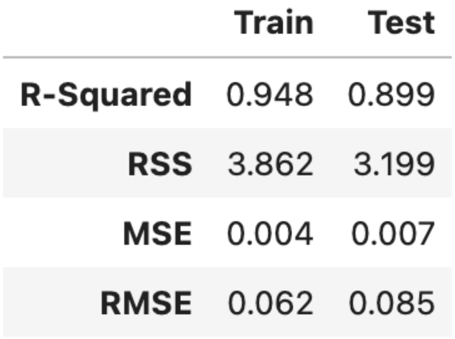
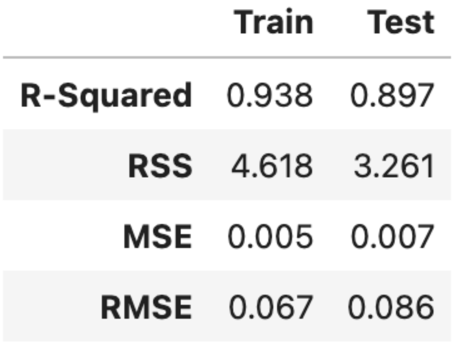
# Assignment-based Subjective Questions

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?**

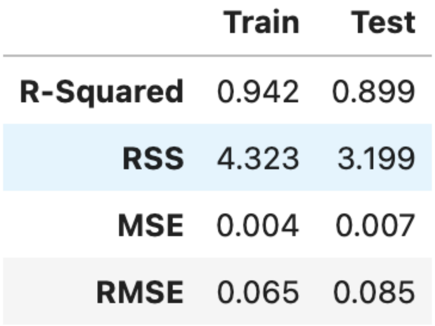
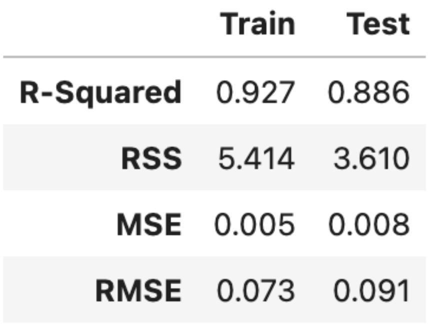
The optimum values of alpha for ridge and lasso regression are

1. Ridge Regression : 7.0
2. Lasso Regression : 0.0005

**Key Error Values**

<table>
<tr><th>Ridge Regression</th><th>Lasso Regression</th></tr>
<tr>
<td>

<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td>R-Squared</td><td>0.948</td><td>0.899</td></tr>
<tr><td>RSS</td><td>3.862</td><td>3.199</td></tr>
<tr><td>MSE</td><td>0.004</td><td>0.007</td></tr>
<tr><td>RMSE</td><td>0.062</td><td>0.085</td></tr>
</table>
</td>
<td>

<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td>R-Squared</td><td>0.938</td><td>0.897</td></tr>
<tr><td>RSS</td><td>4.618</td><td>3.261</td></tr>
<tr><td>MSE</td><td>0.005</td><td>0.007</td></tr>
<tr><td>RMSE</td><td>0.067</td><td>0.086</td></tr>
</table>
</td>
</tr>
</table>

Here are the new error terms after doubling the alpha values.

<table>
<tr><th>Ridge Regression</th><th>Lasso Regression</th></tr>
<tr>
<td>

<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td>R-Squared</td><td>0.942</td><td>0.899</td></tr>
<tr><td>RSS</td><td>4.323</td><td>3.199</td></tr>
<tr><td>MSE</td><td>0.004</td><td>0.007</td></tr>
<tr><td>RMSE</td><td>0.065</td><td>0.085</td></tr>
</table>
</td>
<td>

<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td>R-Squared</td><td>0.927</td><td>0.886</td></tr>
<tr><td>RSS</td><td>5.414</td><td>3.610</td></tr>
<tr><td>MSE</td><td>0.005</td><td>0.008</td></tr>
<tr><td>RMSE</td><td>0.073</td><td>0.091</td></tr>
</table>
</td>
</tr>
</table>

You can see that there is a marginal increase in mean squared error in lasso regression (0.007 to 0.008 in the case of test data).

There is a marginal reduction in the train r-squared values of train data in both the ridge and lasso regression.  The r-squared for test has shown a small decrease in case of lasso, but no decrease in case of ridge regression.

The top predictor variables and their beta coefficients for the ridge regression after doubling the alpha to 14 are:

| **Features** | **Beta Coefficients** |
|---|---|
| GrLivArea | 0.117 |
| TotalBsmtSF | 0.09 |
| OverallQual_Very Good | 0.077 |
| OverallQual_Excellent | 0.073 |
| 1stFlrSF | 0.073 |

The top predictor variables and their beta coefficients for the lasso regression after doubling the alpha to 0.001 are:

| **Features** | **Beta Coefficients** |
|---|---|
| GrLivArea | 0.301 |
| TotalBsmtSF | 0.126 |
| OverallQual_Excellent | 0.115 |
| OverallQual_Very Good | 0.094 |
| Age | -0.082 |

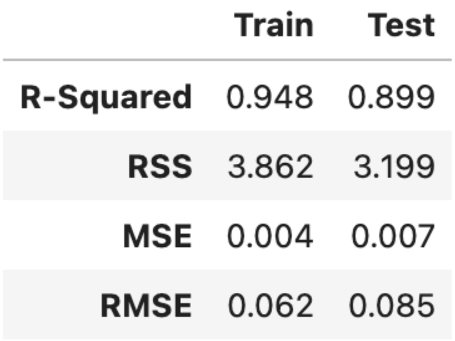
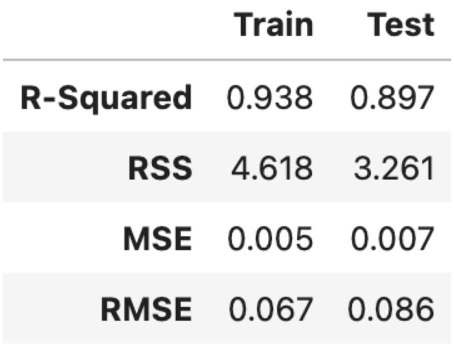
**2. You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?**

Looking at the key model metrics for ridge & lasso methods again

## Alpha values

Optimum alpha value for ridge regression - 7.0
Optimum alpha value for ridge regression - 0.0005

**Key Error Values**

<table>
<tr><th>Ridge Regression</th><th>Lasso Regression</th></tr>
<tr>
<td>
<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td><b>R-Squared</b></td><td>0.948</td><td>0.899</td></tr>
<tr><td><b>RSS</b></td><td>3.862</td><td>3.199</td></tr>
<tr><td><b>MSE</b></td><td>0.004</td><td>0.007</td></tr>
<tr><td><b>RMSE</b></td><td>0.062</td><td>0.085</td></tr>
</table>
</td>
<td>
<table>
<tr><th></th><th>Train</th><th>Test</th></tr>
<tr><td><b>R-Squared</b></td><td>0.938</td><td>0.897</td></tr>
<tr><td><b>RSS</b></td><td>4.618</td><td>3.261</td></tr>
<tr><td><b>MSE</b></td><td>0.005</td><td>0.007</td></tr>
<tr><td><b>RMSE</b></td><td>0.067</td><td>0.086</td></tr>
</table>
</td>
</tr>
</table>

If we look at the r-squared values, ridge is giving marginally better results on train & test data. And it has lower MSE & RMSE values. But, lasso has the additional advantage of helping select features.

In this case, since the r-squared values and the mean squared errors are very close to each other, I will go with the lasso method since it gives additional advantage of feature selection.

3. **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?**

After dropping the 5 most important predictor variables, here are the most important ones now

1. 1stFlrSF
2. 2ndFlrSF
3. GarageArea
4. Intercept
5. BsmtFinSF1

## 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?

Ensuring a linear regression model is robust and generalizable is important part of model creation and involves several steps to enhance its performance and applicability:

**Quality Data**: Begin with clean, high-quality data. Remove outliers, handle missing values appropriately, and use relevant features.

**Feature Selection and Engineering**: Choose relevant features that contribute meaningfully to the model's predictive ability. Feature engineering, such as creating new features or transforming existing ones, can improve the model's performance.

**Cross-Validation**: Employ techniques like k-fold cross-validation to test the model on different subsets of the data, reducing the risk of overfitting and providing a better estimate of its performance.

**Regularization Techniques**: Use techniques like Lasso or Ridge regression to prevent overfitting by adding penalties to the regression coefficients, discouraging excessively large coefficients.

**Residual Analysis**: Residual analysis is a key step to test that the model is robust and that the assumptions of linear regression hold. Test the residuals to ensure they follow a normal distribution and don't exhibit any patterns, indicating the model captures all relevant information.

**Assess Model Performance Metrics**: Assess the model's performance using appropriate evaluation metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or

R-squared. A model that generalizes well will perform similarly on both the training and test datasets.

Striving for a more robust and generalizable model might, in some cases, sacrifice a bit of accuracy on the training set. This is because the model is not tailored too specifically to the training data, focusing instead on understanding the underlying patterns that can be generalized to new data. However, the trade-off is worthwhile as it helps prevent overfitting and ensures the model performs well on unseen data.

### Bias-Variance Trade-off

Balancing bias and variance is crucial.

A model that is too simple might have high bias and low variance (underfitting), leading to low accuracy. On the other hand, a model too complex might have low bias and high variance (overfitting), performing well on training data but poorly on new data. Achieving a balance between these aspects is key to developing a robust and generalizable model.

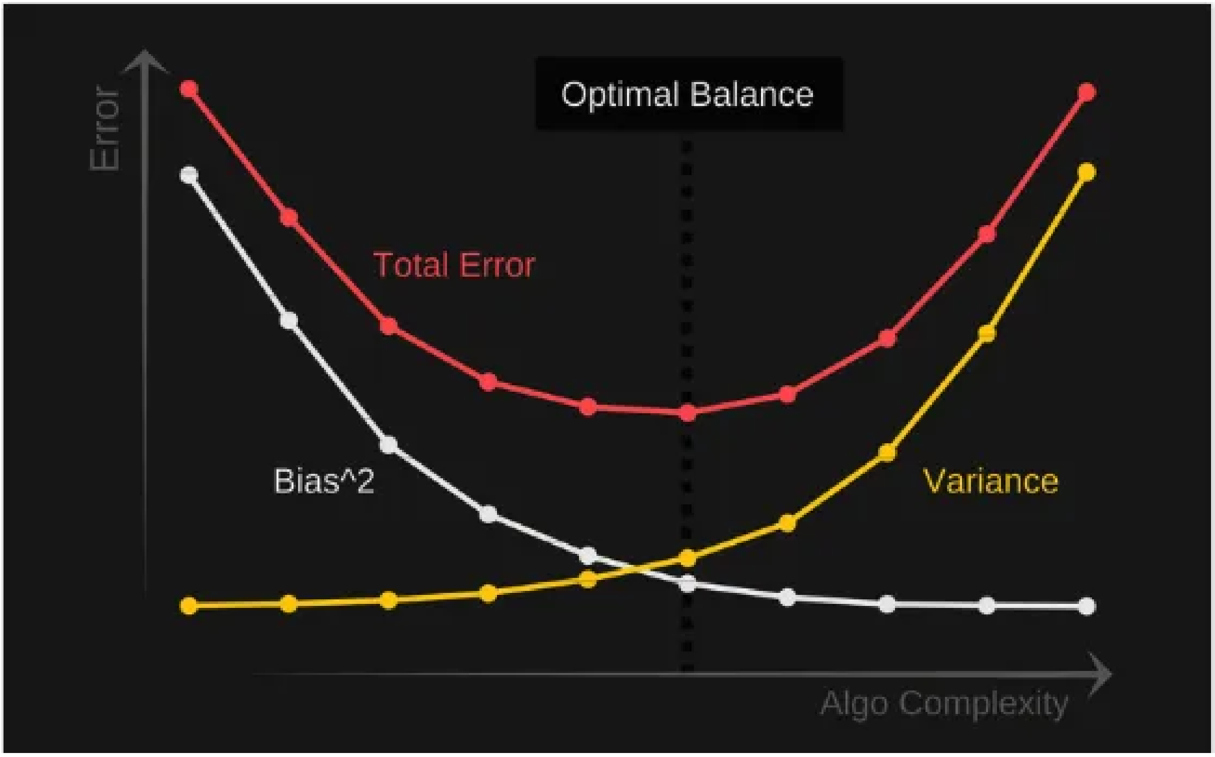
Bias error occurs when a model is too simple to capture the underlying patterns in the data, resulting in systemic errors. In other words, the model's predictions are consistently different from the actual values.

Variance error, on the other hand, occurs when a model is too complex and captures the noise or random fluctuations in the training data instead of the underlying patterns. This results in overfitting, where the model performs well on the training data but poorly on new, unseen data.

There are various techniques to achieve a balance between bias & variance

- Increase the amount and diversity of training data to help the model learn the underlying patterns more accurately.
- Use regularization techniques such as ridge or lasso regularization to reduce variance by adding a penalty term to the loss function.
- Choose a simpler model that has lower capacity, i.e., fewer parameters, to reduce variance and avoid overfitting.
- Find the right balance of model complexity - too simple and model underfits, too complex and it overfits.
- Use cross-validation techniques to evaluate the performance of the model on new data and adjust the hyperparameters accordingly.

The bias-variance tradeoff is visualized through a graph between algorithmic complexity and error. See image below for a visual representation (source: Understanding Bias-Variance Tradeoff


Error
Optimal Balance
Total Error
Bias^2
Variance
Algo Complexity