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?

Answer 1:

- The optimal lamda (or alpha) for Lasso and Ridge regression as following:
    - Lasso(alpha=100) and Ridge(alpha=10): for all full 259 features as model's inputs/predictors
    - Lasso(alpha=100) and Ridge(alpha=1): for top 12 significant features as model's inputs/predictors

- If you choose double the value of alpha for both ridge and lasso, the change in the model of all full 259 features as following:
    - original= {'best_estimator': Lasso(alpha=100),'r2_score_train': 0.8976, 'r2_score_test': 0.8723, 'MAE test': 19242.8}
    - changed => {'best_estimator': Lasso(alpha=200), 'r2_score_train': 0.8744, 'r2_score_test': 0.8647, 'MAE test': 19437.7}
    - original= {'best_estimator': Ridge(alpha=10), 'r2_score_train': 0.8832, 'r2_score_test': 0.8586, 'MAE test': 20292.69}
    - changed => {'best_estimator': Ridge(alpha=20), 'r2_score_train': 0.8641, 'r2_score_test': 0.8451, 'MAE test': 20704.38}

    => Conclusion: overall r2_score (train and test) decreased and MAE (test) increased, it means the models with double the value of optimal alpha would perform worse than the original optimal alpha ones

- There are 84 significant predictors with new alpha as resulted from the models. And the most important predictor variables after the change is implemented are GrLivArea, OverallQual, GarageCars , Neighborhood_NoRidge and Neighborhood_StoneBr etc. all are sorted from highest to lowest importance above

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?

Answer 2:

The optimal lamda (or alpha) for Lasso and Ridge regression as following:

+ Lasso(alpha=100) and Ridge(alpha=10): for all full 259 features as model's inputs/predictors

+ Lasso(alpha=100) and Ridge(alpha=1): for top 12 significant features as model's inputs/predictors

And I am choosing the Lasso(alpha=100) to apply, because both Lasso(alpha=100) and Ridge(alpha=10) have resulted well r2_score > 0.8 in both train and test dataset , overcoming the overfitting issue and but the Lasso(alpha=100) has reduced the model features (complexity) from 259 to 156 features (coef is not zero), which makes us easily interpret the features' significance to the House sale price ultimately for effective decision-making support.

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?

Answer 3:

After the change (removed top 5 important predictors), the top 5 new predictors are TotRmsAbvGrd, 2ndFlrSF, KitchenQual_TA, FullBath, and KitchenQual_Gd (also shown in the Jupyter notebook model – Part 2)

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?

Answer 4:

We can make sure that a model is robust and generalizable if it can fulfill the following criteria:

- Assumptions on residuals are validated and passed (for a valid regression model)
- r2_score (accountable for variances) on training and testing dataset are approximately similar and high (more than 0.8)
- Regularization techniques such as Lasso and Ridge regression are applied to prevent overfitting and reduce model complexity (features reduction by Lasso)
- GirdSearchCV (HyperParameter finetuning with Cross-validation such as k-Fold) technique is applied to let the models go thru multiple subsets of data and be validated, then reduce the risks of overfitting and return the best fit model with best score, performance and set of parameters.
- Data pre-processing techniques such as outliers handling, feature engineering, and data encoding can be helpful to let the model consume quality data input, then it can result more precise, reliable and robust performance.

The implications of the same for the accuracy of the model are training performance and testing performance should be well-important for both similarly. Because, if the model is too simple, has poor training performance (low r2_score in both training and testing), it might have high bias-low variance and underfitting, where it cannot predict precisely as expected; in contrast, if the model is too complex, has perfect training performance (very high r2_score in training, but low in testing), it might be overfitting and unable to generalize or predict unseen data precisely. Therefore, we need the accuracy of the model is the same high in both training and testing, where we can be confident on the model's generalization capability to predict unseen data precisely in practice.