Problem Statement - Part II

Q1. 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: It is important to regularize coefficients and improve the prediction accuracy also with the decrease in variance, and making the model interpretably.

Ridge regression, uses a tuning parameter called lambda as the penalty is square of magnitude of coefficients which is identified by cross validation. Residual sum or squares should be small by using the penalty. The penalty is lambda times sum of squares of the coefficients, hence the coefficients that have greater values gets penalized. As we increase the value of lambda the variance in model is dropped and bias remains constant. Ridge regression includes all variables in final model unlike Lasso Regression.

Lasso regression, uses a tuning parameter called lambda as the penalty is absolute value of magnitude of coefficients which is identified by cross validation. As the lambda value increases Lasso shrinks the coefficient towards zero and it make the variables exactly equal to 0. Lasso also does variable selection. When lambda value is small it performs simple linear regression and as lambda value increases, shrinkage takes place and variables with 0 value are neglected by the model.

Q2. After building the model, you realized 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:

1. OverallQual
2. OverallCond
3. TotalBsmtSF
4. GarageArea
5. GrLivArea

Q3. How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Ans: The model should be as simple as possible, though its accuracy will decrease but it will be more robust and generalizable. It can be also understood using the Bias-Variance trade-off. The simpler the model the more the bias but less variance and more generalizable. Its implication in terms of accuracy is that a robust and generalizable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data.

Bias: Bias is error in model, when the model is weak to learn from the data. High bias means model is unable to learn details in the data. Model performs poor on training and testing data

Variance: Variance is error in model, when model tries to over learn from the data. High variance means model performs exceptionally well on training data as it has very well trained on this of data but performs very poor on testing data as it was unseen data for the model. It is important to have balance in Bias and Variance to avoid overfitting and under-fitting of data.

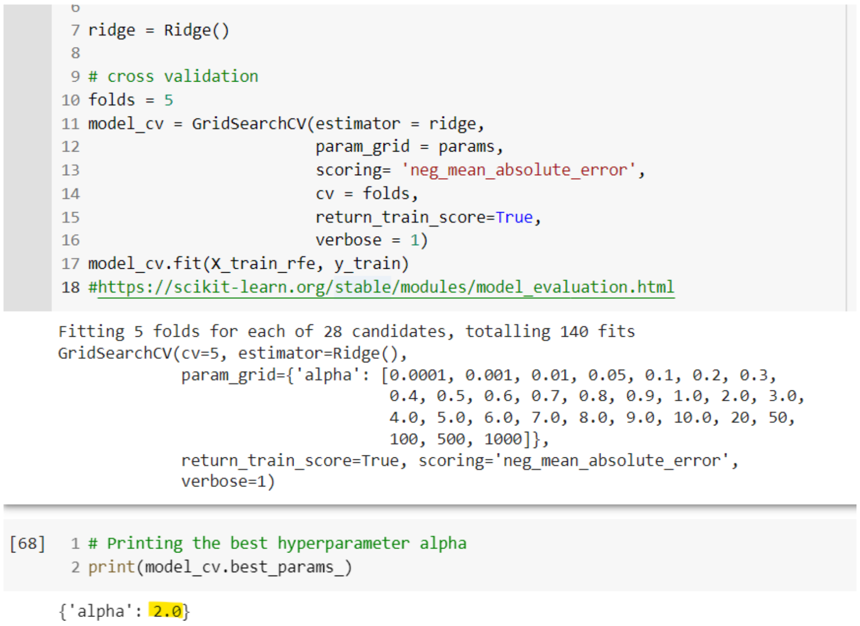
In our model, we got highest train accuracy(92) for Lasso, but test accuracy(72) is low. In this case it is treated as overfitting. However Ridge(as we have high number of variables), got average accuracy(88) with test accuracy(83) which is almost nearer. So here the robust model is Ridge.

Q4. 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:

Ridge:

The optimal value of alpha for Ridge is **2.0.**



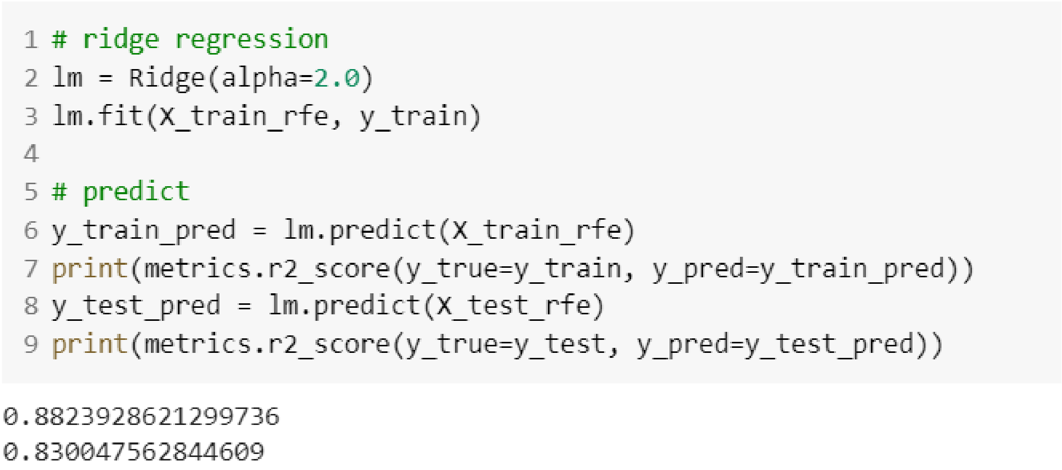
```
ridge = Ridge()

# cross validation
folds = 5
model_cv = GridSearchCV(estimator = ridge,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score=True,
                        verbose = 1)
model_cv.fit(X_train_rfe, y_train)
#https://scikit-learn.org/stable/modules/model_evaluation.html
```

```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
GridSearchCV(cv=5, estimator=Ridge(),
             param_grid={'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3,
                                   0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
                                   4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50,
                                   100, 500, 1000]},
             return_train_score=True, scoring='neg_mean_absolute_error',
             verbose=1)
```

```
[68]  # Printing the best hyperparameter alpha
      print(model_cv.best_params_)
```

```
{'alpha': 2.0}
```

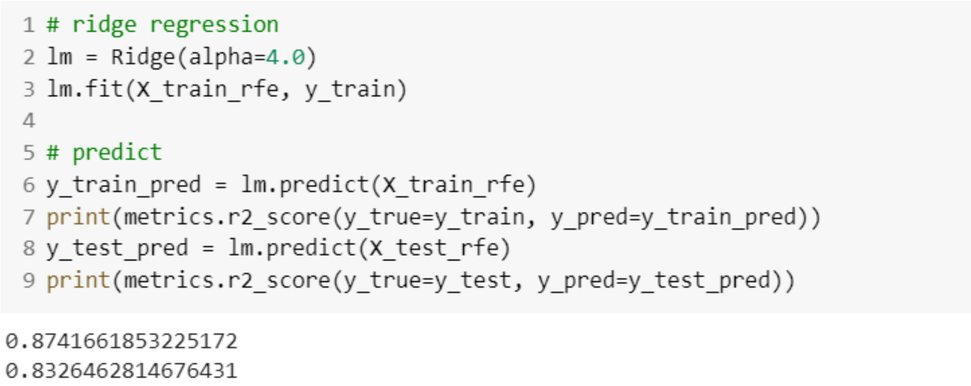


```python
# ridge regression
lm = Ridge(alpha=2.0)
lm.fit(X_train_rfe, y_train)

# predict
y_train_pred = lm.predict(X_train_rfe)
print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = lm.predict(X_test_rfe)
print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
0.8823928621299736
0.830047562844609
```

When the alpha value is doubled, train accuracy is decreased.



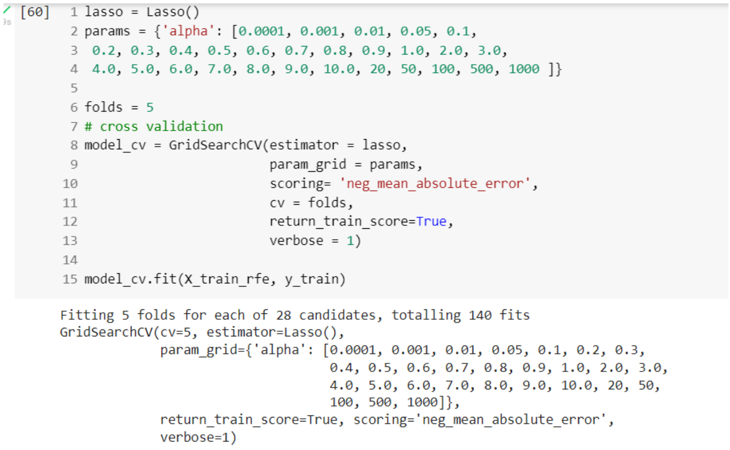
```python
# ridge regression
lm = Ridge(alpha=4.0)
lm.fit(X_train_rfe, y_train)

# predict
y_train_pred = lm.predict(X_train_rfe)
print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = lm.predict(X_test_rfe)
print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
0.8741661853225172
0.8326462814676431
```

Lasso:

The Optimal Value of alpha for Lasso is 20.



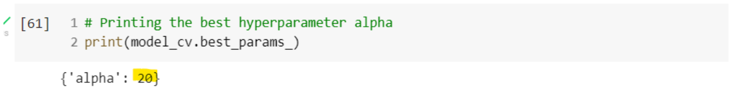
[60]

```python
lasso = Lasso()
params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1,
 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50, 100, 500, 1000 ]}

folds = 5
# cross validation
model_cv = GridSearchCV(estimator = lasso,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score=True,
                        verbose = 1)

model_cv.fit(X_train_rfe, y_train)
```

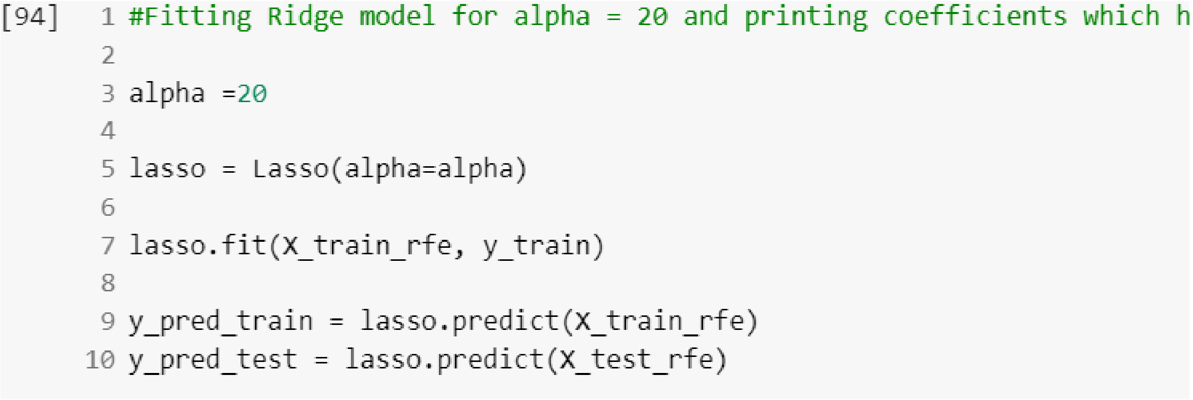
```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3,
                                   0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
                                   4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50,
                                   100, 500, 1000]},
             return_train_score=True, scoring='neg_mean_absolute_error',
             verbose=1)
```



[61]

```python
# Printing the best hyperparameter alpha
print(model_cv.best_params_)
```

```
{'alpha': 20}
```



[94]

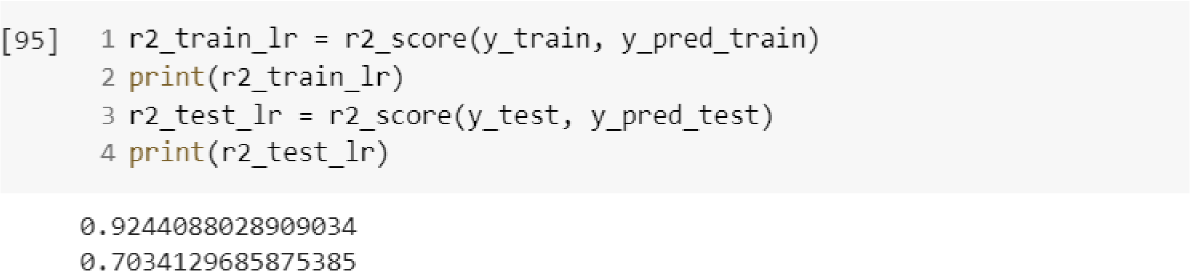
```
#Fitting Ridge model for alpha = 20 and printing coefficients which h

alpha =20

lasso = Lasso(alpha=alpha)

lasso.fit(X_train_rfe, y_train)

y_pred_train = lasso.predict(X_train_rfe)
y_pred_test = lasso.predict(X_test_rfe)
```

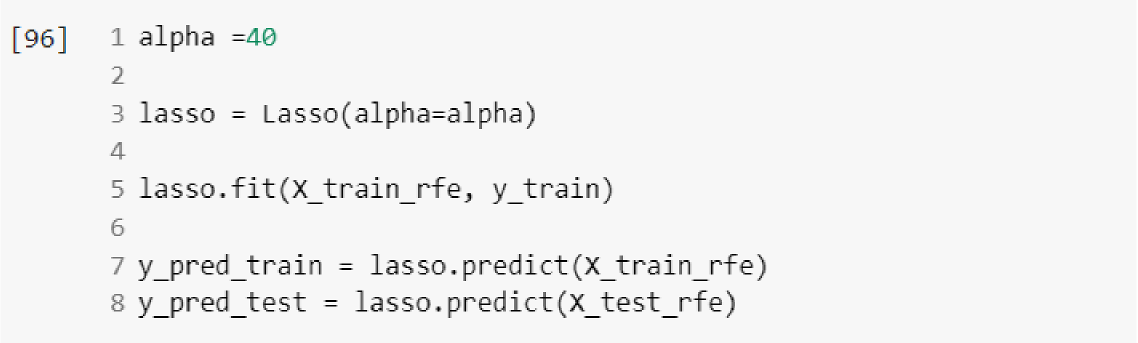


[95]

```
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
```

```
0.9244088028909034
0.7034129685875385
```

When the alpha value is doubled, train accuracy decreased.



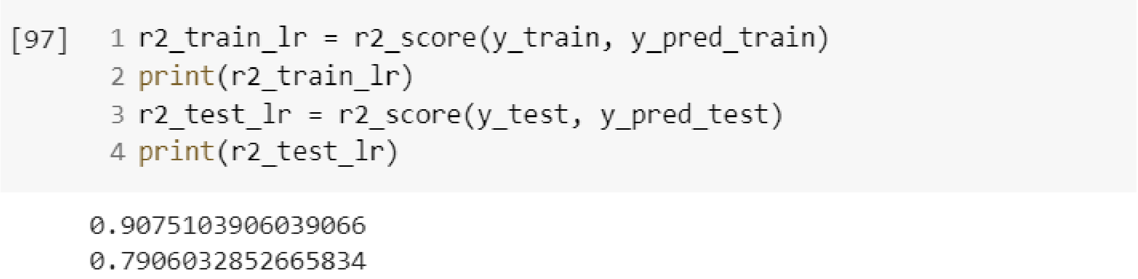
[96]

```
alpha =40

lasso = Lasso(alpha=alpha)

lasso.fit(X_train_rfe, y_train)

y_pred_train = lasso.predict(X_train_rfe)
y_pred_test = lasso.predict(X_test_rfe)
```



[97]

```
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
```

```
0.9075103906039066
0.7906032852665834
```

After changes, we can see Ridge(with 87% and 83% train and test R2 respectively) is still the robust model as the train and test R2 values are closer compared to Lasso(seems to be overfitting with 90% and 79% train and test accuracies respectively)