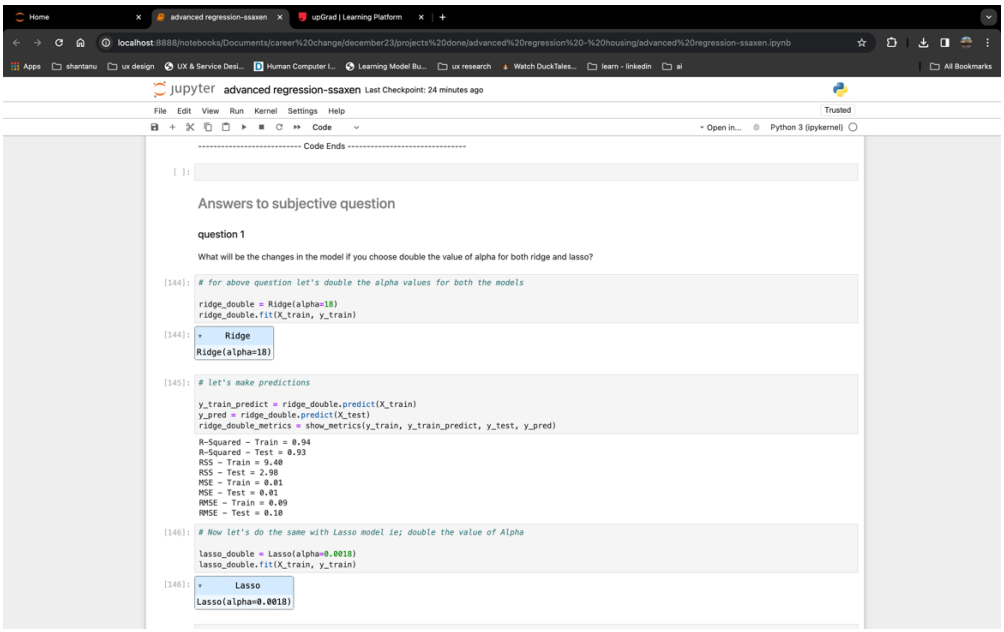
## Question 1

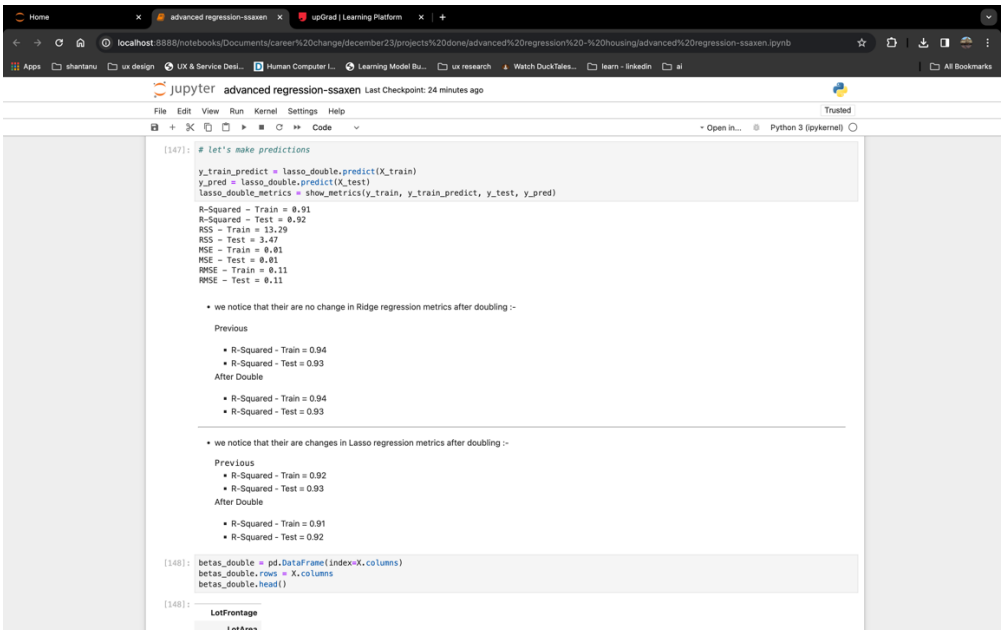
What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Ans :

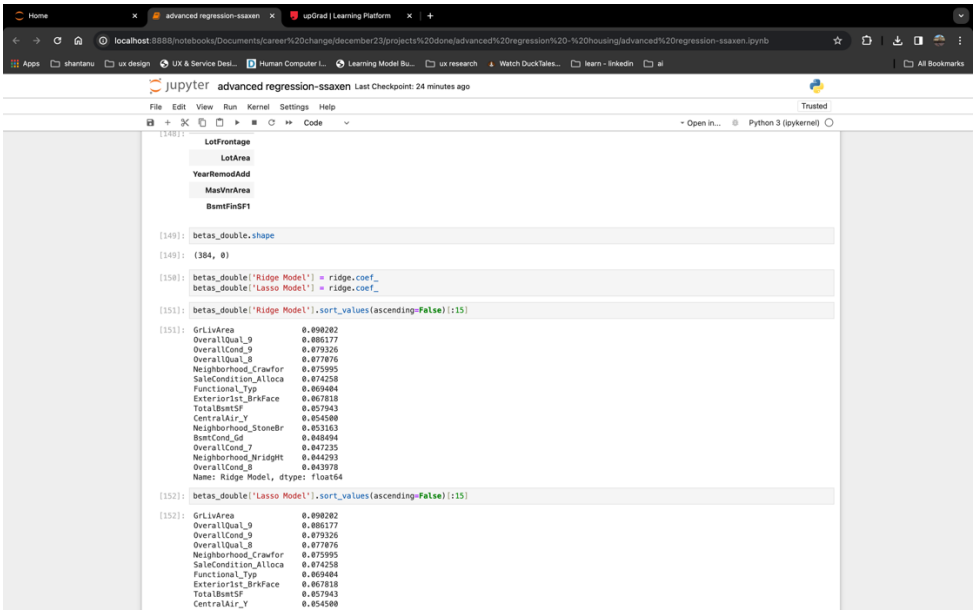
The optimal value of lambda for Ridge Regression is 9. The optimal value of lambda for Lasso is 0.001.

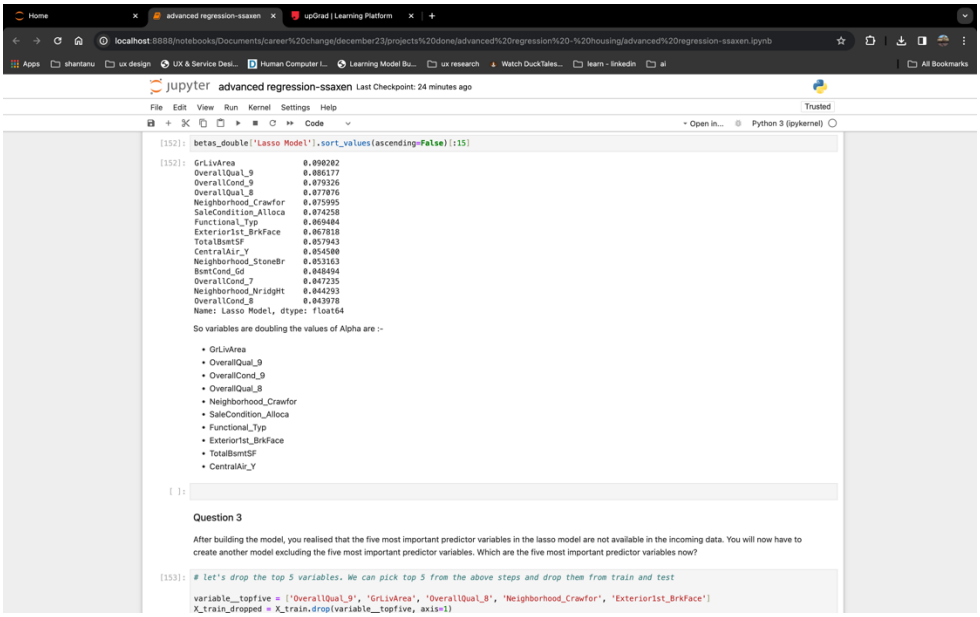
To find the changes in model where we double the value of Alpha for both models we need to build is in Jupyter and see. [ Attaching the screenshot below of the Jupyter file coded ].



# SUBJECTIVE ANSWER – ADVANCED REGRESSION ASSIGNMENT




The most important predictor variables after the change is implemented are (refer the attached jupter file in git for this):-

- GrLivArea
- OverallQual_9
- OverallCond_9
- OverallQual_8
- Neighborhood_Crawfor
- SaleCondition_Alloca
- Functional_Typ
- Exterior1st_BrkFace
- TotalBsmtSF
- CentralAir_Y

SHANTANU SAXENA

## 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 :

Considering the value of R-Squared - Test = 0.93 for Lasso and Regression both we can select based on the use case (refer screenshot in the above question or jupyter file). So let's say if our primary goal is to select features than we can opt for Lasso Regression model.

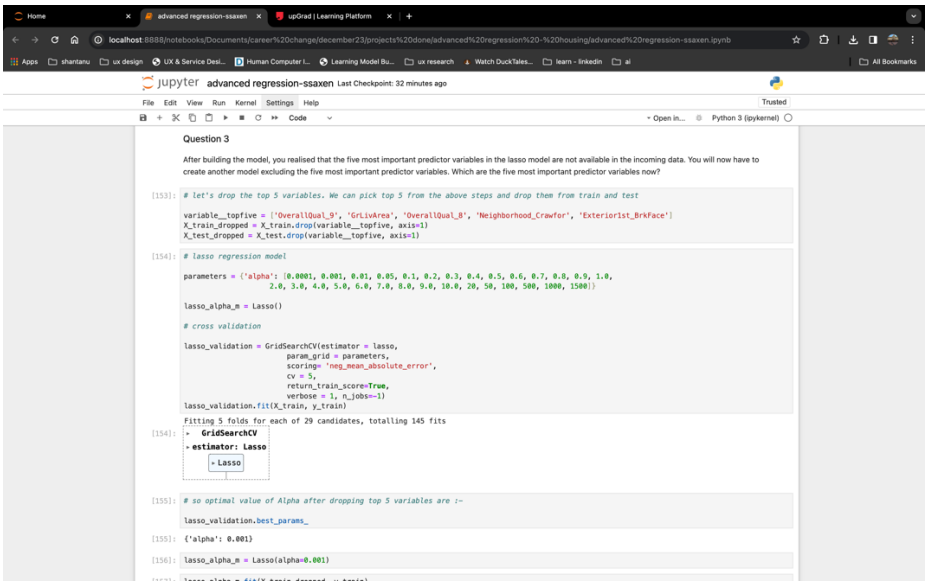
## Question 3

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?

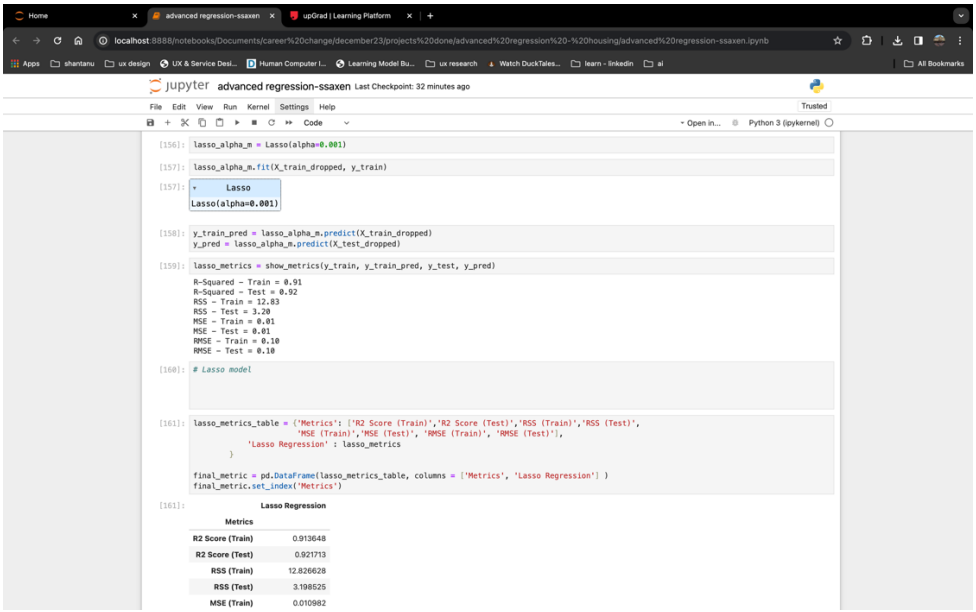
Ans:

After excluding top 5 we get these as models :- (refer below screenshots or the jupyter linked with assignment)

- 2ndFlrSF
- 1stFlrSF
- Functional_Typ
- Neighborhood_Somerst
- TotalBsmtSF



# SUBJECTIVE ANSWER – ADVANCED REGRESSION ASSIGNMENT

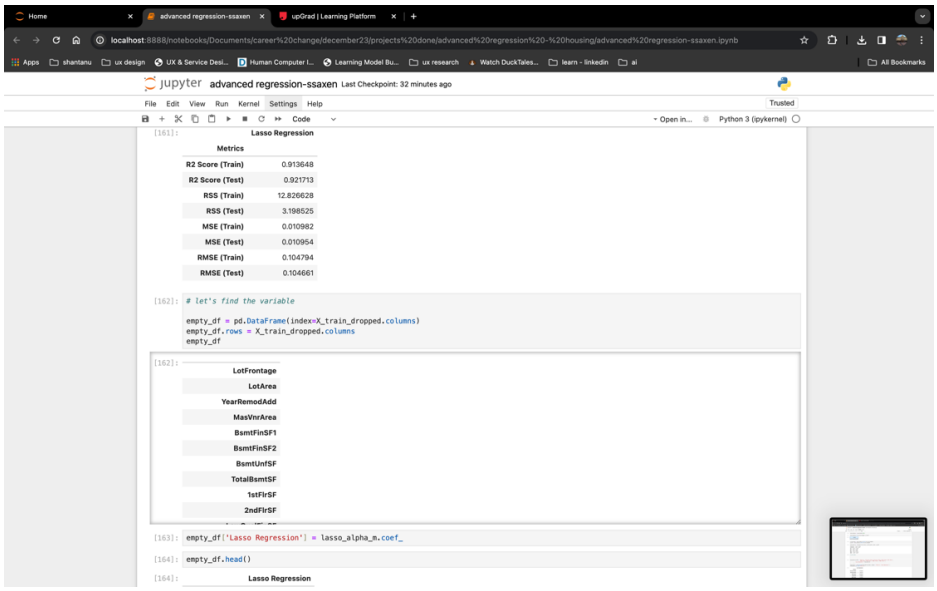


jupyter advanced regression-ssaxen Last Checkpoint: 32 minutes ago

```
[156]: lasso_alpha_m = Lasso(alpha=0.001)

[157]: lasso_alpha_m.fit(X_train_dropped, y_train)

[157]: Lasso
       Lasso(alpha=0.001)

[158]: y_train_pred = lasso_alpha_m.predict(X_train_dropped)
       y_pred = lasso_alpha_m.predict(X_test_dropped)

[159]: lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)

       R-Squared - Train = 0.91
       R-Squared - Test = 0.92
       RSS - Train = 12.83
       RSS - Test = 3.20
       MSE - Train = 0.01
       MSE - Test = 0.01
       RMSE - Train = 0.10
       RMSE - Test = 0.10

[160]: # Lasso model

[161]: lasso_metrics_table = {'Metrics': ['R2 Score (Train)','R2 Score (Test)','RSS (Train)','RSS (Test)',
                                          'MSE (Train)','MSE (Test)', 'RMSE (Train)', 'RMSE (Test)'],
                              'Lasso Regression' : lasso_metrics
                             }

       final_metric = pd.DataFrame(lasso_metrics_table, columns = ['Metrics', 'Lasso Regression'] )
       final_metric.set_index('Metrics')
```

[161]:

| Metrics | Lasso Regression |
|---|---|
| R2 Score (Train) | 0.913648 |
| R2 Score (Test) | 0.921713 |
| RSS (Train) | 12.826628 |
| RSS (Test) | 3.198525 |
| MSE (Train) | 0.010982 |



jupyter advanced regression-ssaxen Last Checkpoint: 32 minutes ago

[161]:

| Metrics | Lasso Regression |
|---|---|
| R2 Score (Train) | 0.913648 |
| R2 Score (Test) | 0.921713 |
| RSS (Train) | 12.826628 |
| RSS (Test) | 3.198525 |
| MSE (Train) | 0.010982 |
| MSE (Test) | 0.010954 |
| RMSE (Train) | 0.104794 |
| RMSE (Test) | 0.104661 |

```
[162]: # let's find the variable

       empty_df = pd.DataFrame(index=X_train_dropped.columns)
       empty_df.rows = X_train_dropped.columns
       empty_df
```

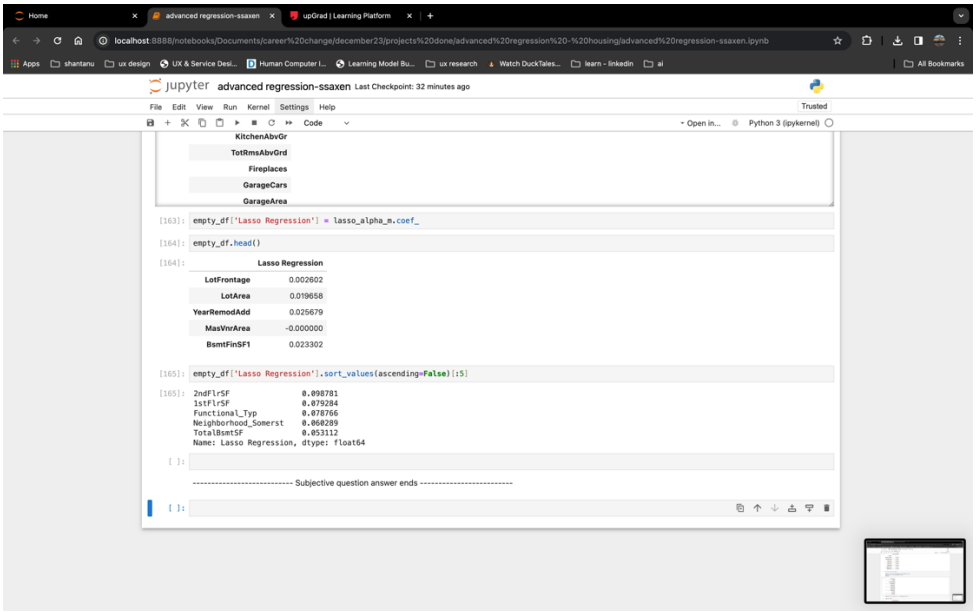
[162]:

| |
|---|
| LotFrontage |
| LotArea |
| YearRemodAdd |
| MasVnrArea |
| BsmtFinSF1 |
| BsmtFinSF2 |
| BsmtUnfSF |
| TotalBsmtSF |
| 1stFlrSF |
| 2ndFlrSF |

```
[163]: empty_df['Lasso Regression'] = lasso_alpha_m.coef_

[164]: empty_df.head()
```

[164]:

| | Lasso Regression |
|---|---|
| LotFrontage | 0.002602 |
| LotArea | 0.019658 |
| YearRemodAdd | 0.025679 |
| MasVnrArea | -0.000000 |
| BsmtFinSF1 | 0.023302 |



jupyter advanced regression-ssaxen Last Checkpoint: 32 minutes ago

```
[165]: empty_df['Lasso Regression'].sort_values(ascending=False)[:5]

[165]: 2ndFlrSF              0.098781
       1stFlrSF              0.079284
       Functional_Typ        0.078766
       Neighborhood_Somerst  0.060289
       TotalBsmtSF           0.053112
       Name: Lasso Regression, dtype: float64

[ ]:

       ---------- Subjective question answer ends ----------

[ ]:
```

SHANTANU SAXENA

**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:

To make sure the model is robust and generalisable we can keep following in mind while making model :-

- By training the data on diverse dataset that covers wide range of scenarios and variations that can be faced later on.
- Cross verify the model performance on various data sets and check for consistency.
- If a model looks too perfect than it has some issue and please cross verify the model.
- Manage the balance between the accuracy and complexity of model because a complex model tends to fail.
- Check for model's performance on unseen dataset for accuracy and robustness

Implications of above strategies are significant but if a model is having good accuracy, robustness and less complex than it over powers the implications.