Assignment-based Subjective Questions

Q1. 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?

Answer: -

a) What is the optimal value of alpha for ridge and lasso regression?

- The optimal value of alpha for Ridge regression is 500 and Lasso regression is 1000.

b) What will be the changes in the model if you choose double the value of alpha for both ridge and lasso?

- **If we double alpha, the models will use even stronger regularization.**
- This will shrink the coefficients further making them smaller in magnitude.
- As a result, **the models will become simpler and focuses more on the most important features**.

c) What will be the most important predictor variables after the change is implemented?

- For Ridge regression, the top 5 important variables (with the largest coefficients) are:
  - *"OverallQual", "GrLivArea", "Neighborhood_NoRidge", "1stFlrSF",* and *"Neighborhood_NridgHt"*.
- For Lasso regression, the top 5 important variables are:
  - *"GrLivArea", "OverallQual", "Condition2_PosN", "Neighborhood_NridgHt", and "GarageCar*.

```
# Compare Ridge coefficients before and after doubling alpha
ridge_orig_coefs = pd.Series(ridge_best.coef_, index=X_train.columns)
ridge_double_coefs = pd.Series(ridge_double.coef_, index=X_train.columns)
print("Ridge - Mean absolute coefficient (original alpha):", ridge_orig_coefs.abs().mean())
print("Ridge - Mean absolute coefficient (double alpha):", ridge_double_coefs.abs().mean())

# Compare Lasso coefficients before and after doubling alpha
lasso_orig_coefs = pd.Series(lasso_best.coef_, index=X_train.columns)
lasso_double_coefs = pd.Series(lasso_double.coef_, index=X_train.columns)
print("Lasso - Mean absolute coefficient (original alpha):", lasso_orig_coefs.abs().mean())
print("Lasso - Mean absolute coefficient (double alpha):", lasso_double_coefs.abs().mean())
```

```
Ridge - Mean absolute coefficient (original alpha): 1086.0818551326024
Ridge - Mean absolute coefficient (double alpha): 926.6028525307662
Lasso - Mean absolute coefficient (original alpha): 852.0384647322263
Lasso - Mean absolute coefficient (double alpha): 618.7328210942761
```

```python
# 1. Optimal alpha values
print("Optimal Ridge alpha:", ridge_cv.best_params_['alpha'])
print("Optimal Lasso alpha:", lasso_cv.best_params_['alpha'])

# 2. Fit Ridge and Lasso with double the optimal alpha
ridge_double = Ridge(alpha=2 * ridge_cv.best_params_['alpha'])
ridge_double.fit(X_train, y_train)
lasso_double = Lasso(alpha=2 * lasso_cv.best_params_['alpha'])
lasso_double.fit(X_train, y_train)

# 3. Most important predictors after doubling alpha
# For Ridge: top 10 by absolute coefficient
ridge_double_coefs = pd.Series(ridge_double.coef_, index=X_train.columns)
print("\nTop 5 Ridge predictors (double alpha):")
print(ridge_double_coefs.abs().sort_values(ascending=False).head(5))

# For Lasso: non-zero coefficients, sorted by absolute value
lasso_double_coefs = pd.Series(lasso_double.coef_, index=X_train.columns)
important_lasso = lasso_double_coefs[lasso_double_coefs != 0].abs().sort_values(ascending=False)
print("\nTop 5 Lasso predictors (double alpha):")
print(important_lasso.head(5))
```

```
Optimal Ridge alpha: 500
Optimal Lasso alpha: 1000

Top 5 Ridge predictors (double alpha):
OverallQual             7113.223661
GrLivArea               6757.685411
Neighborhood_NoRidge    5188.843648
1stFlrSF                4855.019295
Neighborhood_NridgHt    4533.970545
dtype: float64

Top 5 Lasso predictors (double alpha):
GrLivArea               25549.127696
OverallQual             20213.698067
Condition2_PosN          8477.639763
Neighborhood_NridgHt     7409.673862
GarageCars               7316.644603
dtype: float64
```

Q2. 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?

Answer:-

Both Ridge and Lasso regression are effective regularization techniques, but the choice depends on the problem and data characteristics:

- **Ridge regression** is preferred when we believe that many features contribute to the target and we want to shrink their coefficients but not eliminate any. It is effective when multicollinearity is present and we want to retain all predictors.
- **Lasso regression** is preferred when we suspect that only a few features are truly important. Lasso can shrink some coefficients to zero, effectively performing feature selection and resulting in a simpler, more interpretable model.

I would choose Lasso regression for this assignment because my aim is to **pinpoint the most important predictors and make the model simpler.** Lasso helps achieve this by automatically selecting key features and reducing the coefficients of less relevant ones to zero.

Q3. 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?

Answer:-

After discovering that the five most important predictor variables identified by the Lasso model are not available in the incoming data, I would need to rebuild the model without these variables. To find the next set of important predictors, I would:

1. **Remove the top five features** (with the highest absolute coefficients) from my training data.
2. **Retrain the Lasso model** on the remaining features.
3. **Check the new model's coefficients and identify the next five most important predictor variables based on their absolute values.**

The five most important predictor variables now would be the ones with the largest absolute coefficients in the updated Lasso model, after excluding the original top five.

The next five most important predictor variables are:
"2ndFlrSF"
"1stFlrSF"
"KitchenQual_Gd"
"KitchenQual_TA"
"BsmtQual_Gd"

```
# 1. Get the top 5 most important features from the Lasso model
lasso_coefs = pd.Series(lasso_best.coef_, index=X_train.columns)
top5_features = lasso_coefs.abs().sort_values(ascending=False).head(5).index.tolist()
top5_features
```

✓ 0.0s

```
['GrLivArea',
 'OverallQual',
 'Condition2_PosN',
 'Neighborhood_NridgHt',
 'Neighborhood_NoRidge']
```

```
# 2. Remove these features from X_train and X_test
X_train_new = X_train.drop(columns=top5_features)
X_test_new = X_test.drop(columns=top5_features)
```

✓ 0.0s

```
# 3. Retrain the Lasso model on the reduced feature set
lasso_new = Lasso(alpha=lasso_cv.best_params_['alpha'])
lasso_new.fit(X_train_new, y_train)
```

✓ 0.0s

```
# 4. Get the next five most important features
lasso_new_coefs = pd.Series(lasso_new.coef_, index=X_train_new.columns)
next5_features = lasso_new_coefs.abs().sort_values(ascending=False).head(5)
print("The next five most important predictor variables are:")
print(next5_features)
```

✓ 0.0s

```
The next five most important predictor variables are:
2ndFlrSF          23264.056489
1stFlrSF          18876.336120
KitchenQual_Gd    10465.724440
KitchenQual_TA    10215.985749
BsmtQual_Gd        9926.060062
dtype: float64
```

Q4. 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?

Answer:-

a) How can you make sure that a model is robust and generalisable?

- I would perform thorough **data cleaning and preprocessing**, such as **handling missing values** and encoding categorical variables to ensure the quality of the input data.
- **Derived new variables** can further enhance model performance and interpretability.
- To make sure that a model is robust and generalisable, I would use techniques such as **cross-validation, regularization using Ridge and Lasso regression and by ensuring the model is tested on unseen data (as a separate test set).**
- I would also avoid **overfitting** by not making the model too complex and by using feature selection to keep only the most relevant predictors.
- I would monitor **model performance using appropriate metrics such as R^2 and RMSE** on both training and test sets to confirm that the model maintains accuracy and reliability when applied to new data.

b) What are the implications of the same for the accuracy of the model and why?

- **The implication of having a robust and generalisable model is that it will perform well not just on the training data, but also on new, unseen data.** This means the accuracy measured on the test set will be a reliable indicator of real-world performance.
- **If a model is not generalisable, it may show high accuracy on the training data but poor accuracy on new data,** which is not useful in practice. Robust models help ensure that predictions remain accurate and consistent when applied to different datasets.