Answer to Problem Statement - Part II

Assignment Part-II

The following questions are the second part of the graded assignment. Please submit the answers in one PDF file. For writing normal text, please use MS Word (or similar software that can convert documents to PDF). For equations and figures, you can write/draw them on a blank sheet of paper using a pen, click images and upload them in the same Word document.

The final submission will be in the form of one PDF file. A sample PDF to illustrate the submission format is provided below.

Note: **DO NOT** copy or paste answers from *anywhere,* and type the answers in your own words; your solution files will be tested using automatic plagiarism checkers and will attract a heavy penalty if plagiarism is detected.

Please limit your answers to less than 500 words per question.

Question 1

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Ans 1:

The optimal value of alpha regression for ridge is 0.003352 and for las
so regression is 23.101297

```
params = {'alpha': np.logspace(-5, 5, num=100)}

ridge_model, y_test_rdg_predicted, y_train_rdg_pred = get_ml(X_sc_train, y_train, X_sc_test, params,
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
Optimum alpha for Ridge() is 23.101297
Ridge()  for alfa=  23.10129700083158
R2score y_train :  0.9082828786228985
R2score y_test :  0.9080030612348721
mean_squared_error (y_train) :  0.11827143778488118
mean_squared_error (y_test) :  0.13039944168015027
```

```
params = {'alpha': np.logspace(-5, 5, num=100)}

lasso_model, y_test_predicted_lasso,y_train_pred_lasso = get_ml(X_sc_train, y_train, X_sc_test, param
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
Optimum alpha for Lasso() is 0.003352
Lasso()  for alfa=  0.003351602650938841
R2score y_train :  0.9020320218436748
R2score y_test :  0.9121262564240556
mean_squared_error (y_train) :  0.12223532772895537
mean_squared_error (y_test) :  0.12744376938335306
```

When the values chosen double of alpha the corresponding values changes
to

```
params = {'alpha': [46.202594]}
ridge_model, y_test_rdg_predicted, y_train_rdg_pred = get_ml(X_sc_train, y_train, X_sc_test, params,
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
Optimum alpha for Ridge() is 46.202594
Ridge()  for alfa=  46.202594
R2score y_train :  0.9071615226559888
R2score y_test :  0.9081145550471084
mean_squared_error (y_train) :  0.11899224904280559
mean_squared_error (y_test) :  0.1303204002538135
```

```
params = {'alpha': [0.006704]}

lasso_model, y_test_predicted_lasso,y_train_pred_lasso = get_ml(X_sc_train, y_train, X_sc_test, params
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
Optimum alpha for Lasso() is 0.006704
Lasso()  for alfa=  0.006704
R2score y_train :  0.8953899319335854
R2score y_test :  0.9085550559682178
mean_squared_error (y_train) :  0.12631106908188974
mean_squared_error (y_test) :  0.13000764547790272
```

In the above details the MSE error is lowest for alpha r
egression for ridge is 0.003352 and for lasso regression is 23.101297

RIDGE:

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
Optimum alpha for Ridge() is 23.101297
Ridge()  for alfa=  23.10129700083158
R2score y_train :  0.9082828786228985
R2score y_test :  0.9080030612348721
mean_squared_error (y_train) :  0.11827143778488118
mean_squared_error (y_test) :  0.13039944168015027

Fitting 5 folds for each of 1 candidates, totalling 5 fits
Optimum alpha for Ridge() is 46.202594
Ridge()  for alfa=  46.202594
R2score y_train :  0.9071615226559888
R2score y_test :  0.9081145550471084
mean_squared_error (y_train) :  0.11899224904280559
mean_squared_error (y_test) :  0.1303204002538135
```

LASSO:

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
Optimum alpha for Lasso() is 0.003352
Lasso()  for alfa=  0.003351602650938841
R2score y_train :  0.9020320218436748
R2score y_test :  0.9121262564240556
mean_squared_error (y_train) :  0.12223532772895537
mean_squared_error (y_test) :  0.12744376938335306


Fitting 5 folds for each of 1 candidates, totalling 5 fits
Optimum alpha for Lasso() is 0.006704
Lasso()  for alfa=  0.006704
R2score y_train :  0.8953899319335854
R2score y_test :  0.9085550559682178
mean_squared_error (y_train) :  0.12631106908188974
mean_squared_error (y_test) :  0.13000764547790272
```

There is not much change in MSE for ridge but for Lasso it’s a very min or change around 0.04 and .03

Question 2

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Ans 2:

Lasso model se the coefficient of many of feature to zero, it helps to focus on few feartures compared to ridge. Around 45 columns coefficients set to zero or eliminated from the model

	Ridge 23.101297	Lasso 0.003352
Id	-0.003206	-0.000000
MSSubClass	-0.015713	-0.002608
LotFrontage	0.008813	0.004676
LotArea	0.026593	0.027990
OverallQual	0.076978	0.091544
OverallCond	0.048251	0.045153
YearBuilt	0.029777	0.035482
YearRemodAdd	0.013740	0.013079
MasVnrArea	-0.003251	-0.000000
BsmtFinSF1	0.038371	0.032061
BsmtUnfSF	0.015161	0.000899
1stFlrSF	0.103776	0.114220
2ndFlrSF	0.085754	0.098292
BsmtFullBath	0.020251	0.014613

	Ridge 23.101297	Lasso 0.003352
FullBath	0.008549	0.003679
BedroomAbvGr	0.000415	0.000000
TotRmsAbvGrd	0.013887	0.003793
Fireplaces	0.016782	0.018423
GarageArea	0.022992	0.025776
WoodDeckSF	0.009087	0.006666
OpenPorchSF	0.008973	0.007609
MoSold	-0.000723	-0.000000
YrSold	-0.006259	-0.001219
MSZoning_RL	0.022500	0.016515
MSZoning_RM	-0.001613	-0.003764
Street_Pave	0.008719	0.007387
Alley_NONE	-0.003251	-0.000000
LotShape_Reg	-0.001816	-0.000947
LandContour_Lvl	0.000349	-0.000000
LotConfig_CulDSac	0.010695	0.008389
LotConfig_Inside	-0.001517	-0.000000
Neighborhood_CollgCr	-0.006131	-0.000000
Neighborhood_Edwards	-0.023533	-0.016027

	Ridge 23.101297	Lasso 0.003352
Neighborhood_Gilbert	-0.003638	0.000000
Neighborhood_NAmes	-0.015170	-0.005435
Neighborhood_NWAmes	-0.009490	-0.002326
Neighborhood_NridgHt	0.014687	0.015007
Neighborhood_OldTown	-0.011377	-0.005617
Neighborhood_Sawyer	-0.008656	-0.001518
Neighborhood_Somerst	0.019495	0.017048
Condition1_Feedr	0.003938	-0.000000
Condition1_Norm	0.022639	0.018442
Condition2_Norm	0.010747	0.007460
BldgType_TwnhsE	0.011061	0.000000
HouseStyle_1Story	-0.013321	-0.000000
HouseStyle_2Story	0.003952	0.000000
RoofStyle_Gable	-0.032021	-0.001604
RoofStyle_Hip	-0.026971	0.000000
RoofMatl_CompShg	0.005625	0.000000
Exterior1st_HdBoard	-0.016632	-0.002134
Exterior1st_MetalSd	0.002458	0.002081
Exterior1st_Plywood	-0.006621	-0.000000

	Ridge 23.101297	Lasso 0.003352
Exterior1st_VinylSd	0.000653	0.000000
Exterior1st_Wd Sdng	-0.016915	-0.001089
Exterior2nd_HdBoard	0.014479	-0.000000
Exterior2nd_MetalSd	0.004982	0.000000
Exterior2nd_Plywood	0.005988	-0.000000
Exterior2nd_VinylSd	0.007680	0.004339
Exterior2nd_Wd Sdng	0.018848	0.000000
MasVnrType_BrkFace	0.009855	-0.000000
MasVnrType_NONE	0.011166	0.000000
MasVnrType_Stone	0.007281	0.000000
ExterQual_Gd	0.000870	0.000000
ExterQual_TA	-0.009850	-0.010620
ExterCond_Gd	-0.000709	-0.000000
ExterCond_TA	0.004065	0.001385
Foundation_CBlock	0.006478	-0.000000
Foundation_PConc	0.015847	0.016892
BsmtQual_Gd	0.001661	0.001017
BsmtQual_TA	-0.007465	-0.004235
BsmtCond_TA	0.008732	0.007296

	Ridge 23.101297	Lasso 0.003352
BsmtExposure_Gd	0.015704	0.010943
BsmtExposure_Mn	0.001726	0.000000
BsmtExposure_No	0.003357	0.000000
BsmtFinType1_BLQ	-0.000987	0.000000
BsmtFinType1_GLQ	0.001656	0.000424
BsmtFinType1_LwQ	0.001436	0.000000
BsmtFinType1_Rec	-0.004074	-0.000000
BsmtFinType1_Unf	-0.004930	-0.000000
BsmtFinType2_Unf	-0.001524	0.000000
Heating_GasA	-0.006752	-0.000149
HeatingQC_Gd	-0.004495	-0.001287
HeatingQC_TA	-0.008529	-0.007013
CentralAir_Y	0.016475	0.013216
Electrical_SBrkr	0.000243	0.000000
KitchenQual_Gd	-0.012949	-0.000000
KitchenQual_TA	-0.014392	-0.003466
Functional_Typ	0.021270	0.018130
FireplaceQu_Gd	0.005436	0.006183
FireplaceQu_NONE	-0.002721	-0.001060

	Ridge 23.101297	Lasso 0.003352
FireplaceQu_TA	0.000531	0.000000
GarageType_Attchd	0.013684	0.002285
GarageType_BuiltIn	0.003603	-0.000000
GarageType_Detchd	0.011249	0.000000
GarageType_NONE	-0.002359	-0.010029
GarageFinish_NONE	-0.002359	-0.000000
GarageFinish_RFn	-0.002169	0.000000
GarageFinish_Unf	-0.012653	-0.008085
GarageQual_NONE	-0.002359	-0.001758
GarageQual_TA	0.002922	0.000521
GarageCond_NONE	-0.002359	-0.000580
GarageCond_TA	0.003551	0.001124
PavedDrive_Y	0.012138	0.007959
PoolQC_NONE	0.005205	0.004741
Fence_MnPrv	0.004137	0.000000
Fence_NONE	0.003776	0.000000
MiscFeature_NONE	0.005678	0.001571
SaleType_New	0.003550	0.000000
SaleType_WD	-0.003165	-0.000000

	Ridge 23.101297	Lasso 0.003352
SaleCondition_Normal	0.025403	0.019982
SaleCondition_Partial	0.023027	0.025433

```
Incase we want to work on top 10 features then the following features a
re best to focus on.

      Index(['1stFlrSF', '2ndFlrSF', 'OverallQual', 'OverallCond', 'Yea
      rBuilt',
             'BsmtFinSF1', 'LotArea', 'GarageArea', 'SaleCondition_Part
      ial',
             'SaleCondition_Normal'],
            dtype='object')
```

Question 3

After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Ans 3:

The five most important predictor variable can be `Lasso Coefficients:`

```
First top5 are mentioned below.
      '1stFlrSF',
      '2ndFlrSF',
      'OverallQual',
      'OverallCond',
      'YearBuilt'

Incase these 5 features are not availale it can choose the below featur
es

'BsmtFinSF1’
'LotArea'
'GarageArea'
'SaleCondition_Partial'
'SaleCondition_Normal'
```

Question 4

How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

Ans 4:

A model is robust and generalizable, following steps would help to achieve it.

cross-validation:

> We have used cross-validation with ridge pipeline to assess the model performance to get an accurate estimate. The approach was adopted to train on subset of data and evaluate on the test data.

Overfitting:

> The R-squared value on the training set was 0.778. In the test set, the R-squared value has increased to 0.822. Usually, if a model is overfitting, we would expect the performance (as measured by metrics like R-squared) to decrease on the test set. However, in this case, the performance has actually increased. In fact, it appears to be performing well on unseen data, which suggests that it is generalizing well and not overfitting to the training data.
>
> However, be cautious about concluding that the model is perfect, because other factors could be at play. For instance, if the test set is not representative of the

broader population the model could be applied to, the model might not perform as well in practice. Similarly, the model might be underfitting, meaning it's too simple to capture all the relevant patterns in the data. It's always good to further validate the model using different subsets of data or different metrics.

Skew of Data:

Positive skewness found in the residual indicates a distribution with a long tail to the right. With log operation the skewed data become normalized.

Feature Engineering:

We applied measures like EDA operation on the dataset to remove the NA values for all columns and used label encoder for the categorical data. With co-relation matrix, features are selected those who have high co-relation (>.7) and re-filtered with high p-value columns and finalized the columns that can be used for the model.

Model Selection:

Used techniques like grid search or random search to find the best model hyper parameters, by ensuring that the model is not too complex for the data.

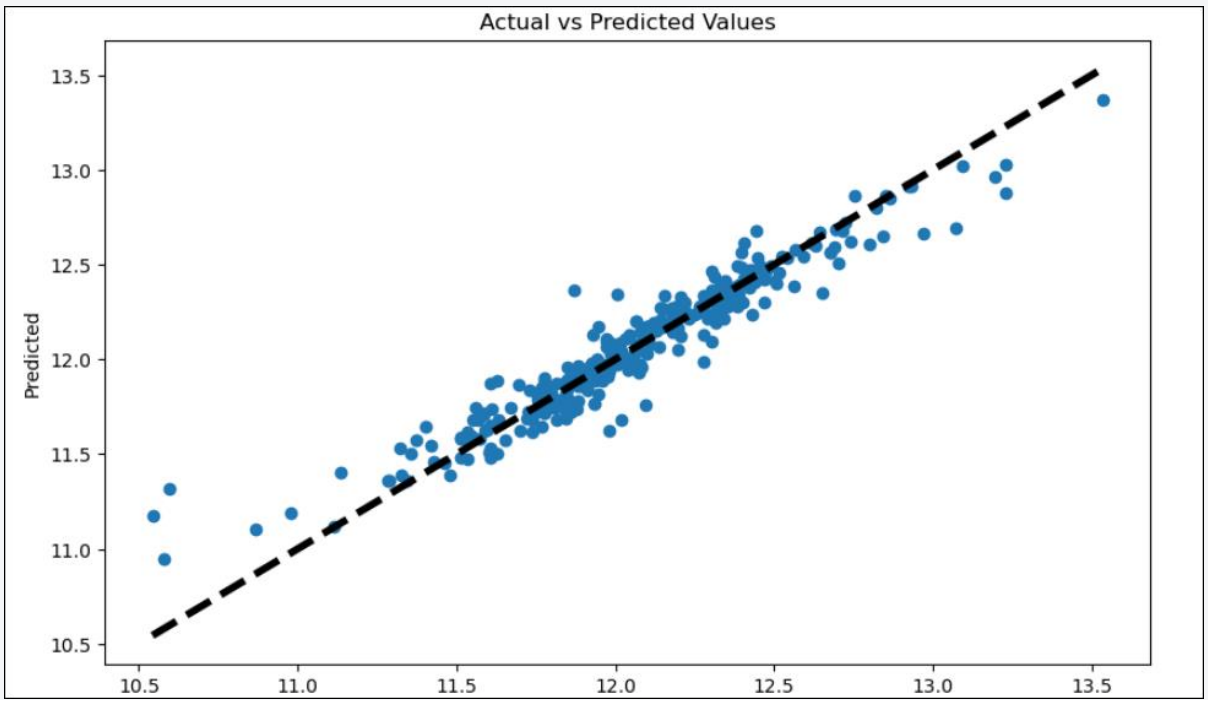
Regularization:

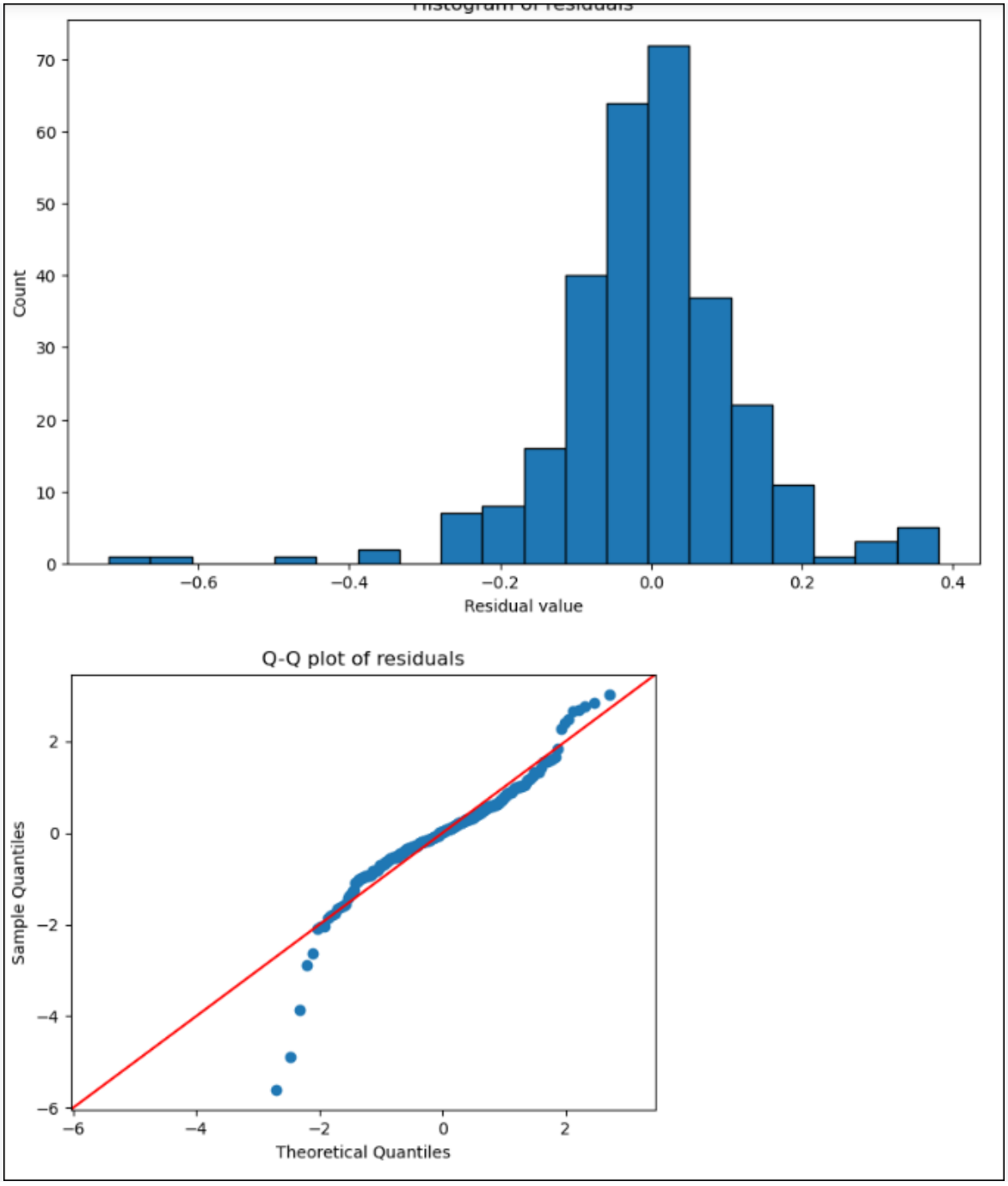
Applied techniques like L1 (Ridge) or L2(Lasso) regularization in regression models.

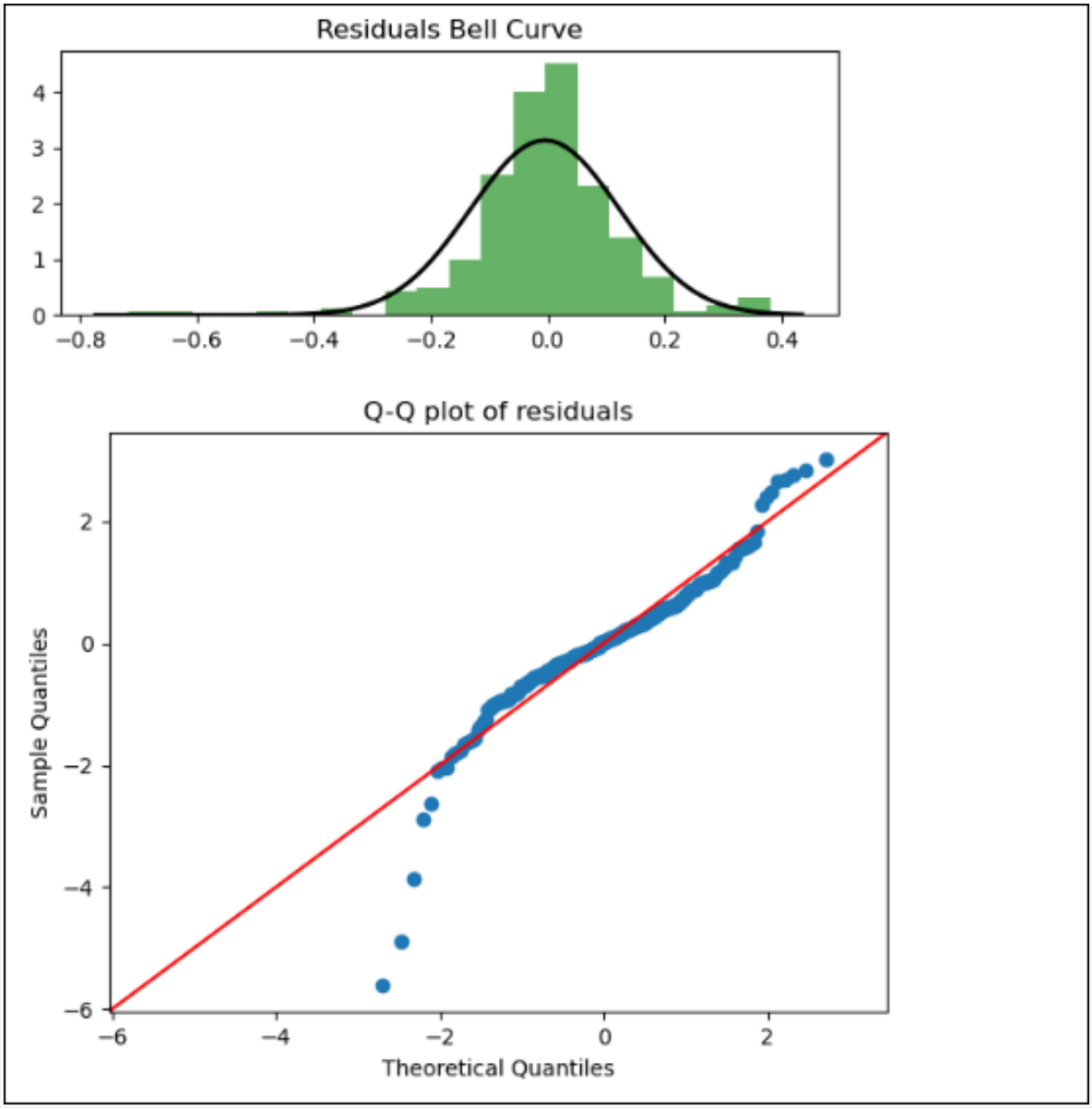
Q-Q Plot:

Quantile-Quantile Plot (Q-Q plot): The data points fall along a roughly straight line at a 45-degree angle, seems data is normally distributed.

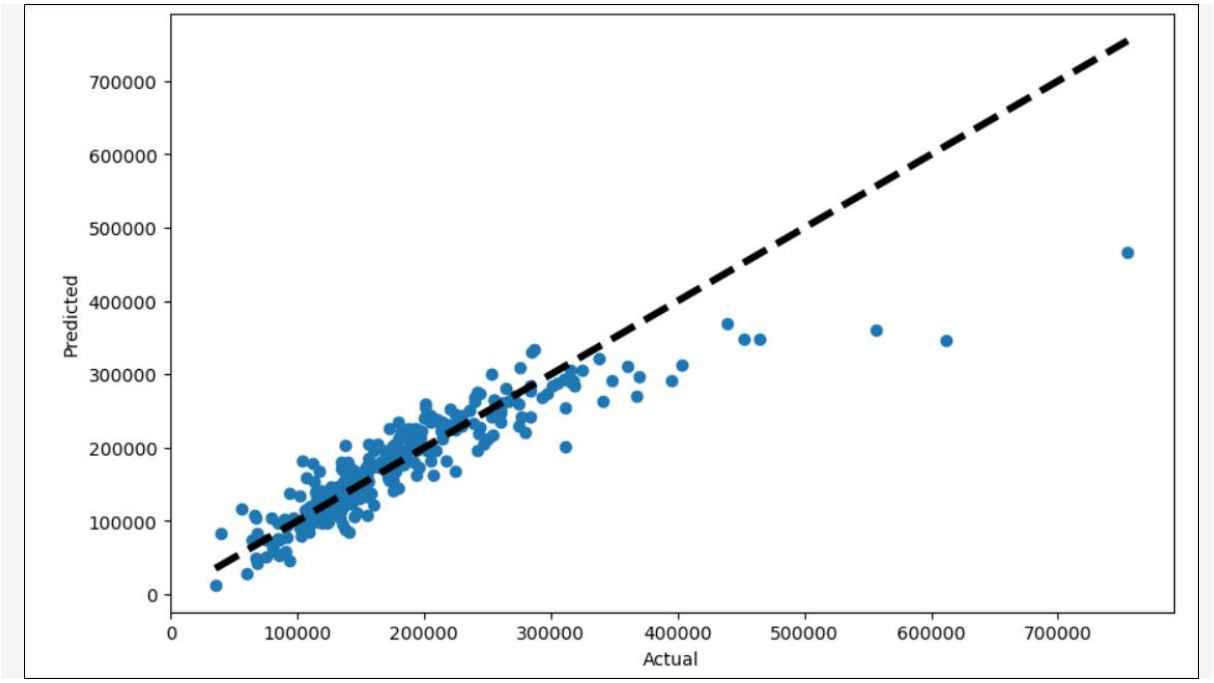
Avoid Data Leakage: Make sure that your test data does not leak any information that could bias the model's predictions. For instance, if you're building a model to predict house prices, make sure that the features you use in your model are all available before the price was known.

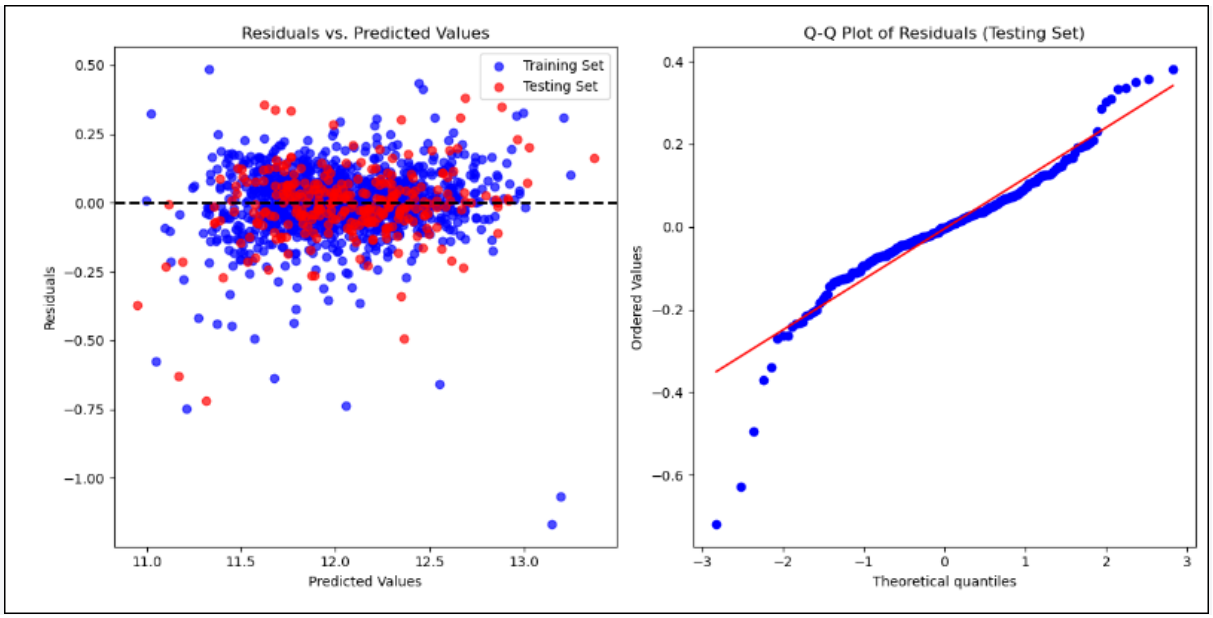


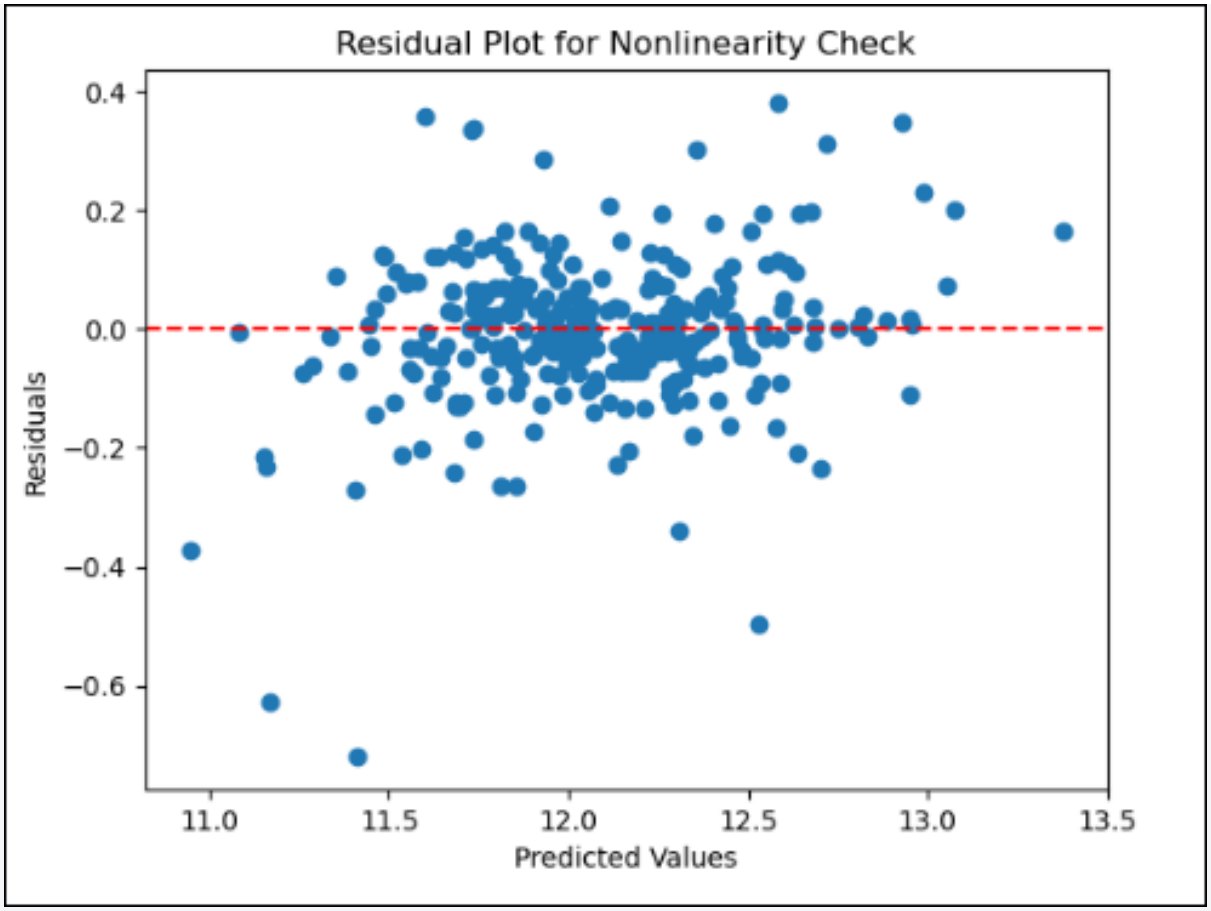

Count
Residual value
70
60
50
40
30
20
10
0
−0.6
−0.4
−0.2
0.0
0.2
0.4
Q-Q plot of residuals
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−6
−6
−4
−2
0
2

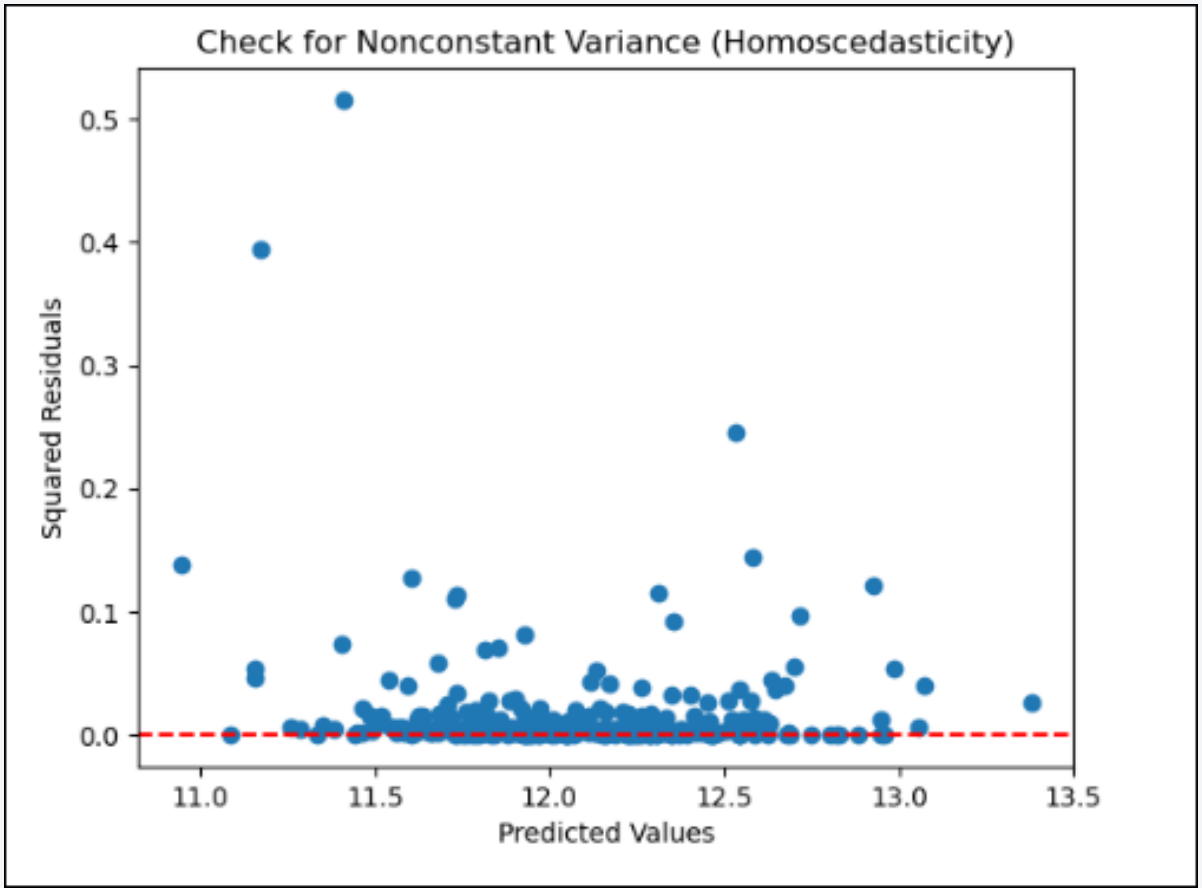


Assumptions: For linear regression models, the assumptions of normality, linearity, homoscedasticity (having constant variance), and independence of errors.


Predicted
Actual
0
100000
200000
300000
400000
500000
600000
700000


Residuals vs. Predicted Values
Training Set
Testing Set
Residuals
Predicted Values
Q-Q Plot of Residuals (Testing Set)
Ordered Values
Theoretical quantiles


Residual Plot for Nonlinearity Check
Residuals
Predicted Values
0.4
0.2
0.0
−0.2
−0.4
−0.6
11.0
11.5
12.0
12.5
13.0
13.5


Check for Nonconstant Variance (Homoscedasticity)
Squared Residuals
Predicted Values
0.5
0.4
0.3
0.2
0.1
0.0
11.0
11.5
12.0
12.5
13.0
13.5

```
Bias vs Variance
```