Problem Statement - Part II

Q1. What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose to double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer: Below are the results for different alpha values in Lasso regression. Considering Low RMSE along with minimal difference between train and test results, I selected the value of alpha as 0.01.

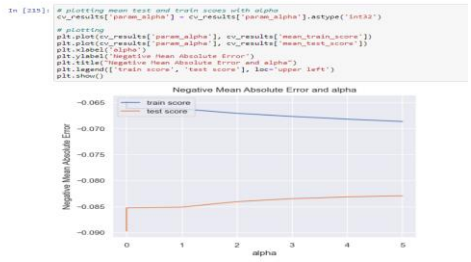
Lasso Regression

```
alpha = 0.001
0.92067
0.90677
RMSE : 0.11546

alpha = 0.01
0.88546
0.88946
RMSE : 0.12573

alpha = 0.05
0.81057
0.82914
RMSE : 0.15631

alpha = 0.1
0.70122
0.71900
RMSE : 0.20047
```

From the negative mean absolute error value vs alpha graph for Ridge Regression we can clearly see that after the alpha value of 2 the graph stabilizes. Hence I went ahead and chose alpha value as 2 and RMSE seems to be pretty good as well.

Ridge Regression:

```
alpha = 2
0.93645
0.90775
RMSE : 0.11485
```



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

If you double the value of alpha for both ridge and lasso, the model will become more regularized, and the coefficients for the predictor variables will shrink further towards zero. This means that the model will be more biased but less prone to overfitting.

- If we double the alpha value of Lasso Regression(i.e. to 0.02), RMSE increases but test and train value gets decreased

  ```
  alpha = 0.02
  0.87118
  0.88101
  RMSE : 0.13045
  ```

- If we double the alpha value of Ridge Regression(i.e. to 4), there is no considerable difference

  ```
  alpha = 4
  0.93442
  0.90865
  RMSE : 0.11429
  ```

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

For Ridge the top 5 important predictor value are given in the below screen shot.

```
In [279]: predRFE.head(10)
```

Out[279]:

	Variable	Coeff
0	constant	11.739
29	MSZoning_FV	0.149
31	MSZoning_RL	0.125
50	Neighborhood_Crawfor	0.114
30	MSZoning_RH	0.105
32	MSZoning_RM	0.097

For Lasso the top 5 important predictor value are given in the below screen shot.

```
In [207]: # Chose variables whose coefficients are non-zero
pred = pd.DataFrame(para[(para['Coeff'] != 0)])
pred
```

Out[207]:

	Variable	Coeff
0	constant	12.003
13	GrLivArea	0.125
4	OverallQual	0.112
5	OverallCond	0.050
9	TotalBsmtSF	0.042
7	BsmtFinSF1	0.035

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?

Lasso Regression:

```
alpha = 0.01
0.88546
0.88946
RMSE : 0.12573
```

Ridge Regression:

```
alpha = 2
0.93645
0.90775
RMSE : 0.11485
```

I will choose Lasso Regression because the prediction becomes easy since the insignificant parameters are removed. This discourages the model from becoming too complex, avoiding the risk of overfitting. Ridge is more efficient when compared to Lasso but it has too many parameters to consider.
If the Business team is concerned only with the train, test and RMSE results, then only in that case I will opt for Ridge Regression.

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?

Below are the variables which are the most important predictor variables after creating another model

	Variable	Coeff
0	constant	12.003
13	BedroomAbvGr	0.125
4	MasVnrArea	0.112
5	BsmtFinSF2	0.050
9	BsmtFullBath	0.042
7	2ndFlrSF	0.035

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?

Ensuring that a model is robust and generalizable involves several practices to enhance its performance on new, unseen data. Here are key strategies and their implications for model accuracy:

Cross-Validation: Use techniques like k-fold cross-validation to assess the model's performance on multiple subsets of the data. Cross-validation helps ensure that the model's performance is consistent across different partitions of the dataset. This gives a more reliable estimate of the model's generalization performance.

Implication: Cross-validation provides a more robust estimate of the model's accuracy by evaluating its performance on multiple data subsets, reducing the risk of overfitting to a specific dataset.

Hold-Out Test Set: Reserve a portion of the data as a test set that the model has not seen during training. This set is used for final evaluation after model training is complete.

Implication: The performance on the hold-out test set gives an indication of how well the model generalizes to new, unseen data. If the model performs well on the test set, it's likely to be more generalizable.

Feature Engineering: Select relevant features, apply transformations, and preprocess the data effectively to capture important patterns and relationships in the data.

Implication: Thoughtful feature engineering can improve the model's ability to generalize by focusing on relevant information and reducing noise.

Regularization: Use regularization techniques like Ridge or Lasso regression to prevent overfitting. Regularization adds a penalty for complex models, promoting simplicity and better generalization.

Implication: Regularization helps prevent the model from fitting the training data too closely, making it more likely to generalize well to new data.

Model Evaluation Metrics: Choose appropriate evaluation metrics that align with the problem and the desired model behavior. For example, use metrics like precision, recall, or F1 score for classification problems.

Implication: Selecting metrics that reflect the model's performance on specific aspects of the data ensures that the model is evaluated based on the most relevant criteria.

Ensemble Methods: Consider using ensemble methods like Random Forests or Gradient Boosting, which combine multiple models to improve generalization.

Implication: Ensemble methods leverage the strengths of multiple models to create a more robust and accurate prediction.

Data Augmentation: For certain types of data (e.g., images, text), apply data augmentation techniques to increase the diversity of the training dataset.

Implication: A diverse training dataset helps the model learn a broader range of patterns, making it more adaptable to variations in new data.

Ensuring robustness and generalization is crucial for a model's success in real-world scenarios. The implications for model accuracy involve achieving a balance between fitting the training data well and being able to make accurate predictions on new, unseen data. A model that is too complex or over fit to the training data may perform poorly on new data, resulting in reduced accuracy and reliability. Therefore, strategies that promote robustness and generalization contribute to improved accuracy on unseen data, which is a key goal in machine learning applications.