: 683304704
0.0000 - val_mse: 5947438080.0000
Epoch 94/100
20/20 - 0s - loss: 6939122176.0000 - mse: 6056875008.0000 - val_loss: 682544640
0.0000 - val_mse: 5946891264.0000
Epoch 95/100
20/20 - 0s - loss: 7196422144.0000 - mse: 6321172480.0000 - val_loss: 683754444
8.0000 - val_mse: 5965988352.0000
Epoch 96/100
20/20 - 0s - loss: 7669862400.0000 - mse: 6801556480.0000 - val_loss: 681156556
8.0000 - val_mse: 5946951168.0000
Epoch 97/100
20/20 - 0s - loss: 7070863872.0000 - mse: 6209444864.0000 - val_loss: 681297152
0.0000 - val_mse: 5955244544.0000
Epoch 98/100
20/20 - 0s - loss: 7302370816.0000 - mse: 6447784448.0000 - val_loss: 680583782
4.0000 - val_mse: 5954943488.0000
Epoch 99/100
20/20 - 0s - loss: 7379338752.0000 - mse: 6531531776.0000 - val_loss: 682063360
0.0000 - val_mse: 5976516096.0000
Epoch 100/100
20/20 - 0s - loss: 6895243776.0000 - mse: 6054325248.0000 - val_loss: 679524864
0.0000 - val_mse: 5957855744.0000

```

In [455... visualize_nn(history, model, x_tr, y_tr, x_val, y_val)

Training Evaluation:

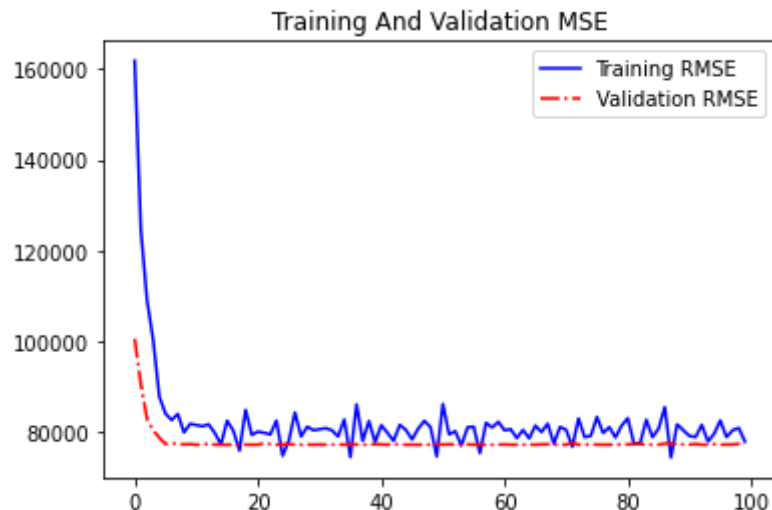
11/11 [=====] - 0s 2ms/step - loss: 7245062144.0000 - mse: 6407669248.0000

Validation Evaluation:

8/8 [=====] - 0s 1ms/step - loss: 6757675520.0000 - mse: 5920282624.0000

Train Evaluation RMSE: 80047.9184488891

Validation Evaluation RMSE: 76943.37284003088



Testing Best Fit Model

```
In [585... x_train = train_df.drop(['SalePrice'], axis=1)
y_train = train_df['SalePrice']
```

```
In [586... x_tr, x_val, y_tr, y_val = train_test_split(x_train, y_train, test_size=0.25, ra

x_tr = transformer.fit_transform(x_tr)
x_val = transformer.transform(x_val)

rfe = RFE(LinearRegression(normalize=False), n_features_to_select=80)
x_tr = rfe.fit_transform(x_tr, y_tr)
x_val = rfe.transform(x_val)

# Fitting Model
model = Ridge(alpha=10.0, max_iter=500, random_state=1000)
results = model.fit(x_tr, y_tr)

# Getting Predictions
y_tr_preds = results.predict(x_tr)
y_val_preds = results.predict(x_val)
```

```
In [587... get_results_preds(y_tr, y_tr_preds, y_val, y_val_preds)
```

R-Squared score for the training data: 0.8443560551411233

R-Squared score for the testing data: 0.8625669061039603

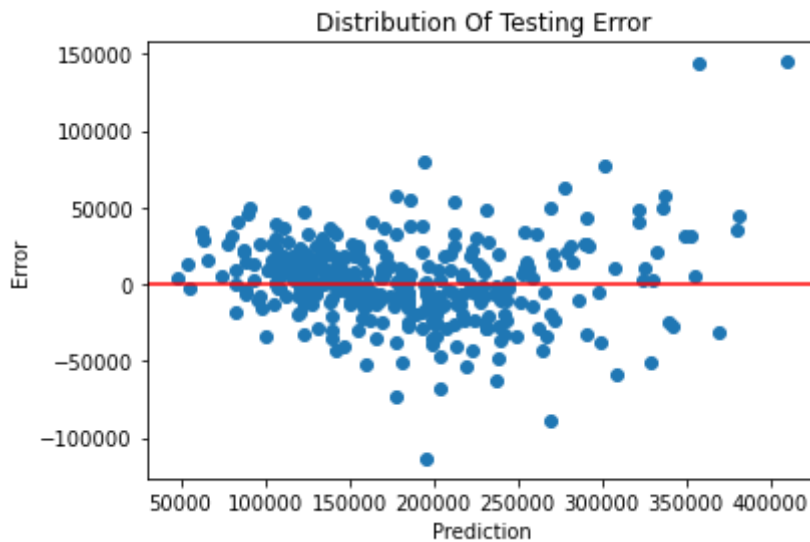
Root Mean Squared Error for the training data: 32014.381493841953

Root Mean Squared Error for the testing data: 27146.66162434128

```
In [588... train_error_distribution(y_tr, y_tr_preds)
```



```
In [589... test_error_distribution(y_val, y_val_preds)
```



Results

After using grid search to cross validate simple linear regressors, ridge regressors, lasso regressors, decision tree regressors, random forest regressors, and fully connected dense neural networks, the model that appeared to have the best fit in regard to the bias-variance tradeoff was a ridge regressor with $\alpha = 10.0$ and max iterations = 500 as parameters.

On the training data, this model returned an r-squared score of 0.844 and a root mean squared error score of approximately 32,014. On the validation data, the model returned an r-squared score of 0.863 and a root mean squared error score of approximately 27,146.

Note:

Because the test set does not contain a column for sale price, as competitors in the kaggle competition submit their final predictions for scoring, I will not be testing the best model on the test set. However, I've included the code I would use if I was testing the holdout set:

```
In [553... test_df = pd.read_csv('data/test.csv')

In [554... test_df = test_df.apply(impute_lot_frontage, axis=1)

In [555... replace_NaN(test_df, ['GarageType', 'GarageFinish', 'GarageQual', 'GarageCond'],

In [556... test_df.drop('Id', axis=1, inplace=True)

In [557... test_df.drop(['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature'], axis=1,

In [558... test_df.drop(['GarageArea', 'GarageYrBlt', 'TotRmsAbvGrd', '1stFlrSF'], axis=1,

In [559... test_df.drop(low_var, axis=1, inplace=True)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
inplace=True)

```

In [561...] test_df.dropna(inplace=True)

In [562...] map_function(test_df, ['ExterQual', 'ExterCond', 'BsmtQual', 'BsmtCond', 'Heatin

In [563...] test_df['Bath_Fireplaces'] = test_df['Fireplaces'] + test_df['FullBath']

In [568...] x_train = train_df.drop(['SalePrice'], axis=1)
            y_train = train_df['SalePrice']

In [566...] x_test = test_df

In [570...] x_train = transformer.fit_transform(x_train)
            x_test = transformer.transform(x_test)

            rfe = RFE(LinearRegression(normalize=False), n_features_to_select=80)
            x_train = rfe.fit_transform(x_train, y_train)
            x_test = rfe.transform(x_test)

            # Fitting Best Model
            model = Ridge(alpha=10.0, max_iter=500, random_state=1000)
            results = model.fit(x_train, y_train)

            # Getting Predictions
            y_train_preds = results.predict(x_train)
            y_test_preds = results.predict(x_test)

In [574...] get_results_preds(y_train, y_train_preds, y_test, y_test_preds)

R-Squared score for the training data: 0.8509148503685335

Root Mean Squared Error for the training data: 30596.69300007709

In [ ]: train_error_distribution(y_train, y_train_preds)

In [ ]: test_error_distribution(y_test, y_test_preds)

```