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

Alpha is a hyperparameter that controls how strongly the model penalizes large coefficients. Mathematically, it is the weight of the penalty term added to the loss function.

There is no single universally optimal value of α (alpha) for Ridge or Lasso regression. Alpha must be tuned from data, and its optimal value depends entirely on:

- the scale and noise level of your features
- the signal-to-noise ratio
- the amount of multicollinearity
- the underlying true sparsity of the model

Doubling alpha (α) increases the strength of regularization.

Ridge penalizes squared coefficients:

- Coefficients shrink more strongly
- Model becomes more biased
- Variance decreases (model becomes more stable)
- Coefficients do not become zero (Ridge never eliminates features)
- Decision boundary becomes smoother / less flexible

Lasso penalizes absolute values of coefficients, so if you double α:

- Coefficients shrink more aggressively
- More coefficients become exactly zero
- Stronger feature selection
- Model becomes simpler and more sparse
- Bias increases, variance decreases

Ridge: those with largest coefficients and Lasso: those that remain non-zero.

Ridge Regression

Example:

- Before doubling α: β = [4, 2, 0.5]
- After doubling α: β = [2.5, 1.3, 0.3]

Lasso Regression

- Before doubling α: β = [6, 3, 0.8, 0.1]
- After doubling α: β = [4, 2, 0, 0]

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

I Choose Lasso Regression
Because:

- It improved accuracy
- It performed feature selection
- It produced meaningful coefficient shrinkage
- It generalizes well
- Ridge regularization had almost *no effect* on your dataset
- Lasso gives a simpler, more interpretable model for house price prediction

And the Best alpha ≈ 0.0001

➢ Lasso generalizes well with stable Test performance
In our Lasso graph:

- Train and Test MAE lines were close
- No overfitting
- Smooth decreasing error trend

This indicates a well-generalizing model.

➢ Ridge graph was completely flat — no improvement
our Ridge plot showed:

- Train and test errors were almost identical
- Errors barely changed across alpha 0 → 5
- No gain from regularization

This means:
Ridge is not adding value — the model behaves the same regardless of alpha.
So applying Ridge does not help your model improve.

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:

Some of the important predictor before applying lasso were
Importance ≈ |coefficient| (assuming features were scaled or are on comparable scales).

OverallQual	0.007
HalfBath	0.006
BsmtFullBath	0.004
FullBath	0.004
Foundation_pconc	0.004
KitchenAbvGr	-0.004

OverallCond	0.003
TotRmsAbvGrd	0.003
GarageType_attchd	0.002
ExterQual_ta	-0.002
KitchenQual_ta	-0.001
BedroomAbvGr	0.001

Identify the 5 strongest (original) predictors
Sorted by absolute value, there's a tie at |coef| = 0.004:

1. OverallQual (0.007)
2. HalfBath (0.006)
   3–6. KitchenAbvGr (−0.004), BsmtFullBath (0.004), FullBath (0.004), Foundation_pconc (0.004) ← *four-way tie*

Because of the tie, any five among these six could be considered "most important."
You mentioned five are missing from incoming data, but didn't specify which five of these six. So the "next five" depends on which one of the tied features *remains*.

The "next five" after excluding the top five
Case A — if I exclude:
OverallQual, HalfBath, KitchenAbvGr, BsmtFullBath, FullBath
Then the next five (largest |coef| among remaining) are:

1. Foundation_pconc (0.004)
2. OverallCond (0.003)
3. TotRmsAbvGrd (0.003)
4. GarageType_attchd (0.002)
5. ExterQual_ta (−0.002)

Case B — if i exclude:
OverallQual, HalfBath, BsmtFullBath, FullBath, Foundation_pconc
Then the next five are:

1. KitchenAbvGr (−0.004)
2. OverallCond (0.003)
3. TotRmsAbvGrd (0.003)
4. GarageType_attchd (0.002)
5. ExterQual_ta (−0.002)

Question 4
How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?
Answer:

<u>How can you make sure that a model is robust and generalizable?</u>
To ensure robustness and generalization, I followed several important steps in our model-building workflow:

- Train/Test Split → Prevents Overfitting

Trained the model on one dataset and tested it on another.
This ensures the model learns patterns, not just memorizes data.

✓ Why this improves generalization?

If training $R^2 \approx$ test R^2 (which you achieved), it means:

- The model doesn't overfit
- It performs similarly on unseen data

❖ Cross-Validation (GridSearchCV) → Makes the model robust

Used GridSearchCV, which:

- Splits the data into multiple folds
- Trains and tests the model repeatedly
- Averages performance

This stabilizes the performance and reduces dependence on any single data split.

Why this improves generalization?

Whatever lambda (λ) the model chooses is not due to luck, but due to consistent performance across folds.

❖ Regularization (Lasso & Ridge) → Controls model complexity

You applied:

- Lasso (L1): removes weak features
- Ridge (L2): shrinks coefficients

Regularization prevents the model from having large, unstable coefficients.

✓ Why this improves robustness?

Regularization forces the model to be simpler and more stable, which reduces noise-fitting.

❖ Hyperparameter Tuning (alpha selection) → Optimal complexity

plotted alpha vs Negative MAE for Lasso and Ridge.

For Lasso:

- Error decreased as alpha increased
- You found the best alpha = 0.0001

For Ridge:

- Error was flat → Ridge didn't add value

✓ Why this improves generalization?

Choosing the right Lambda(λ) ensures:

- Model is not too flexible (overfitting)
- Not too rigid (underfitting)

❖ Feature Selection (RFE + Lasso) → Removes irrelevant predictors

Used RFE to reduce initial dimensionality.

Then Lasso removed even more noise features by shrinking their coefficients to zero.

Why this improves robustness?

- Fewer features → less noise
- Only truly predictive variables kept
- Simplifies the model
- Reduces variance

A simpler model generalizes better to new data.

❖ Monitoring Train vs Test R^2 and MAE → Ensures stable performance

You carefully observed that:

- Train $R^2 \approx$ Test R^2

- No major gap → no overfitting
- MAE stable across alpha
  - ✓ Why this matters?

  Close train/test scores show:
- The model performs consistently
- Predictions are stable under new data

<u>Impact on Model Accuracy and Why</u>

Ensuring robustness and generalization has specific implications for accuracy:

- ❖ Training accuracy may slightly decrease

Because:
- Regularization penalizes large coefficients
- Lasso removes features
- Model complexity is reduced

A simpler model → slightly lower train accuracy

This is expected and GOOD.

- ❖ Test accuracy increases and becomes more reliable

This is the real goal.

Your model:
- Performed similarly on train and test
- Had stable MAE across folds
- Reduced noise features via Lasso
- Chose optimal alpha

All of this improves real-world test accuracy.

- ❖ Overall accuracy becomes more trustworthy

Because you:
- Avoided overfitting
- Tuned hyperparameters properly
- Used cross-validation
- Applied regularization
- Removed weak variables

The model is stable → accuracy is consistent → predictions are reliable.